ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Midterm - Computer Networks

Prof. J.-P. Hubaux and Dr. M. H. Manshaei

November 4, 2008

Duration: 2 hours, closed book.

Please write your answers on these sheets, at the end of each question;
use extra sheets if necessary (put your name on them).

You may write your answers in English or in French.

The total number of points is 40.

This document contains 19 pages.

Student First name:
Last name:

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)

1 Short questions (5 points)

For each question, please circle a single best answer.

1. What are three common HTTP 1.1 message method fields?

   (a) GET, HTML, POST
   (b) GET, PUT, HTML
   (c) GET, UPLOAD, HTML
   (d) GET, POST, PUT
   (e) None of the above.

   GET, POST, PUT

2. Given that the requested information is not available at any intermediate databases, a purely recursive DNS query from a requesting host would follow the path:

   (a) Root name server, local name server, authoritative name server.
   (b) Authoritative name server, root name server, host name server.
   (c) Local name server, root name server, local name server, authoritative name server.
   (d) Local name server, root name server, TLD name server, authoritative name server.
   (e) None of the above.

   Local name server, root name server, TLD name server, authoritative name server.

3. The request line of text sent to an HTTP 1.1 server by a client for the URL "http://epfl.ch/class/compnet2008" is:

   (a) GET /class/compnet2008/ HTTP/1.1
   (b) HTTP/1.1 GET http://epfl.ch/class/compnet2008/
   (c) GET epfl.ch/class/compnet2008/ HTTP/1.1
   (d) HTTP/1.1 GET www.epfl.ch/class/compnet2008/

   GET /class/compnet2008/ HTTP/1.1

4. A router typically handles the following layer in the ISO/OSI reference model:

   (a) Network
   (b) Link
   (c) Routing
   (d) Forwarding

   Network

5. Consider the following Java application:

```
socket = new DatagramSocket(12345);
while (true) { socket.receive(packet); }
```

   What happens if two instances of this application are run simultaneously on one machine and 4 UDP segments arrive at port 12345?

(a) Both instances receive all 4 segments.

(b) One instance receives all 4 segments.

(c) Some segments are received by one instance, other segments are received by the other instance.

(d) One instance receives segments 1 and 3, the other receives segments 2 and 4.

One instance receives all 4 segments.

6. Assume `out` is a DataOutputStream created from a connected TCP socket and `message` is a String. We execute:
`out.writeBytes(message);`
How many TCP segments are sent and received as a result?

(a) sent: 1, received: 0

(b) sent: 1, received: 1

(c) sent: n, received: 0, where $n \in \mathbb{N}$

(d) sent: n, received: 1, where $n \in \mathbb{N}$

(e) sent: n, received: m, where $n, m \in \mathbb{N}$

sent: n, received: m, where $n, m \in \mathbb{N}$

7. Which one is the UDP checksum of the following three 16-bit words? w_1 = 0110011001100000, w_2 = 0101010101010101, w_3 = 1000111100001100.

(a) 0100101011000010

(b) 0100101011000001

(c) 1011010100111101

(d) 1011010100111110

1011010100111101

8. What is the main difference between stop-and-wait and pipelined reliable data transfer protocol?

(a) The pipelined protocol uses the NAK packets, whereas in the stop-and-wait protocol senders always wait for ACK packets.

(b) With the pipelined protocol, the sender can send several packets in row, whereas in the stop-and-wait protocol the sender cannot send the packets in row.

(c) With the pipelined protocol, the receiver must send one ACK for several packets (cumulative ACK), whereas in the stop-and-wait protocol the receiver can not send the cumulative ACK.

(d) The pipelined protocol uses timeouts, whereas the stop-and-wait protocol does not use the timeout.

With the pipelined protocol, the sender can send several packets in row, whereas in the stop-and-wait protocol the sender cannot send the packets in row.

9. In network-assisted congestion control:

(a) End systems detect congestions only based on observed congestion in the network.

(b) End systems use the ATM protocol to detect congestions.

(c) Network-layer components provide explicit feedback to the sender regarding the congestion state in the network.

(d) None of the above.

The network-layer components provide explicit feedback to the sender regarding the congestion state in the network.

10. Transport-layer packets are called:

(a) Messages

(b) Datagrams

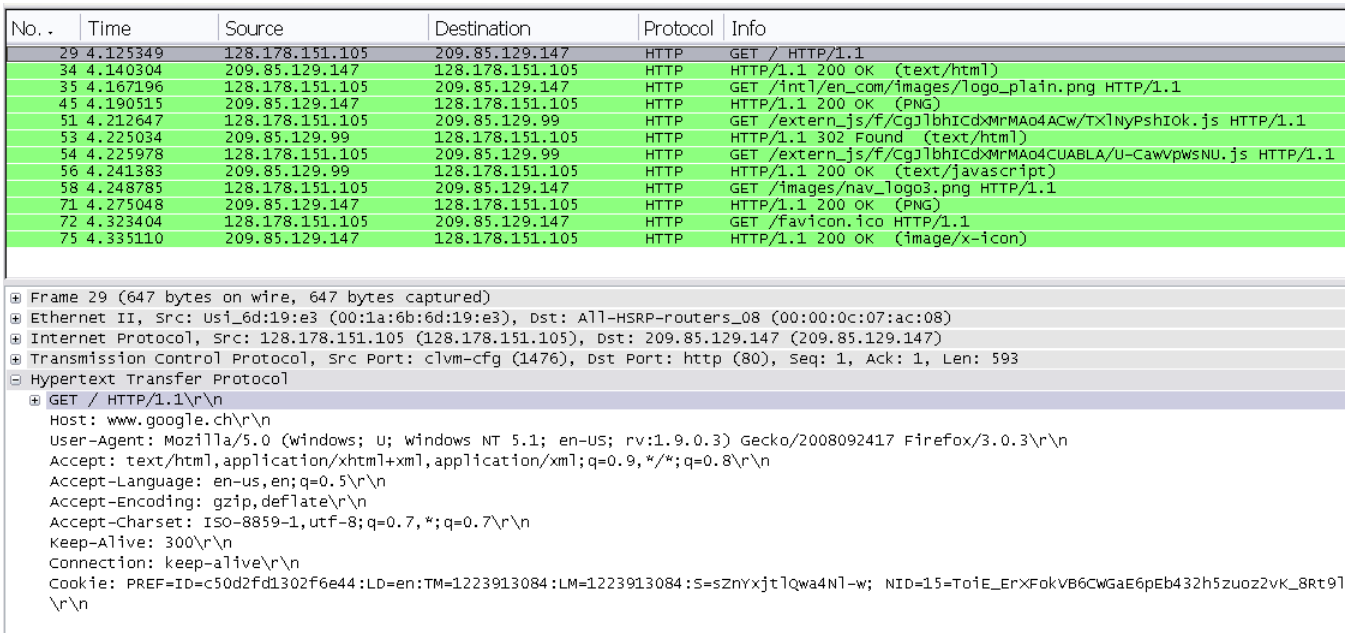
(c) Segments

(d) Frames

Segments

2 Web and HTTP (8 points)

A student at EPFL visits a webpage with a given URL. The generated traffic was captured with Wireshark and the traces are given in Figure 1 and Figure 2. By analyzing the given traces, answer the following questions:



No.	Time	Source	Destination	Protocol	Info
29	4.125349	128.178.151.105	209.85.129.147	HTTP	GET / HTTP/1.1
34	4.140304	209.85.129.147	128.178.151.105	HTTP	HTTP/1.1 200 OK (text/html)
35	4.167196	128.178.151.105	209.85.129.147	HTTP	GET /intl/en_com/images/logo_plain.png HTTP/1.1
45	4.190515	209.85.129.147	128.178.151.105	HTTP	HTTP/1.1 200 OK (PNG)
51	4.212647	128.178.151.105	209.85.129.99	HTTP	GET /extern_js/f/CgJlbhICdXMrMAo4ACw/TXlNyPshIok.js HTTP/1.1
53	4.225034	209.85.129.99	128.178.151.105	HTTP	HTTP/1.1 302 Found (text/html)
54	4.225978	128.178.151.105	209.85.129.99	HTTP	GET /extern_js/f/CgJlbhICdXMrMAo4CUABLA/U-CawVpWsNU.js HTTP/1.1
56	4.241383	209.85.129.99	128.178.151.105	HTTP	HTTP/1.1 200 OK (text/javascript)
58	4.248785	128.178.151.105	209.85.129.147	HTTP	GET /images/nav_logo3.png HTTP/1.1
71	4.275048	209.85.129.147	128.178.151.105	HTTP	HTTP/1.1 200 OK (PNG)
72	4.323404	128.178.151.105	209.85.129.147	HTTP	GET /favicon.ico HTTP/1.1
75	4.335110	209.85.129.147	128.178.151.105	HTTP	HTTP/1.1 200 OK (image/x-icon)

Figure 1: Wireshark traces - HTTP GET request

Question 1: What is the URL of the visited webpage?

The URL is www.google.ch.

Question 2: Which version of HTTP is the browser running? Which version of HTTP is the server running?

Both are running HTTP 1.1.

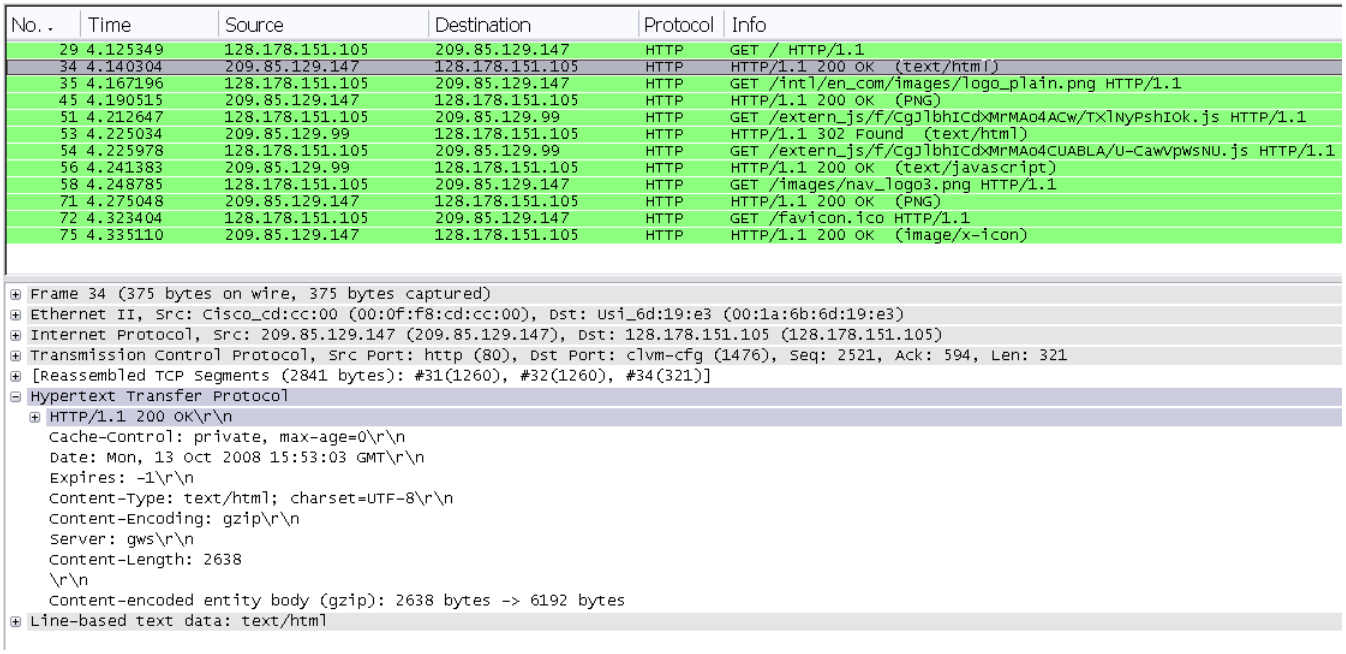


Figure 2: Wireshark traces - HTTP response from the web server

Question 3: What does the "Host" header specify? Is it necessary to specify the "Host" in the HTTP GET request for the HTTP version run by the server? Is an explicit specification of the "Host" an advantage or a disadvantage? Explain.

The "Host" header specifies the machine name from the URL, in this case www.google.ch. It is required in HTTP version 1.1. This allows a single web server to host many different domains at the same time (virtual hosting). With this header, the web server can tell from the request which web server the client was trying to contact and can respond with different content for each one. This is the advantage of HTTP 1.1 over HTTP 1.0.

Question 4: What is the IP address of the machine from which the request was sent? What is the IP address of the server that responded to the first HTTP GET request?

The IP address of the user's machine is 128.178.151.105 and 209.85.129.147 is the IP address of the server.

Question 5: What is the meaning of the "Keep-Alive" and "Connection" headers?

The "Keep-Alive" and "Connection" headers specify information about the TCP connection over which the HTTP requests and responses are sent. It indicates if the connection should be kept active following a request and for how long.

Question 6: Does the value of the "Connection" header affect the download performance? Explain.

*Connections that do not terminate after each request but rather stay open to allow multiple requests from the same server are called **persistent connections**. Persistent connections greatly improve the performance when fetching multiple objects from the same server because the overhead of creating multiple TCP connections is avoided.*

Question 7: What is the meaning of the "Cache-Control" header in the response from the web server? What is the consequence (regarding local and proxy cache) of the "Cache-Control" header value in the response given by the server?

*The "Cache-Control" header is used to specify whether and how copies of the data may be stored or cached for future reference. It can also be used to make modifications of the basic expiration mechanism. Individual web browsers typically store a cache of recently visited pages on the local machine. Similarly, groups of computers on the same network may share a cache of pages to prevent multiple users from fetching the same data (e.g. proxy cache). The "Cache-Control" value is **"private"** which indicates that the server has generated a personalized response for this user and it can be cached in user's local cache but **not** in a shared proxy cache. Note: This usage of the word "private" only controls where the response may be cached, and cannot ensure the privacy of the message content.*

Question 8: A cookie was sent with the first HTTP GET request. Why is the cookie sent? Explain what would have happened if no cookie was sent with the first HTTP GET request.

As the cookie is sent with the first GET request it means that the user has visited this site in the past. Cookies allow web sites to keep track of users. For example, cookies can be used to identify the user in order to generate content as a function of the user's identity or to restrict/allow user access. If the user was visiting www.google.ch for the first time, no cookie would be sent with the first HTTP GET request. Server would include in the HTTP response the "Set-Cookie" header with an associated value. The browser on the user's machine then appends a line to the special cookie file that it manages. The entry in the file consists of the hostname of the server and the value in the "Set-Cookie" header.

Question 9: Although a single URL was entered, there is a second HTTP GET request (packet number 35). What causes this second request? Write the URL that should be typed in the browser to fetch the object directly.

The second request requests the picture of the Google logo that is displayed on the page. It is requested by the browser and not by the user himself. The second request is equivalent to http://www.google.ch/intl/en_com/images/logo_plain.png .

3 Network Delays (3 points)

This question uses the methods that you have learned in TP2 for measuring network performance. We ran ping with packet sizes of 16 and 106 bytes from a workstation in INF2 to two locations: www.stanford.edu (West Coast, USA) and www.smu.edu.sg (Singapore). The results are summarized in the following tables:

Hostname	Packet size (bytes)	Min. (ms)	Avg. (ms)
www.stanford.edu	16	183.156	183.439
	106	183.195	183.474
www.smu.edu.sg	16	376.595	411.948
	106	377.171	449.549

Question 1: Compute approximately the end-to-end throughput of each link (INF2 workstation to www.stanford.edu and to www.smu.edu.sg, respectively).

$t_{tot} = t_{prop} + t_{trans} + t_{queue} + t_{proc}$
To assume that $t_{queue} = 0$*, we use* $t_{tot} = Min$ *in our computations*
$t_{prop} + t_{proc} = c$ *where* c *is a constant* $\Rightarrow t_{tot} = t_{trans} + c$
Hence 106 - 16 = 90 bytes are transmitted in $t_{trans,106} - t_{trans,16} = Min_{106} - Min_{16}$
*www.stanford.edu: 8*90*1000/(183.195 - 183.156) = 18.46 Mbps*
*www.smu.edu.sg: 8*90*1000/(377.171 - 376.595) = 1.25 Mbps.*

Question 2: Compute approximately the round-trip propagation delay of each link.

A good approximation is Min_{16} *because* $t_{queue} = 0$ *and* t_{trans} *is minimal. It is also correct to subtract* t_{trans} *based on 3.1*
www.stanford.edu: 183.156 ms
www.smu.edu.sg: 376.595 ms.

Question 3: What could cause the difference between the minimum and average propagation delays?

If "propagation delay" means t_{prop} then route change could cause the difference. If it means RTT, then queueing delay could cause the difference. Both answers are considered correct.

Question 4: Is one of the two destinations closer geographically to EPFL? If yes, which one? Explain your answer.

We cannot deduce this from the ping data. Although the geographical distance between two hosts can affect the end-to-end delay, there are other factors, such as link quality, that also affect the end-to-end delay. For example, the propagation speed varies depending on the medium and is much higher in optical fibers than in satellite links. Hence, the propagation delay cannot be used to compare geographical distances between hosts.

4 Network Topology (8 points)

We performed a traceroute from the same workstation as in Question 3 to www.stanford.edu and obtained the following result:

traceroute to www5.stanford.edu (171.67.20.37), 64 hops max, 40 byte packets
1 c6-ic-dit-1-v158 (128.178.158.251) 0.547 ms 0.259 ms 0.252 ms
2 c6-gigado-1-v100 (128.178.100.18) 0.373 ms 0.283 ms 0.277 ms
3 c6-ext-v200 (128.178.200.1) 0.445 ms 0.307 ms 0.282 ms
4 swiel2 (192.33.209.33) 0.686 ms 0.688 ms 0.669 ms
5 swiCE3-10GE-1-3.switch.ch (130.59.37.65) 1.513 ms 1.461 ms 1.464 ms
6 swiCE2-10GE-1-4.switch.ch (130.59.36.209) 1.462 ms 1.433 ms 1.430 ms
7 switch.rt1.gen.ch.geant2.net (62.40.124.21) 1.443 ms 1.449 ms 1.445 ms
8 so-7-2-0.rt1.fra.de.geant2.net (62.40.112.22) 9.561 ms 9.556 ms 9.611 ms
9 abilene-wash-gw.rt1.fra.de.geant2.net (62.40.125.18) 102.342 ms 102.401 ms 102.461 ms
10 so-0-0-0.0.rtr.atla.net.internet2.edu (64.57.28.6) 115.944 ms 116.341 ms 116.590 ms
11 so-3-2-0.0.rtr.hous.net.internet2.edu (64.57.28.43) 145.502 ms 139.277 ms 139.337 ms
12 so-3-0-0.0.rtr.losa.net.internet2.edu (64.57.28.44) 175.284 ms 171.293 ms 171.319 ms
13 hpr-lax-hpr–i2-newnet.cenic.net (137.164.26.132) 173.113 ms 173.652 ms 177.004 ms
14 svl-hpr–lax-hpr-10ge.cenic.net (137.164.25.13) 181.578 ms 181.007 ms 181.268 ms
15 oak-hpr–svl-hpr-10ge.cenic.net (137.164.25.9) 182.425 ms 182.434 ms 182.356 ms
16 hpr-stan-ge–oak-hpr.cenic.net (137.164.27.158) 183.414 ms 183.328 ms 183.394 ms
17 bbra-rtr.Stanford.EDU (171.64.1.134) 183.544 ms 183.536 ms 183.506 ms
18 * * *
19 www5.Stanford.EDU (171.67.20.37) 183.706 ms 183.577 ms 183.509 ms

Question 1: Is this information obtained from a circuit-switched or a packet-switched network?

Packet-switched network.

Question 2: What do the three last values in each line of the trace represent?

Three RTT measurements.

Question 3: Would a second traceroute yield the same trace? Why?

Not necessarily. It may go through other routers, for example because of network congestion. RTTs will also differ.

Question 4: Let t_{tx} be the transmission time of a given packet sent by traceroute. Assume the transmission of this packet starts at $t = 0$. Where is the last bit of this packet at time $t = t_{tx}$?

The bit is just leaving the sending host or entering the physical line.

Question 5: Assume that there are 1000 hosts connected to the first router (c6-ic-dit-1-v158) and that the capacity of any link connecting a host to this router is 1Gbps. What is the maximum possible throughput for each of these hosts?

1 Gbps, if a host is transmitting alone.

Question 6: Assume that the capacity of links halves at each hop (for example, the link between the routers c6-ic-dit-1-v158 and c6-gigado-1-v100 has a capacity of 500 Mbps). What is the end-to-end throughput of a connection between the INF2 workstation and a host at Stanford, assuming that all hosts at Stanford are connected to the same router as the Web server (www5.Stanford.EDU)?

$1Gbps/2^{18}$ = *3.815 Kbps. This is the bottleneck link.*

Question 7: Assume that a packet sent by this traceroute command arrives at the last router (number 18 in the trace). Assume there are already 4 packets waiting in the outgoing queue of this router and a fifth packet is partially transmitted; 100 ms have elapsed since the router started transmitting this fifth packet. Assume that a packet transmitted by traceroute consists of 60 bytes of header and a probe datagram; the total size of the packet = header + length of probe datagram. Assuming that the last router has the transmission rate computed in the previous question, what is the queueing delay of the arriving packet?

*The total packet size is 60 + 40 = 100 bytes. The length of the probe datagram appears in the first line of the trace. The transmission delay of a single packet is 8*100/3815 = 209.7 ms. The packet being transmitted still needs 209.7 - 100 = 109.7 ms to finish. The total queueing delay is 4*209.7 + 109.7 = 948.5 ms.*

5 DNS and Peer-to-Peer Applications (8 points)

Question 1: A domain in Domain Name System (DNS) is served by multiple nameservers. Describe the two main benefits from having multiple nameservers. Under which conditions these multiple nameservers deliver these benefits in practice?

Scalability and redundancy in the face of server and network failures. The multiple servers should fail independently and therefore should be located in different geographic locations and have separate connections to the Internet.

Question 2: What is the application architecture for DNS? What is it for Gnutella?

DNS employs the client-server model and Gnutella employs the peer-to-peer model.

Question 3: For DNS and Gnutella, briefly describe and compare the organizational structure of where information/data/files are stored.

Both DNS and Gnutella use a distributed database/filesystem. The DNS database has an hierarchical structure while the Gnutella filesystem is flat and does not have an hierarchy. DNS: distributed database of DNS servers which are arranged in an hierarchical structure (Root DNS, TLD (Top Level Domain) DNS, Authoritative DNS). Gnutella: distributed file system where files are distributed in peer end systems.

Question 4: Explain the main differences between the DNS and Gnutella application architectures.

The DNS application is of the client-server type, since the client requesting the IP address is not a peer and can never itself be a DNS server. Also, DNS servers in the hierarchical structure are not peers, that is a Root DNS is not a peer to an Authoritative DNS. In Gnutella, all peers can be clients or servers.

Question 5: In the topology shown in Figure 3, S is a Web server, machine A is a desktop client, N is a name server (but not the authoritative name server for S), C is a Web cache and R is a router. Client A is configured to use Web cache C for all requests (assume that the Web cache resolves the name for any Web server and that the client is configured with the IP address of the cache). All wires/links are Ethernet segments.

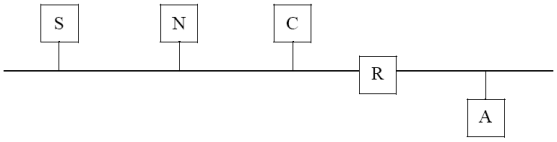


Figure 3: Network topology

Assume the following:

a. All the caches (DNS, Web, persistent connection) are empty

b. http://S/index.html fits in a single packet

c. Persistent HTTP connections are used among A, C, and S

The user on machine A requests the web page http://S/index.html. The table below shows some of the messages that are exchanged in serving the request (NOTE: this is not necessarily an exhaustive list of all exchanged packets!). In addition, there are several bogus packets that were never sent/received. The packets are not listed in temporal order. Fill in the order column to indicate the order in which each packet was sent/received (1 =first, 2 =second, etc.). Place an X in the order column if the packet is bogus. Fill in the Transport Protocol column.

ID	Source	Destination	Source Port	Destination Port	Transport Protocol	Contents	Order
1	C	DNS root		DNS		query for S	
2	A	C		Web Cache		get http://S/index.html	
3	N	DNS root		DNS		query for S	
4	A	R		DNS		query for S	
5	S	A	HTTP			index.html	
6	C	S		HTTP		get index.html	
7	C	A	Web Cache			index.html	
8	A	S		HTTP		get http://S/index.html	
9	N	C	DNS			address for S	
10	S	C	HTTP			index.html	

Table 1: HTTP request

ID	Source	Destination	Source Port	Destination Port	Transport Protocol	Contents	Order
1	C	DNS root		DNS	UDP	query for S	X
2	A	C		Web Cache	TCP	get http://S/index.html	1
3	N	DNS root		DNS	UDP	query for S	2
4	A	R		DNS	UDP	query for S	X
5	S	A	HTTP		TCP	index.html	X
6	C	S		HTTP	TCP	get index.html	4
7	C	A	Web Cache		TCP	index.html	6
8	A	S		HTTP	TCP	get http://S/index.html	X
9	N	C	DNS		UDP	address for S	3
10	S	C	HTTP		TCP	index.html	5

Table 2: HTTP request - solution

Question 6: Assume that the client A has no Web or DNS cache and that cache C has no DNS cache. However, all other cacheable information is cached. On a subsequent request for http://S/index.html, which of the messages in Table 1 would be eliminated (use the ID to name the messages)?

Depending on caching parameters that determine whether and for how long the index.html page could have been cached at the proxy C several solutions are possible. Full number of points is given for any of the following answers:

- *If the index.html page was not cached at the proxy C: only message* 3 *would be eliminated;*
- *If the index.html page was cached at the proxy C and it was fresh when the user A requested it: All messages except* 2 *and* 7 *would be eliminated, i.e. messages* $3, 6, 9, 10$*;*
- *If the index.html page was cached at the proxy C and it was fresh when the user A requested it and the proxy C had to resolve the IP address of the S: messages* $3, 6, 10$ *would be eliminated;*
- *If the proxy C had to send a conditional GET to the web server S and the webpage had been modified, then the web server replies with the new version of the index.html: only message* 3 *would be eliminated;*
- *If the proxy C had to send a conditional GET to the web server S and the webpage had not been modified: messages* $3, 10$ *would be eliminated.*

In addition, messages $1, 4, 5, 8$ *are bogus and are never sent/received.*

6 Transport Layer (8 points)

Consider the Go-Back-N protocol with a sender window size of 3 and a sequence number range of 1024. Suppose that at time t, the next in-order packet that the receiver is expecting has a sequence number of k. First, assume that the medium does not reorder messages. Answer the following questions.

Question 1: What are the possible sets of sequence numbers inside the sender's window at time t? Justify your answer.

If the receiver is expecting k then it has received k-1 and ACKed it.

- *if ACK's for $k-3$, $k-2$, $k-1$ sent but not yet received, sender's window is $k-3$, $k-2$, $k-1$. Note these ACK's could be lost.*
- *since $k-1$ received at receiver, sender's window must be bigger than $k-4$*
- *if all ACK's up to $k-1$ received at sender, senders window is $(k, k+1, k+2)$*

Thus the range of possible values is $k-3$ to $k+2$.

Question 2: What are all possible values of the ACK field in messages currently propagating back to the sender at time t: Justify your answer.

The sender has sent packets [k-3, k-1], it must be the case that the sender has already received an ACK for k-4. By arguments above, $k-3$, $k-2$, $k-1$ ACK's could be in the medium, on their way to sender. ACK for k not in medium since receiver awaits k.
NOTE: If in this problem we consider the premature timeout for packet $k-4$, the ACK for packet $k-4$ could also be in the medium and arrive again before the ACK for packet $k-3$. Note that the sender has already received the ACK for $k-4$ and the medium does not reorder messages. Consequently the premature ACKs for packets sent before $k-4$ cannot arrive after the ACK for $k-4$.

Now consider the Go-Back-N protocol, but suppose that the channel can reorder messages such that when the n^{th} packet is sent, it can only be bypassed (i.e., reordered) by the packet the was sent immediately before it.

Question 3: Are timers still needed to insure correct operation of the Go-Back-N protocol? Justify your answer.

Yes, timers are still needed. They have nothing to do with channel reordering. They are used to recover from lost packets.

Question 4: Will the standard Go-Back-N protocol still work in this case (i.e., with reordering only one packet), for a sequence number space of 1024 and a window size of 3? Justify your answer.

GBN will still work. The receiver can treat a reordered packet as out of order and ignore. As long as ACK's are cumulative, reordered ACK's are OK also. With Go-Back-N the sequence number space must be at least one larger than the window size to avoid wraparound problems. With reordering by at most one, sequence number space must be at least two bigger than the window size, which is satisfied here.