Computer Networks - Midterm

Prof. J.-P. Hubaux and Dr. M. Jadliwala

November 8, 2011

Duration: 1:45 hours, closed book.

Please write your answers on these sheets in a ***readable*** way.
Poorly written answers will ***not*** be corrected.
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 16 pages.

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)

1 Short questions (5 points)

For each question, please circle a single best answer.

1. Sort the following physical media based on their maximum throughput, from lowest to highest.
   (a) Twisted Pair Category 5, IEEE 802.11g, Fiber Optic
   (b) Twisted Pair Category 5, Fiber Optic, IEEE 802.11g
   (c) IEEE 802.11g, Twisted Pair Category 5, Fiber Optic *CORRECT*
   (d) Fiber Optic, IEEE 802.11g, Twisted Pair Category 5

2. Packet-switching versus circuit-switching: Which of the following is correct?
   (a) Packet-switching is more flexible, circuit-switching is more robust to link failures
   (b) Packet-switching is more secure, circuit-switching is more efficient
   (c) Packet-switching and circuit-switching offer the same bandwidth guarantees
   (d) Packet-switching allows more users to access the network, circuit-switching provides quality-of-service guarantees *CORRECT*

3. Consider the following Java code:

```
1:  try {
2:      ServerSocket ss = new ServerSocket(2011);
3:      Socket s1 = ss.accept();
4:      Socket s2 = ss.accept();
5:      DataInputStream is = new DataInputStream(s1.getInputStream());
6:      DataOutputStream os = new DataOutputStream(s2.getOutputStream());
7:      while (true) {
8:          os.writeByte(is.readByte());
9:      }
10:  } catch (Exception e) {
11:     // crash!
12:  }
```

   What does this code do?
   (a) Does not compile, because the `ServerSocket` constructor takes no parameters.
   (b) Always crashes in line 4, because you can only call `accept()` once.
   (c) Creates an "echo" program that responds to the client with the exact same bytes he sent.
   (d) Creates a "relay" program that sends the bytes received from one client to another client. *CORRECT*

4. An organization's Web server and mail server:
   (a) can have exactly the same alias for a hostname (for example, foo.com). *CORRECT*
   (b) cannot have exactly the same alias for a hostname (for example, foo.com).
   (c) must run on the same machine.
   (d) can run on the same machine only if they share one DNS record.

5. Assume that the following lines are inserted at the DNS server:

   (midterm.edu, service1.edu, CNAME)
   (midterm.edu, service2.edu, MX)
   (service1.edu, 8.11.11.101, A)
   (service2.edu, 8.11.11.104, A)

   As a consequence, when you open midterm.edu in your browser:

   (a) the request is served by the server with IP 8.11.11.101. *CORRECT*
   (b) the request is served by the server with IP 8.11.11.104.
   (c) the request is served by the server with IP 8.11.11.101 or 8.11.11.104, depending on the current server load.
   (d) the DNS server cannot hold two records for midterm.edu.
   (e) none of the above.

6. Which of the following regarding UDP is true?

   (a) UDP would work perfectly for applications such as Email and File Transfer.
   (b) UDP segments have a smaller packet header size as compared to TCP segments. *CORRECT*
   (c) UDP requires an explicit connection establishment using three-way handshake protocol.
   (d) An application using UDP can never have reliable data transfer.

7. In a Go-Back-N (GBN) protocol, if the last correctly received (and delivered to the upper layer) segment has sequence number 5 and the receiver next receives a segment with sequence number 7, it does the following:

   (a) Buffers segment 7 and sends an ACK for segment 5.
   (b) Buffers segment 7 and sends an ACK for segment 7.
   (c) Discards segment 7 without sending any ACK.
   (d) Discards segment 7 and sends an ACK for segment 5. *CORRECT*

8. In TCP, a sender performs a fast retransmit (re-transmitting a lost segment before the timeout event) in the following event:

   (a) It received three duplicate ACKs for the same data. *CORRECT*
   (b) The RTT (Round Trip Time) of the last ACK was greater than the average RTT.
   (c) Data in the sender's TCP buffer exceeds a threshold value.
   (d) None of the above.

9. Suppose within your Web browser you click on a link to obtain a Web page. The IP address for the associated URL is not cached in your local host, so a DNS lookup is necessary to obtain the IP address. Suppose that n DNS servers are visited before your host received the IP address from DNS; the successive visits incur an RTT of $RTT_1, ..., RTT_n$. Further suppose that the Web page associated with the link contains exactly one object, consisting of a small amount of HTML text. Let RTT_0 denote the RTT between the local host and the server containing the object. Assuming zero transmission time of the object, how much time elapses from when the client clicks on the link until the client receives the object?

   (a) $RTT_1 + RTT_2 + ... + RTT_n + RTT_0$
   (b) $RTT_1 + RTT_2 + ... + RTT_n + 2RTT_0$ *CORRECT*
   (c) $2(RTT_1 + RTT_2 + ... + RTT_n) + RTT_0$
   (d) $2(RTT_1 + RTT_2 + ... + RTT_n + RTT_0)$

2 The Web (10 points)

A student sequentially visits two different websites that use the same advertisement provider to include online advertisements into the content of their webpages. The generated traffic was captured with Wireshark and simplified traces (trace 1 corresponding to website 1 and trace 2 to website 2) are given in Figure 1, Figure 2, Figure 3, Figure 4, Figure 5 and Figure 6. By analyzing the given traces, answer the following questions:

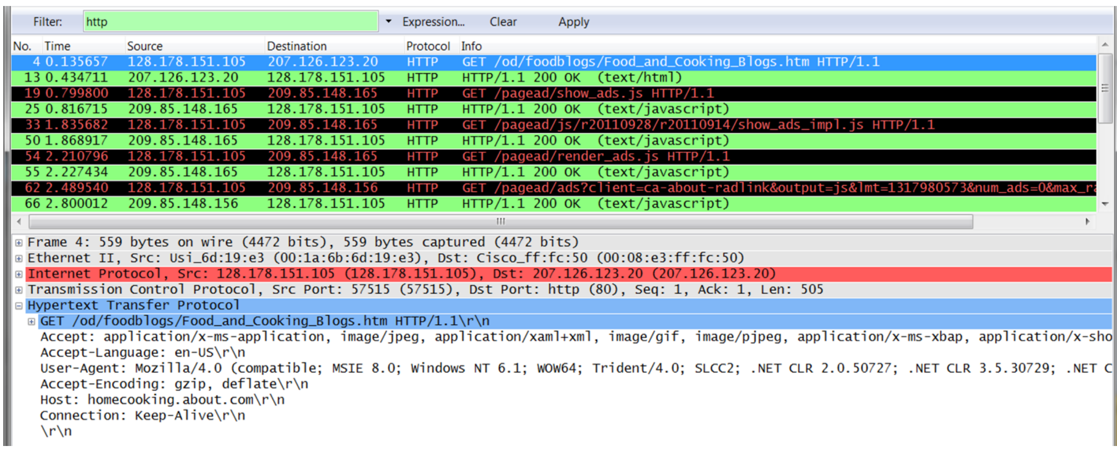

Figure 1: Wireshark trace 1 - The first HTTP request

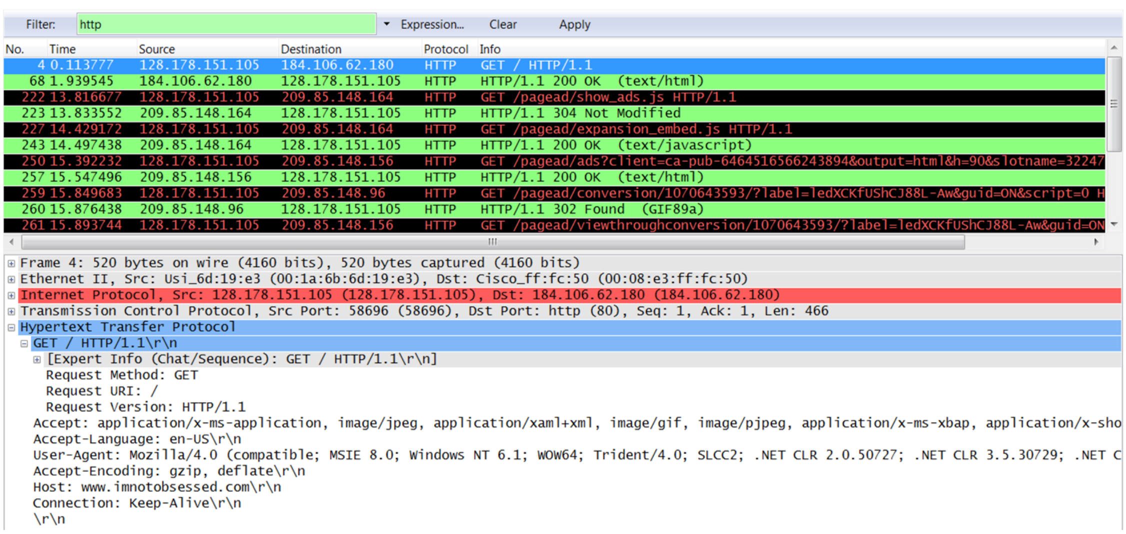

Figure 2: Wireshark trace 2 - The first HTTP request

Question 1: What are the *URLs* of the visited Web pages?

First: `homecooking.about.com/od/foodblogs/Food_and_Cooking_Blogs.htm`
Second: `www.imnotobsessed.com`

Question 2: What is the size of the first HTTP GET request in trace 1?
505B, noted in the Len field of the TCP packet details.

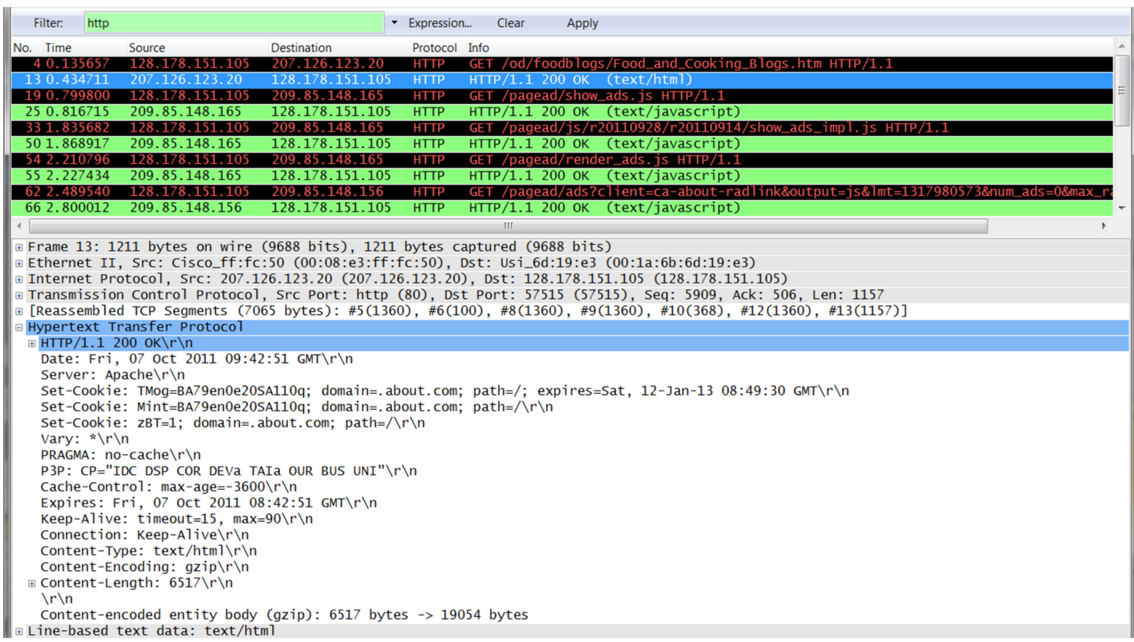

Figure 3: Wireshark trace 1 - The first HTTP response

Question 3: Is there a domain for which the first visited website can track the student's subsequent browsing behavior? Justify your answer.

Yes, on the domain `about.com`*. Set-Cookie headers in the first HTTP response message indicate that the first website has placed cookies in the student's browser for the domain* `about.com`*. Unless the student deletes cookies, for subsequent visits the website will recognize the student as a returning visitor.*

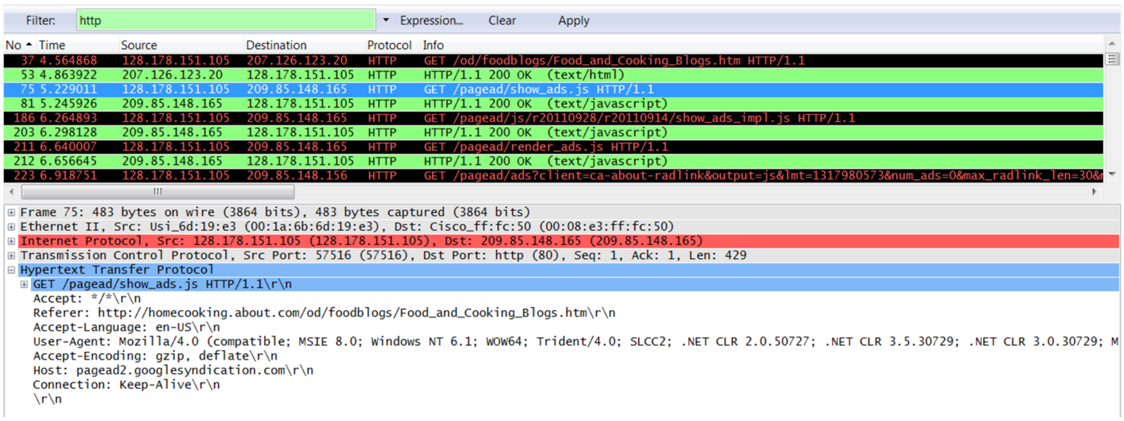

Figure 4: Wireshark trace 1- The second HTTP request

Question 4: Which mechanism is used to include ads into the content of visited Web pages? Which server is contacted for inclusion of ads? To which organization does that server belong? What is its IP address?

> *From the second HTTP GET request: Host:* `pagead2.googlesyndication.com`*. The server belongs to Google and the website includes Google ads via show_ads.js javascript. IP of the server is:* 209.85.148.165*.*

Question 5: When the student's browser communicates with the ad provider, does the ad provider know which webpage the student is visiting? Justify your answer.

> *Yes, based on the value of the Referer field (e.g., in the second HTTP GET request:* `http://homecooking.about.com/od/foodblogs/Food_and_Cooking_Blogs.htm`*).*

Question 6: Can the second website know that the student has previously visited the first website? Justify your answer.

> *No, because the student has independently entered the URL of the second website thus the referer field is empty. Also, the websites belong to different domains and cookies are not shared between these two websites.*

Question 7: Can the second website know that the student has previously been served ads from the same ad provider? Justify your answer.

> *No. For the second website to learn this information, the cookies for the domain of the ad provider would have to be shared with the second website, which is not the case (i.e., the domain of the second website is different from the domain of the ad provider).*

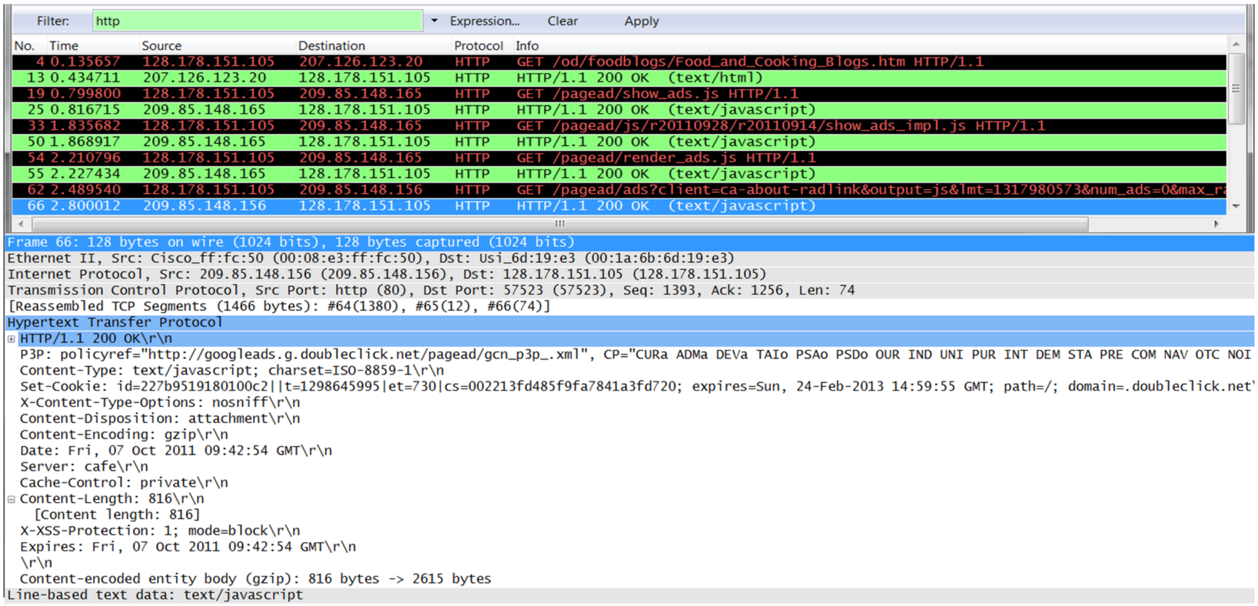

Figure 5: Wireshark trace 1 - The fifth HTTP response

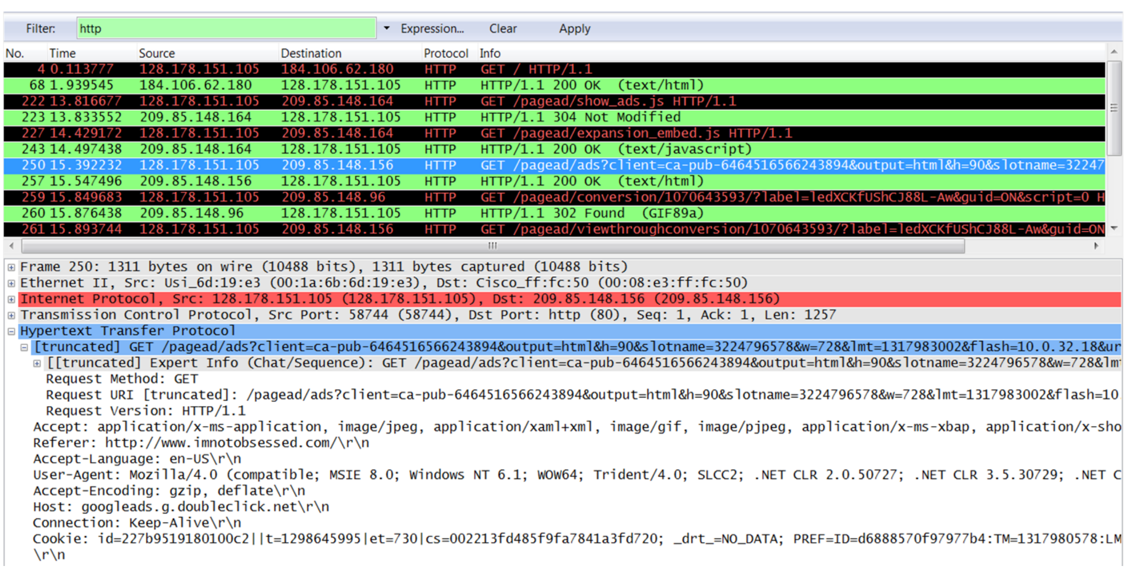

Figure 6: Wireshark trace 2 - The fourth HTTP request

Question 8: Can the ad provider know that the student has visited the first website before visiting the second website? Justify your answer.

Yes. In the fifth HTTP response in the first trace (packet 66*, Figure 5) the ad provider sets a cookie in the student's browser (Cookie id=227b9519180100c2 t=1298645995 et=730 cs=002213fd485f9fa7841a3fd720) for the domain* `.doubleclick.net`*. When the following requests are sent to the same domain, the browser includes this cookie (second trace, fourth HTTP GET request, packet* 250*, Figure 6). Thus, the ad provider "recognizes" the student and can link the requests from the first and second trace with the same student. Based on the Referer field of the requests the ad provider learns which websites the student has visited.*

Question 9: After visiting the first website, is there any domain at which the ad provider can track the student's subsequent browsing behavior? Justify your answer.

The ad provider has placed a cookie for the domain `.doubleclick.net` *in the student's browser. This means that the student's browser will send this cookie to doubleclick ad servers with each request to this domain. This enables the ad provider to link these requests with the student and track his behavior on this domain. However, since each of these requests contain Referer field that informs the ad provider which page the student is currently visiting, in practice the ad server will know all the pages that the student has browsed that include ads from this ad provider.*

Question 10: When visiting the second website, what does the code 304 of the second HTTP response message (packet 223, Figure 2) signify?

304 *code of the HTTP message means that the conditional HTTP GET was sent. Since the object has not been modified, it was not downloaded again from the server but from the cache. The object was cached after being downloaded when the student has visited the first Web page.*

3 Network Delays (7 points)

Consider a route in the Internet connecting hosts A and B. The route is composed of n links. The ith link has length d_i meters, propagation speed v_i m/s, and throughput r_i bps. Unless stated otherwise, assume that routers incur no processing or queueing delay. Assume that the routers work in a store-and-forward fashion: A packet has to be fully received before the router can start to retransmit it.

Question 1: What is the propagation delay from A to B for this route?

$$\sum_{i=1}^{n} \frac{d_i}{v_i}$$

Question 2: Assume that you send a packet of L bits from A to B. What is the transmission delay for link i?

$$\frac{L}{r_i}$$

Question 3: Give the total delay of transmitting an L bit packet from A to B over this route.

$$\sum_{i=1}^{n} (\frac{d_i}{v_i} + \frac{L}{r_i})$$

For the following questions, assume that you are logged in at host A. You can launch pings with a packet of chosen size, and obtain the round-trip time for each ping. Try not to launch excessive pings.

Question 4: Assume that all links have propagation speed $v_i = v$ (for this question only). Discover the total length of the route in meters. What pings do you launch and what computation do you perform? (Recall that we assume no processing or queueing delays, and try to be as precise as possible.)

t_1 = ping B with packet size L bits.
t_2 = ping B with packet size $2L$ bits.
From Question 3, we know that $t_1 = 2\sum_{i=1}^{n}(\frac{d_i}{v} + \frac{L}{r_i})$ and $t_2 = 2\sum_{i=1}^{n}(\frac{d_i}{v} + \frac{2L}{r_i})$
Hence: $d = 0.5(2t_1 - t_2)v$

Question 5: Assume that all n links have throughput $r_i = r$. Discover the value of r. What pings do you launch and what computation do you perform? (Recall that we assume no processing or queueing delays.)

t_1 = ping B with packet size L bits.
t_2 = ping B with packet size $2L$ bits.
From Question 3, we know that $t_1 = 2(\sum_{i=1}^{n} \frac{d_i}{v_i} + \frac{Ln}{r})$ and $t_2 = 2(\sum_{i=1}^{n} \frac{d_i}{v_i} + \frac{2Ln}{r})$
Hence: $r = \frac{nL}{0.5(t_2 - t_1)}$

Question 6: Assume that link j has a significantly lower throughput than all the other links. Discover the approximate throughput of this link (r_j). What pings do you launch and what computation do you perform? (Recall that we assume no processing or queueing delays.)

Is your approximation a lower-bound or an upper-bound? What is the maximum error (relative or absolute) that you can make with this approximation when trying to estimate the throughput of the link with lowest throughput?

t_1 = ping B with packet size L bits.
t_2 = ping B with packet size $2L$ bits.
Approximation: $r_j = \frac{L}{0.5(t_2 - t_1)}$
This is a lower-bound. The relative error is

$$\frac{\sum_{i \neq j} r_i^{-1}}{\sum_{i=1}^{n} r_i^{-1}}$$

which takes the maximal value $\frac{n-1}{n}$ when all links have the same throughput.

Question 7: Assume now that queueing delays cannot be neglected. Discover the (approximate) average queueing delay for this route. What pings do you launch and what computation do you perform? (We assume no processing delays.)

We launch N pings with size L and get RTTs t_k
We approximate the queueing delay $t_{queue} = 0.5(mean\{t_k\} - min\{t_k\})$

4 Peer-to-Peer (4 points)

Consider distributing a file of F bits to N peers using a client-server architecture. Assume a fluid model where the server can simultaneously transmit to multiple peers, transmitting to each peer at different rates, as long as the combined rate does not exceed the upload rate u_s of the server's access link. Denote the download rate of the i-th client's access link by d_i and d_{min} is the download rate of the client with the lowest download rate, i.e., $d_{min} = \{d_1, d_2, \cdots, d_N\}$.

Question 1: Suppose that $u_s/N \leq d_{min}$. Specify a distribution scheme that has a distribution time of NF/u_s.

Consider a distribution scheme in which the server sends the file to each client, in parallel, at a rate of u_s/N. Note that this rate is less than each of the client's download rate, since by assumption $us/N \leq d_{min}$. Thus each client can also receive at rate u_s/N. Since each client receives at rate u_s/N, the time for each client to receive the entire file is $F/(u_s/N) = NF/u_s$. Since all the clients receive the file in NF/u_s, the overall distribution time is also NF/u_s.

Question 2: Suppose that $u_s/N \geq d_{min}$. Specify a distribution scheme that has a distribution time of F/d_{min}.

Consider a distribution scheme in which the server sends the file to each client, in parallel, at a rate of d_{min}. Note that the aggregate rate, $N \cdot d_{min}$, is less than the server's link rate u_s, since by assumption $u_s/N \geq d_{min}$. Since each client receives at rate d_{min}, the time for each client to receive the entire file is F/d_{min}. Since all the clients receive the file in this time, the overall distribution time is also F/d_{min}.

5 Transport Layer Error Recovery (8 points)

Consider a Selective Repeat (SR) error recovery protocol with a window size N of 3 and valid sequence numbers ranging from 0 to 8. Assume that there is no re-ordering of segments possible on the channel, but segments can be lost. Answer the following questions:

Question 1: If the receiver is expecting segment with sequence number 3 (i.e., $recv\_base = 3$) at time instant t and if it receives segments with sequence numbers 2, 0 and 1 (in that order), which segments have been lost, assuming that any segment in the network can be lost at most once? What is the response of the receiver on receiving each of these segments 2, 0 and 1? You are expected to answer this question by completing the sequence diagram below by clearing showing the segment exchanges between the sender and the receiver, including the lost segments and ACKs, the timeout events and the sender/receiver windows when segments (including ACKs) are received by the sender/receiver.

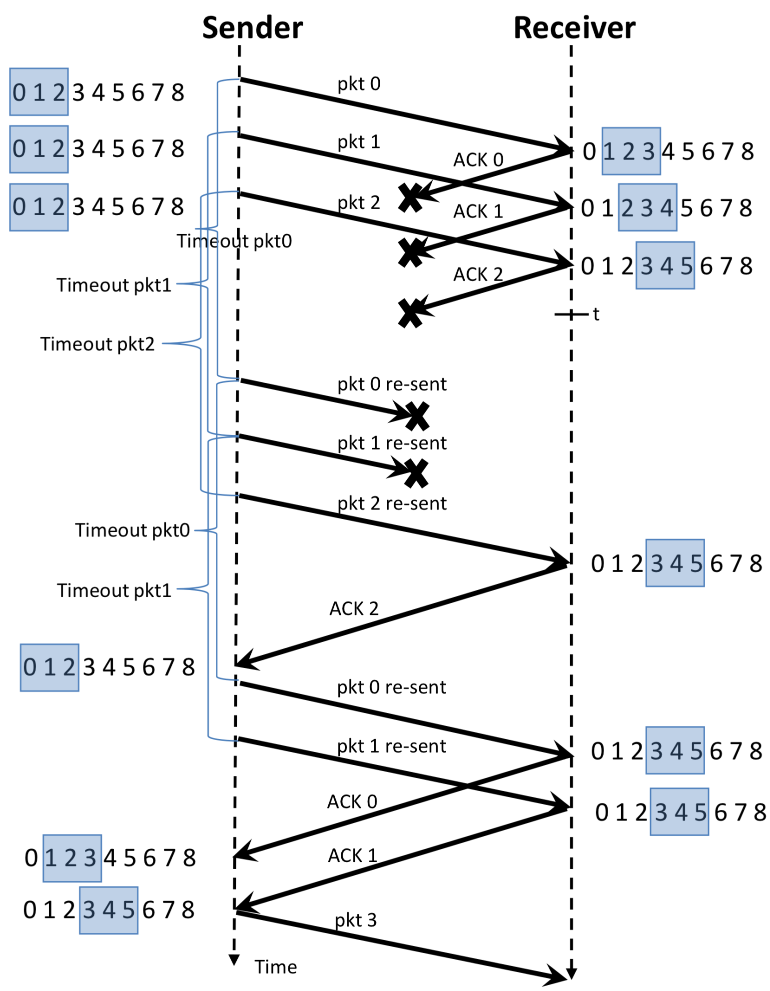

Figure 7: Selective Repeat (SR) Protocol (Solution to Question 1).

Question 2: Assume that the window size (N) is the same as in the previous question, i.e., 3, but the sequence number range is now only from 0 to 3. The receiver is expecting segment with sequence number 3 (i.e., $recv\_base = 3$) at time t, but it receives a segment with sequence number 0 instead. Does the receiver treat this segment as a new segment or a re-transmitted segment? Please justify your answer.

The receiver cannot decide if this segment is a new segment or a re-transmission of an old segment. There are two situations when this can happen. The sender can send a segment with sequence number 0 because all the previous acknowledgments (from the receiver) were received correctly and is thus moving its window from 0,1,2 to 3,0,1. The sender can thus send a new segment with sequence number 0 in this window. Another situation in which the sender sends a segment with sequence number 0 is when it does not receive the acknowledgement of the segment 0 sent when the window was 0,1,2. In the latter case, this segment is a re-transmission of the previously sent segment. The receiver is not able to distinguish between these two cases. This problem arises because the size of the congestion window in the protocol is too large as compared to the range of sequence numbers.

6 TCP Connection Management (6 points)

A client initiates a TCP connection with a server, exchanges some data and then closes the connection. Assume that the communication channel can lose segments but cannot re-order segments. Answer the following questions:

Question 1: Figure 8 shows the TCP three-way handshake protocol used by the client to initiate a TCP connection with the server. Complete Figure 8 by filling in the missing SYN flags, SEQ numbers and ACK numbers of the segments exchanged between the client and the server.

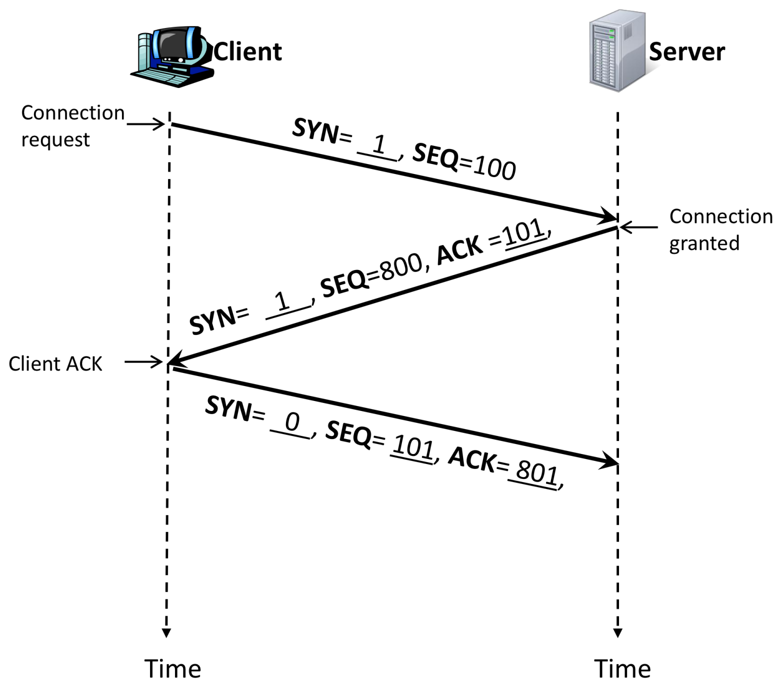

Figure 8: TCP connection setup: Three-way handshake protocol (Solution to Question 1).

Question 2: Assume that the server reserves memory for TCP buffers at the time instant the connection is granted (before sending the ACK). If the server has limited memory, how can a malicious client run the server out of TCP buffer memory using the TCP three-way handshake protocol? Explain.

The malicious client can send a large number of TCP SYN segments (on different port addresses), without completing the third handshake step. The attack can be amplified by sending the SYN segments from multiple sources. Soon the server will run out of available (free) TCP buffer space and will not be able to accept additional TCP requests. Such a type of Denial of Service (DoS) attack is called a ***SYN flood attack****.*

Question 3: Figure 9 shows the TCP connection closing protocol (initiated by the client). During this connection closing, what will happen if the last ACK is lost? Show the next steps of the protocol by drawing on Figure 9. You can assume that no more segments will be lost.

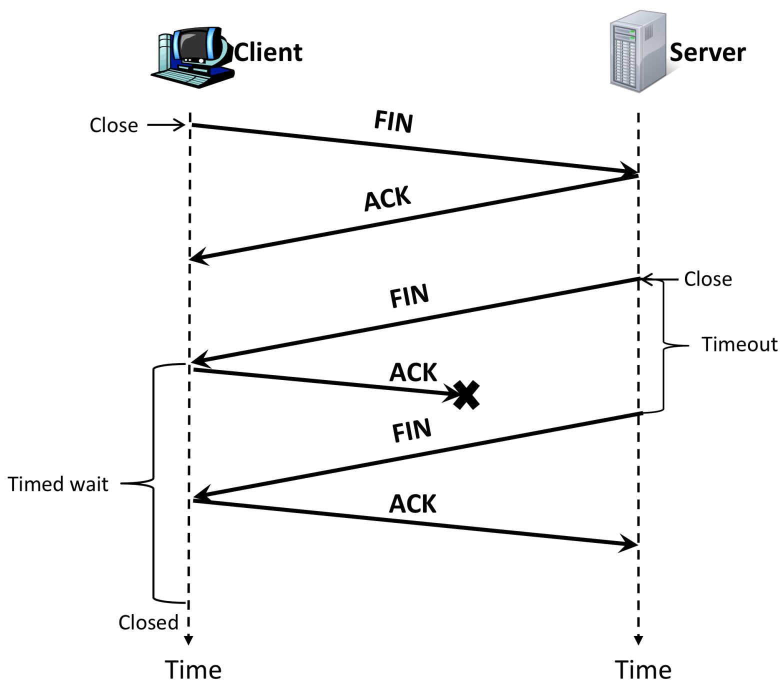

Figure 9: TCP connection close (Solution to Question 3).

Question 4: What behavior do you expect if client closes connection immediately after sending the final ACK, i.e., the *Timed wait* time interval is zero? Is this good or bad? Explain why?

If the client immediately closes connection after sending the final ACK and destroys the client socket on the associated port, and the client ACK is lost (as shown in Figure 9), then the server will never know that the client closed the TCP connection. Further re-transmissions of the FIN from the server (after time-out) will be immediately dropped by the client because there is no longer a TCP socket running on that port. This is not good. The server will keep transmitting FINs expecting the client to reply with ACK. Moreover, if there is piggybacked data with the server FIN, that data will be lost.