HW1 Review

P6. This elementary problem begins to explore propagation delay and transmission delay, two central concepts in data networking. Consider two hosts, A and B, connected by a single link of rate *R* bps. Suppose that the two hosts are separated by *m* meters, and suppose the propagation speed along the link is *s* meters/sec. Host A is to send a packet of size *L* bits to Host B.

a. Express the propagation delay, d_{prop}, in terms of *m* and *s*.

$$d_{prop} = m / s$$

b. Determine the transmission time of the packet, d_{trans}, in terms of *L* and *R*.

$$d_{trans} = L / R$$

c. Ignoring processing and queuing delays, obtain an expression for the end-to-end delay.

$$d_{end-to-end} = (m / s + L / R)$$

d. Suppose Host A begins to transmit the packet at time $t=0$. At time d_{trans}, where is the last bit of the packet?

The bit is just leaving Host A.

e. Suppose d_{prop} is greater than d_{trans}. At time $t=d_{trans}$, where is the first bit of the packet?

The first bit is in the link and has not reached Host B.

f. Suppose d_{prop} is less than d_{trans}. At time $t=d_{trans}$, where is the first bit of the packet?

The first bit has reached Host B.

g. Suppose $s=2.5x10^8$, $L=120$ bits and R=56 kbps. Find the distance m so that d_{prop} equals d_{trans}.

$$m = \frac{L}{R} s = \frac{120}{56 \times 10^3}\left(2.5 \times 10^8\right) = 536 \text{ km}$$

P34. Skype offers a service that allows you to make a phone call from a PC to an ordinary phone. This means that the voice call must pass through both the Internet and through a telephone network. Discuss how this might be done.

VoIP gateway converts signals between packet switching and circuit switching.

Internet => telephone network:

1. Convert packets to analog voice
2. Send analog data to destination

Telephone network => Internet:

1. Convert analog voice data to digital data
2. Divide digitized data into packets
3. Send packets to destination

P4. Consider the following string of ASCII characters that were captured by Wireshark when the browser sent an HTTP GET message (i.e., this is the actual content of an HTTP GET message).

The characters *<cr><lf>* are carriage return and line-feed characters (that is, the italized character string *<cr>* in the text below represents the single carriage-return character that was contained at that point in the HTTP header). Answer the following questions, indicating where in the HTTP GET message below you find the answer.

```
GET /cs453/index.html HTTP/1.1<cr><lf>Host: gai
a.cs.umass.edu<cr><lf>User-Agent: Mozilla/5.0 (
Windows;U; Windows NT 5.1; en-US; rv:1.7.2) Gec
ko/20040804 Netscape/7.2 (ax) <cr><lf>Accept:ex
t/xml, application/xml, application/xhtml+xml, text
/html;q=0.9, text/plain;q=0.8, image/png,*/*;q=0.5
<cr><lf>Accept-Language: en-us, en;q=0.5<cr><lf>Accept-
Encoding: zip, deflate<cr><lf>Accept-Charset: ISO
-8859-1, utf-8;q=0.7,*;q=0.7<cr><lf>Keep-Alive: 300<cr>
<lf>Connection:keep-alive<cr><lf><cr><lf>
```

a. What is the URL of the document requested by the browser?
b. What version of HTTP is the browser running?
c. Does the browser request a non-persistent or a persistent connection?
d. What is the IP address of the host on which the browser is running?
e. What type of browser initiates this message? Why is the browser type needed in an HTTP request message?

Answers:

a. The document request was http://gaia.cs.umass.edu/cs453/index.html. The Host : field indicates the server's name and /cs453/index.html indicates the file name.

b. The browser is running HTTP version 1.1, as indicated just before the first <cr><lf> pair.

c. The browser is requesting a persistent connection, as indicated by the Connection: keep-alive.

d. This is a trick question. This information is not contained in an HTTP message anywhere. So there is no way to tell this from looking at the exchange of HTTP messages alone. One would need information from the IP datagrams (that carried the TCP segment that carried the HTTP GET request) to answer this question.

e. Mozilla/5.0. The browser type information is needed by the server to send different versions of the same object to different types of browsers.

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?

DNS resolution time = $RTT_1 + \ldots + RTT_n$
TCP connection establishment = RTT_0
HTTP request & response = RTT_0

Total delay = $RTT_1 + \ldots + RTT_n + 2\ RTT_0$

P22. Consider distributing a file of F=15 Gbits to N peers. The server has an upload rate of u_s=30 Mbps, and each peer has a download rate of d_i=2 Mbps and an upload rate of u. For N=10, 100, and 1,000 and u=300 Kbps, 700 Kbps, and 2 Mbps, prepare a chart giving the minimum distribution time for each of the combinations of N and u for both client-server distribution and P2P distribution.

Example for N=100 and u= 700 Kbps:

$$
\begin{aligned}
D_{c\text{-}s} &= max\{NF/u_{s,}, F/d_{min}\} \\
&= max\{100x15x10^9/(30x10^6), 15x10^9/(2x10^6)\} \\
&= max\ \{50000, 7500\} \\
&= 50000
\end{aligned}
$$

$$
\begin{aligned}
D_{P2P} &= max\{F/u_{s,},F/d_{min,},NF/(u_s + \Sigma u_i)\} \\
&= max\{15x10^9/(30x10^6), 15x10^9/(2x10^6), \\
&\quad 100x15x10^9/(30x10^6+100x700x10^3\} \\
&= max\{500, 7500, 1500\} \\
&= 7500
\end{aligned}
$$

Client-server:

		N		
		10	100	1000
u	300 Kbps	7500	50000	500000
	700 Kbps	7500	50000	500000
	2 Mbps	7500	50000	500000

P2P:

		N		
		10	100	1000
u	300 Kbps	7500	25000	45455
	700 Kbps	7500	15000	20548
	2 Mbps	7500	7500	7500

P26. Suppose Bob joins a BitTorrent torrent, but he does not want to upload any data to any other peers (so called free-riding).

a. Bob claims that he can receive a complete copy of the file that is shared by the swarm. Is Bob's claim possible? Why or why not?

Yes. His first claim is possible, as long as there are enough peers staying in the swarm for a long enough time. Bob can always receive data through optimistic unchoking by other peers.

b. Bob further claims that he can further make his "free-riding" more efficient by using a collection of multiple computers (with distinct IP addresses) in the computer lab in his department. How can he do that?

His second claim is also true. He can run a client on each host, let each client "free-ride," and combine the collected chunks from the different hosts into a single file. He can even write a small scheduling program to make the different hosts ask for different chunks of the file. This is actually a kind of Sybil attack in P2P networks.