CSCI4211: Introduction to Computer Networks
Fall 2017
Homework Assignment 1

Due 11:59pm Friday October 6

Instructions:

1. Please submit your homework using the on-line electronic submission system (via Moodle) – click on the"Submit" link on the class website.

   In case you could not use the on-line electronic submission system, please hand in your homework to the instructor or TAs on the due day; or slide it under the instructor's office door (Keller Hall 6-187). Please email csci4211-help@cs.umn.edu to let us know that you have handed in a hard-copy of your homework immediately afterwards. (*Make sure that you also make and retain a copy of your homework!*)

   Please make sure that you include your name and student id in your submission, and retain a copy of your submission!

2. There are **six** questions in total. The number of points for each question is given in parentheses. There are 125 points in total (plus *15 bonus points*). An *estimated* time for answering each question is also given in parentheses. This is just a guideline, you may take less or more time on each problem.

3. Partial credit is possible for an answer. Please try to be as concise and make your homework as neat as possible. We *must* be able to read your handwriting in order to be able to grade your homework.

4. Enjoy!

---

1. Short Questions and Answers: (24 points; 20 minutes)
(*Two or three sentences would generally suffice.*)

a. (4 points) Name one key advantage of packet switching.

b. (4 points) Using one or two sentences, define what is a protocol.

c. (4 points) Using one or two sentences, describe what is transmission delay?

d. (4 points) Using one or two sentences, describe what is propagation delay?

e. (4 points) Using a couple of sentences, discuss why we need both source and destination port numbers in transport layer protocols such as TCP or UDP.

f. (4 points) Name one or two major advantages of a layered network architecture.

2. Statistical Multiplexing: Circuit Switching vs. Packet Switching (26 points; 25 minutes)

Do Problem P8, Chapter 1 (page 71) in the textbook (the 7th edition).

In case you do not have the current version of the textbook, the problem is reproduced below for you.

Suppose users share a 3 Mbps link. Also suppose each user requires 150 kbps when transmitting, but each users transmits only 10 percent of the time. (See the discussion of packet switching versus circuit switching in Section 1.3.)

(a) (8 points) When circuit switching is used, how many users can be supported?

(b) (5 points) For the remainder of this problem, suppose that packet switching is used. Find the probability a given user is transmitting.

(c) (8 points) Suppose there are 120 users. Find the probability that at any given time, exactly n users are transmitting simultaneously. (Hint: Think *Binomial* distribution – *Instructor's Note: writing down a formula suffices!*)

(d) (5 points) Find the probability that there are 21 or more users transmitting simultaneously.

3. Network delay for circuit switching vs. packet switching (25 points; 20 minutes)

Minneapolis and *New York* are two end hosts on the Internet. Consider the following topology, link capacity and other specifications:

- *New York*, the destination host, is 5 hops away from *Minneapolis*, the source, *i.e.* there are 4 intermediate routers: *Minneapolis – R1 – R2 – R3 – R4 – New York*.
- The distance between any two adjacent nodes is 250 kilometers (km).
- The signal propagation speed is 2.5×10^5 km per second.
- The message size is 2Mbits ($1\text{M} = 10^6$)
- The maximum packet size is 100 Kbits ($1\text{K} = 10^3$). The header size is negligible. Note for packet switching the message should be divided to packets, each of which cannot exceed the maximum packet size.
- The transmission speed of each link is 100Mbps).
- The circuit setup time is 0.5 second for the case of circuit switching.
- The processing time for routing & forwarding decision at each node can be ignored.

Please answer the following questions. (Note please make sure to illustrate your calculations clearly; you may receive partial credits even if your final answer is incorrect.)

(a) (10 points) The end to end delay in delivering the message using circuit switching.

(b) (10 points) The end to end delay in delivering the message using packet switching.

(c) (5 points) Will the delays calculated in parts (a) and (b) alter if *Minneapolis* was the destination and *New York* the source? Justify your answer.

4. Name Resolution, DNS and DHT (10 points + 10 bonus points; 15 minutes)

a. (10 points) Do Problem P7, Chapter 2 (page 175) in the textbook (the 7th edition).

In case you do not have the current version of the textbook, the problem is reproduced below for you.

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 receives the IP address from DNS; the successive visits incur an RTT (round-trip-time) of $RTT_1, \ldots, RTT_n$. 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?

b. (*Optional: 15 points*) Briefly discuss the pros and cons of using *dynamic hash table* (DHT) techniques to implement the domain name system instead of the hierarchically distributed mechanism used in today's DNS.

5. Wireshark Hands-on Practice: HTTP (15 points total; 15-20 minutes)
(Approximate time excludes Wireshark set-up time, learning the basics of how to use Wireshark, etc.)

Wireshark is a free network protocol analyzer that runs on Windows, Linux/Unix, and Mac computers (see pp. 78 – 79 of the textbook for a brief description of Wireshark). Its an ideal packet analyzer – it is stable, has a large user base and well-documented support that includes:

1. A user-guide (`http://www.wireshark.org/docs/wsug_html_chunked/`),
2. Man pages (`http://www.wireshark.org/docs/man-pages/`),
3. A detailed FAQ (`http://www.wireshark.org/faq.html`),
4. You can search "wireshark tutorial" on YouTube to find a number of tutorial videos (e.g., "Tutorial using Wireshark" by *3dmasters* at `http://www.youtube.com/watch?v=y-4UQSXkqig` is a good one), or google "Wireshark Tutorial" to find some tutorials in pdf. (We have some links on the class website also, click "Useful References" on the left panel of the class website.)

The software, for all platforms, can be downloaded free from
`http://www.wireshark.org/download.html`

Please note that you need *root/admin* permissions for installing and using Ethereal/Wireshark. You should install it on your own laptop, as opposed to, say, on a CSE lab machine.

Lets start our exploration of HTTP by downloading a very simple HTML file - one that is very short, and contains no embedded objects. Please do the following:

1. Start up your web browser.
2. Start up the Wireshark packet sniffer. Enter "http" (just the letters, not the quotation marks) in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet-listing window.
3. Wait a bit more than one minute, and then begin the Wireshark packet capture.
4. Enter the following to your browser
   `http://www-users.cselabs.umn.edu/classes/Fall-2017/csci4211/Assignments/HTTP-1.htm`
5. Your browser should display the very simple, one-line HTML file.

6. Stop Wireshark packet capture.

The example in Figure1 shows in the packet-listing window that two HTTP messages were captured: the GET message and the response message from the server to your browser. The packet-contents window shows details of the selected message (in this case the HTTP GET message, which is highlighted in the packet-listing window). Recall that since the HTTP message was carried inside a TCP segment, which was carried inside an IP datagram, which was carried within an Ethernet frame, Wireshark displays the Frame, Ethernet, IP, and TCP packet information as well. We want to minimize the amount of non-HTTP data.

Now let's see what happens when we download a long HTML file. Please do the following:

1. Start up your web browser, and make sure your browsers cache is cleared, as discussed above.
2. Start up the Wireshark packet sniffer.
3. Enter the following URL into your web browser.
   `http://www-users.cselabs.umn.edu/classes/Fall-2017/csci4211/Assignments/HTTP-2.htm`
4. Your browser should display the rather lengthy US Bill of Rights.
5. Stop Wireshark packet capture, and enter "http in the display-filter-specification window, so that only captured HTTP messages will be displayed.

In the packet-listing window, you should see your HTTP GET message, followed by a multiple-packet response to your HTTP GET request. This multiple-packet response deserves a bit of explanation. Recall from Section 2.2 (see Figure 2.9 in the textbook) that the HTTP response message consists of a status line, followed by header lines, followed by a blank line, followed by the entity body. In the case of our HTTP GET, the entity body in the response is the entire requested HTML file. In our case here, the HTML file is rather long, and at 4500 bytes is too large to fit in one TCP packet. The single HTTP response message is thus broken into several pieces by TCP, with each piece being contained within a separate TCP segment. Each TCP segment is recorded as a separate packet by Wireshark, and the fact that the single HTTP response was fragmented across multiple TCP packets is indicated by the "Continuation phrase displayed by Wireshark. We stress here that there is no "Continuation message in HTTP!

Please answer the following questions. (If possible, please include a print-out or embed the screenshot you have captured (similar to the Figure 1) (or simply take a picture using your mobile phone and embed the picture) into your solution document, and use it as the basis for your answer to the following questions.)

1. Is your browser running HTTP version 1.0 or 1.1 (or HTTP/2)? What version of HTTP is the server running?

2. What is the IP address of your computer and that of the web server?

3. How many HTTP GET request messages were sent by your browser?

4. How many data-containing TCP segments were needed to carry this single HTTP response?

5. What is the status code and phrase associated with the response to the HTTP GET request?

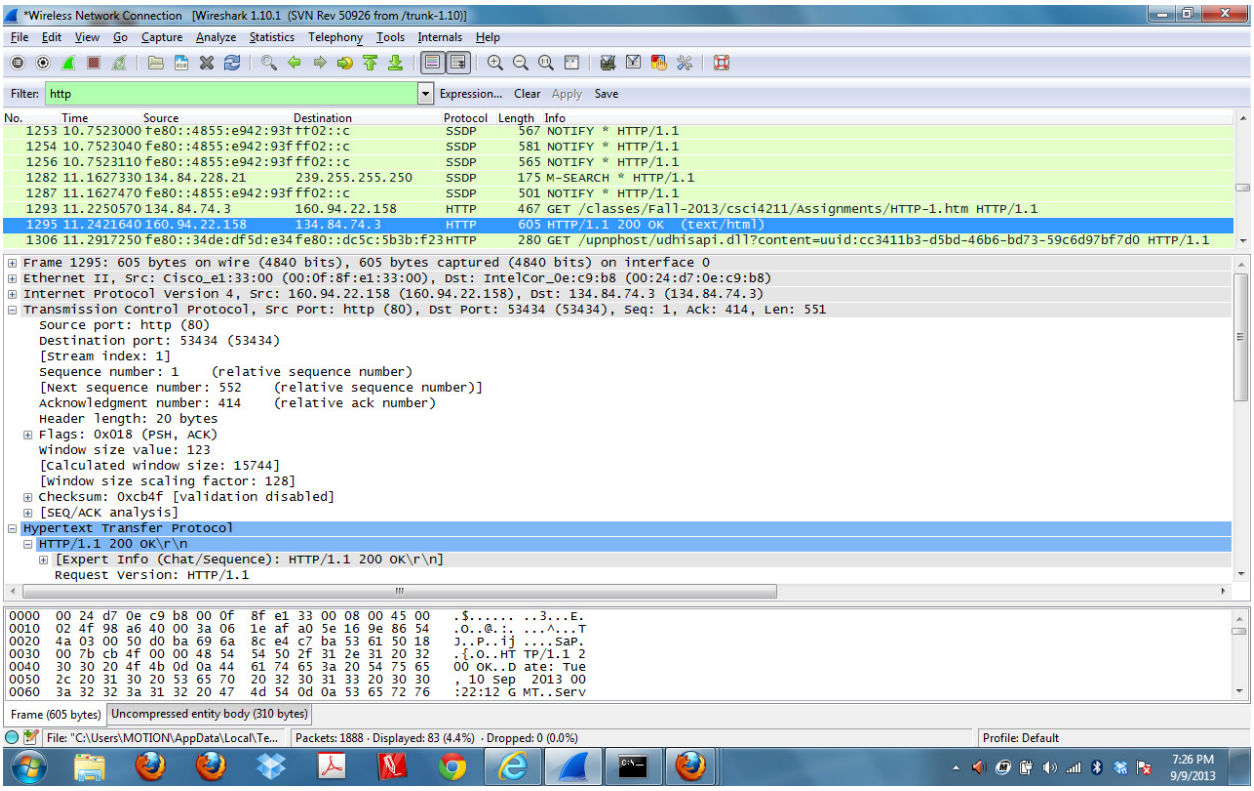

Figure 1: Wireshark Screenshot.

6. **TCP Connection Management** (25 points total. Approx. 20 minutes)

The following figure shows the control messages sent among the client and the server under normal operations using the *three-way handshake* protocol. (Note: in `SYNACK(y,x)` `and ACK(x,y)`, the first number is the sequence number of the message, the second number is the acknowledgment number, i.e., the sequence number of the message being acknowledged.)

a. (7 points) Consider the following scenario (see Figure 3) where the `SYNACK(y,x)` message sent by the server is lost during the transmission. What will happen at either the client

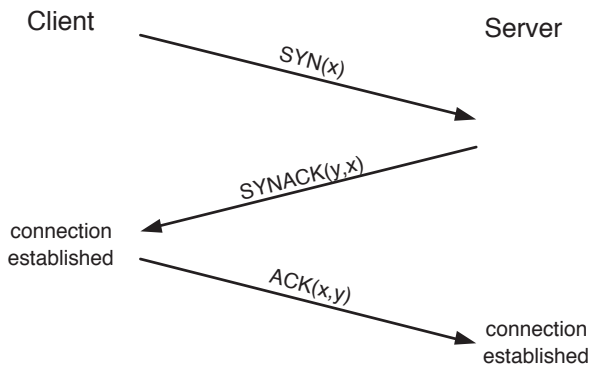

Figure 2: TCP 3-way handshake.

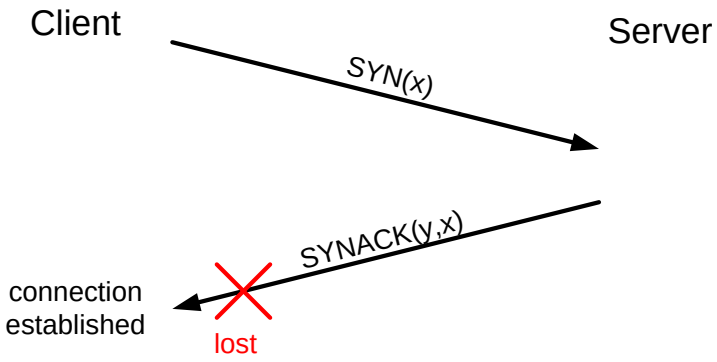

Figure 3: Figure for Question 6a.

or the server side?

b. (7 points) Suppose that by now the connection in **a.** (where the client used the initial sequence no. `x`, and the server used the initial sequence no. `y`) has been closed. An old, duplicate message `SYNACK(y,x)` now pops up at the client side (see Figure 4). First, can this scenario happen at all? Second, in response to this `SYNACK(y,x)` message, what will the **client** do? Briefly explain your answers to both questions.

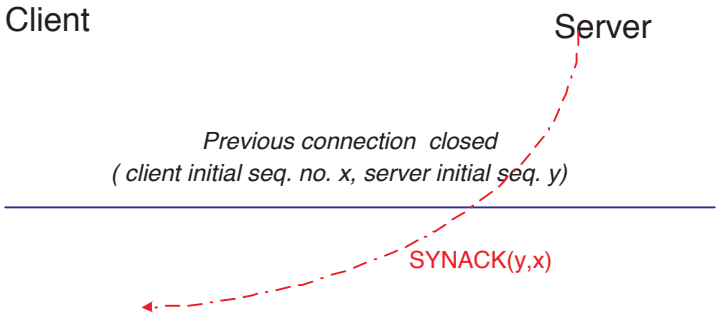

Figure 4: Figure for Question 6b.