CSCI4211: Introduction to Computer Networks
Spring 2020
Homework Assignment 1

Due 11:55pm Friday February 21

Instructions:

1. Please submit your homework using the on-line electronic submission system (via Canvas)

   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 150 points in total. 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) Using one or two sentences, describe what is propagation delay?

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

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

d. (4 points) Name one key advantage of a *layered* network architecture.

e. (4 points) **True or False:** That HTTP/1.1 by default uses a *persistent* TCP connection means that HTTP/1.1 is a *stateful* application-layer protocol, i.e., the HTTP/1.1 client and server will maintain some states to remember the previous HTTP requests/replies. [Please briefly justify your choice.]

f. (4 points) **True or False:** UDP data segments ("packets") do not need to carry source and destination port numbers because it does *not* provide reliable data transfer service. [Please briefly justify your choice.]

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

Suppose that many users share a 200Mbps bottleneck link. Also assume that each user needs 1Mbps when transmitting data, but each user will transmit randomly only 10% of time. Please answer the following questions. [Note: please make sure to illustrate your reasoning and derivations clearly; you may receive partial credits even if your final answer is incorrect.]

(a) (7 points) When circuit switching is used, how many users can be supported simultaneously?

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

(c) (7 points) Suppose that there are 300 users. What is the probability that at any given time, exactly half of the users are transmitting simultaneously? [Hint: Use the *binomial* distribution. Note that writing down a formula suffices – in other words, you don't need to carry out the calculation!]

(d) (7 points) Write down the probability that the network (the link) is *congested*. [Hint: Recall that a network (or a link) is congested when the demand (i.e., the amount of data transmitted) exceeds the capacity of the (bottleneck) link. Again a formula suffices, no need to carry out the calculation.]

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

Do Problem P31, Chapter 1 (page 76) 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.]

P31. In modern packet-switched networks, including the Internet, the source host segments long, application-layer messages (for example, an image or a music file) into smaller packets and sends the packets into the network. The receiver then reassembles the packets back into the original message. We refer to this process as *message segmentation*. Figure 1.27 illustrates the end-to-end transport of a message with and without message segmentation. Consider a message that is $8 \cdot 10^6$ bits long that is to be sent from source to destination in Figure 1.27. Suppose each link in the figure is 2 Mbps. Ignore propagation, queuing, and processing delays.

a. Consider sending the message from source to destination *without* message segmentation. How long does it take to move the message from the source host to the first packet switch? Keeping in mind that each switch uses store-and-forward packet switching, what is the total time to move the message from source host to destination host?

b. Now suppose that the message is segmented into 800 packets, with each packet being 10,000 bits long. How long does it take to move the first packet from source host to the first switch? When the first packet is being sent from the first switch to the second switch, the second packet is being sent from the source host to the first switch. At what time will the second packet be fully received at the first switch?

c. How long does it take to move the file from source host to destination host when message segmentation is used? Compare this result with your answer in part (a) and comment.

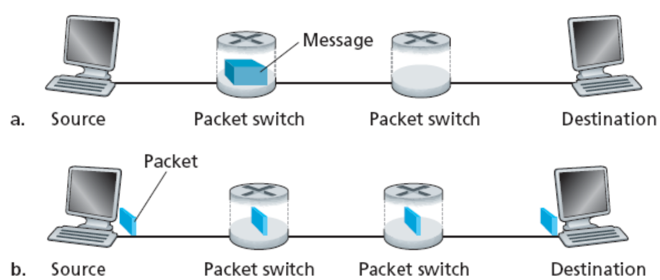


Figure 1.27 ♦ End-to-end message transport: (a) without message segmentation; (b) with message segmentation

d. In addition to reducing delay, what are reasons to use message segmentation?

e. Discuss the drawbacks of message segmentation.

4. Name Resolution, DNS and HTTP (24 points; 20 minutes)

1. (12 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.]

P7. 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 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?

2. (12 points) Do Problem P8, 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.]

P8. Referring to Problem P7, suppose the HTML file references eight very small objects on the same server. Neglecting transmission times, how much time elapses with

a. Non-persistent HTTP with no parallel TCP connections?

b. Non-persistent HTTP with the browser configured for 5 parallel connections?

c. Persistent HTTP?

5. Wireshark Hands-on Practice: DNS (26 points total; 20-40 minutes)

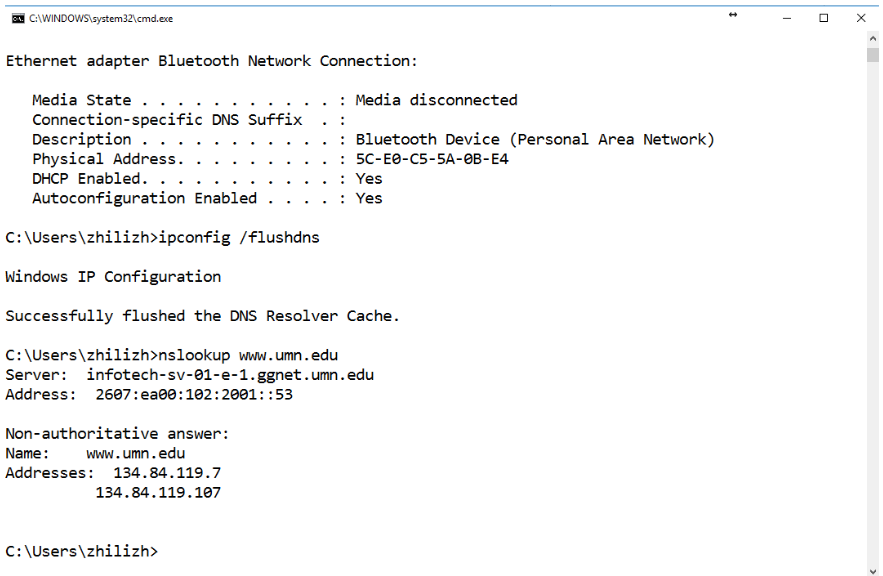
(Approximate time excludes Wireshark set-up time, learning the basics of how to use Wireshark, etc.)

Wireshark is a free network protocol analyzer that runs on Windows, Linux/Unix, and Mac computers (see pp. 77 – 78 of the textbook for a brief description of Wireshark). It is 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.



```
C:\WINDOWS\system32\cmd.exe

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Bluetooth Device (Personal Area Network)
   Physical Address. . . . . . . . . : 5C-E0-C5-5A-0B-E4
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes

C:\Users\zhilizh>ipconfig /flushdns

Windows IP Configuration

Successfully flushed the DNS Resolver Cache.

C:\Users\zhilizh>nslookup www.umn.edu
Server:  infotech-sv-01-e-1.ggnet.umn.edu
Address:  2607:ea00:102:2001::53

Non-authoritative answer:
Name:    www.umn.edu
Addresses:  134.84.119.7
          134.84.119.107


C:\Users\zhilizh>
```

Figure 1: Figure for Question 5.

In this lab, we also need to use *nslookup* tool, which allows the host running the tool to query any specified DNS server for a DNS record. To accomplish this task, *nslookup* sends a DNS query to the specified DNS server (a root DNS server, a top-level-domain DNS server, an authoritative DNS server, or an intermediate DNS server), receives a DNS reply from that same DNS server, and displays the result. *nslookup* is available in most Linux/Unix and Microsoft platforms today. To run nslookup in Linux/Unix, you just type the nslookup command on the command line. To run it in Windows, open the Command Prompt and run nslookup on the command line.

Let's start our exploration of DNS by using *nslookup* to send queries. Please do the following:

1. Run "ipconfig /flushdns" (just the letters, not the quotation marks) in the Command Prompt. This command is to flush the DNS cache clears all entries and reloads the entries from the hosts file. (see Fig. 1)

2. Start up the Wireshark packet sniffer. Enter "dns" (just the letters, not the quotation marks) in the display-filter-specification window, so that captured DNS messages will be displayed later in the packet-listing window.

3. Start the Wireshark packet capture.

4. Open the Command Prompt and do an nslookup on www.umn.edu.

5. Stop Wireshark packet capture.

You should get a trace that looks something like the following:

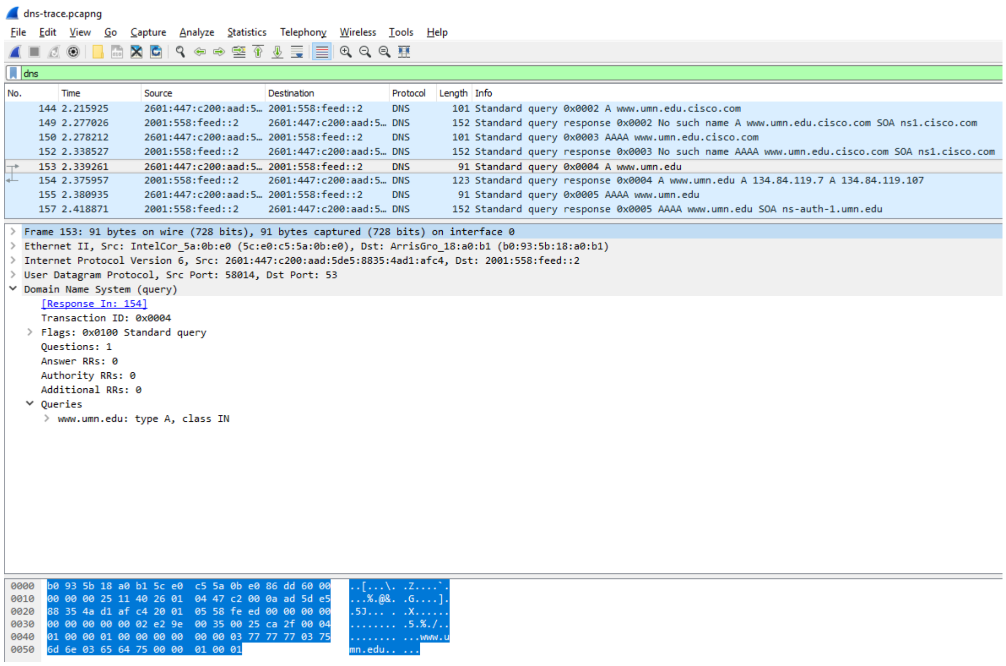


Figure 2: Figure for Question 5.

We see from the above screenshot that nslookup actually sent four DNS queries and received four DNS responses. For the purpose of this assignment, in answering the following questions, ignore the first two sets of queries/responses, as they are specific to nslookup and are not normally generated by standard Internet applications. You should instead focus on the last two sets of queries and responses messages.

Please answer the following questions. (If possible, please include a print-out or embed the screenshot you have captured into your solution document, and use it as the basis for your answer to the following questions.)

1. What is the destination port for the DNS query message? What is the source port of DNS response message?

2. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?

3. What does DNS server send the responses to your host? Do the responses come from the your local DNS server?

4. Examine the third DNS query message (as shown in Fig. 2). What "Type" of DNS query is it? Does the query message contain any "answers"?
5. Examine the third DNS response message (as shown in Fig. 2). How many "answers" are provided? What does each of these answers contain?
6. Examine the fourth DNS response message (as shown in Fig. 2). What is the authoritative name server of the university of Minnesota.
7. Provide a 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.)

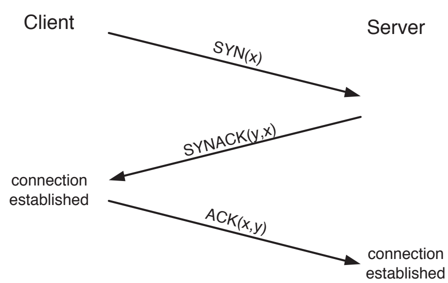


Figure 3: TCP 3-way handshake.

a. (8 points) Consider the following scenario (see Figure 4) where the `ACK(x,y)` message sent by the client is lost during the transmission. What will happen at either the client side or the server side?

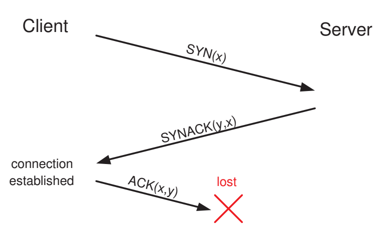


Figure 4: Figure for Question 6a.

b. (10 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 `SYN(x)` now pops up at the **server** side (see Figure **??**). First, can this scenario happen at all? Second, in response to this `SYN(x)` message, what will the *server* do? Briefly explain your answers to both questions.

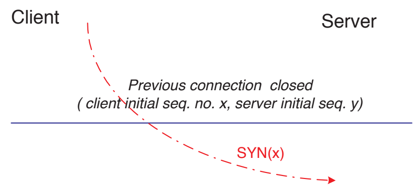


Figure 5: Figure for Question 6b.

c. (7 points) Suppose that after what happened in **b.**, another old, duplicate message `ACK(x,y)` from the previous connection pops up at the **server** side (see Figure 6). In response to this `ACK(x,y)` message, what will the server do? Briefly explain your answer.

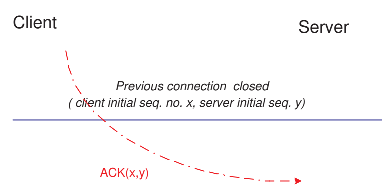


Figure 6: Figure for Question 6c.