EE450: Solutions to HW #2

Chapter 1: #20, 21, 23 and 31

Problem 20 (5 points)

Throughput = $min\{R_s, R_c, R/M\}$

Problem 21 (10 points)

If only use one path, the max throughput is given by:

$$\max\{\min\{R_1^1, R_2^1, \ldots, R_N^1\}, \min\{R_1^2, R_2^2, \ldots, R_N^2\}, \ldots, \min\{R_1^M, R_2^M, \ldots, R_N^M\}\}.$$

If use all paths, the max throughput is given by $\sum_{k=1}^{M} \min\{R_1^k, R_2^k, \ldots, R_N^k\}$.

Problem 23 (15 points)

Let's call the first packet A and call the second packet B.

a) (5 points) If the bottleneck link is the first link, then packet B is queued at the first link waiting for the transmission of packet A. So the packet inter-arrival time at the destination is simply L/R_s.

b) (10 points) If the second link is the bottleneck link and both packets are sent back to back, it must be true that the second packet arrives at the input queue of the second link before the second link finishes the transmission of the first packet. That is,

$L/R_s + L/R_s + d_{prop} < L/R_s + d_{prop} + L/R_c$

The left hand side of the above inequality represents the time needed by the second packet to *arrive at* the input queue of the second link (the second link has not started transmitting the second packet yet). The right hand side represents the time needed by the first packet to finish its transmission onto the second link.

If we send the second packet *T* seconds later, we will ensure that there is no queuing delay for the second packet at the second link if we have:

$L/R_s + L/R_s + d_{prop} + T >= L/R_s + d_{prop} + L/R_c$

Thus, the minimum value of T is $L/R_c - L/R_s$.

Problem 31 (4+4+4+4+4=20 points)

a) Time to send message from source host to first packet switch = $\frac{8\times10^6}{2\times10^6}\text{sec} = 4\text{sec}$ With store-and-forward switching, the total time to move message from source host to destination host = $4\text{sec}\times 3\ hops = 12\text{sec}$

b) Time to send 1st packet from source host to first packet switch = . $\frac{1\times10^4}{2\times10^6}\text{sec} = 5\ m\text{sec}$. Time at which 2nd packet is received at the first switch = time at which 1st packet is received at the second switch = $2\times 5m\text{sec} = 10\ m\text{sec}$

c) Time at which 1st packet is received at the destination host = $5\ m\text{sec}\times 3\ hops = 15\ m\text{sec}$. After this, every 5msec one packet will be received; thus time at which last (800^{th}) packet is received = $15\ m\text{sec} + 799 * 5m\text{sec} = 4.01\text{sec}$. It can be seen that delay in using message segmentation is significantly less (almost $1/3^{rd}$).

d)

i. Without message segmentation, if bit errors are not tolerated, if there is a single bit error, the whole message has to be retransmitted (rather than a single packet).

ii. Without message segmentation, huge packets (containing HD videos, for example) are sent into the network. Routers have to accommodate these huge packets. Smaller packets have to queue behind enormous packets and suffer unfair delays.

e)

i. Packets have to be put in sequence at the destination.

ii. Message segmentation results in many smaller packets. Since header size is usually the same for all packets regardless of their size, with message segmentation the total amount of header bytes is more.

To graders: for d) and e), students only need to provide one valid reason in each question.

Chapter 2: #7, 8, 10

Problem 7 (5 points)

The total amount of time to get the IP address is
$RTT_1 + RTT_2 + \cdots + RTT_n$.

Once the IP address is known, RTT_O elapses to set up the TCP connection and another RTT_O elapses to request and receive the small object. The total response time is
$2RTT_o + RTT_1 + RTT_2 + \cdots + RTT_n$

Problem 8 (12 points)

a) (3 points)
$RTT_1 + \cdots + RTT_n + 2RTT_o + 8 \cdot 2RTT_o$
$= 18RTT_o + RTT_1 + \cdots + RTT_n$.

b) (3 points)
$RTT_1 + \cdots + RTT_n + 2RTT_o + 2 \cdot 2RTT_o$
$= 6RTT_o + RTT_1 + \cdots + RTT_n$

c) (6 points)
Without pipelining:
$$RTT_1 + \cdots + RTT_n + 2RTT_0 + 8 \cdot RTT_0 = 10RTT_0 + RTT_1 + \cdots + RTT_n$$
With pipelining:
$RTT_1 + \cdots + RTT_n + 2RTT_o + RTT_o$
$= 3RTT_o + RTT_1 + \cdots + RTT_n$.

Problem 10 (19 points)

Note that each downloaded object can be completely put into one data packet. Let Tp denote the one-way propagation delay between the client and the server.

First consider parallel downloads using non-persistent connections. Parallel downloads would allow 10 connections to share the 150 bits/sec bandwidth, giving each just 15 bits/sec. Thus, the total time needed to receive all objects is given by:

(200/150+*T*p + 200/150 +*T*p + 200/150+*T*p + 100,000/150+ *T*p)
+ (200/(150/10)+*T*p + 200/(150/10) +*T*p + 200/(150/10)+*T*p + 100,000/(150/10)+ *T*p)
= 7377 + 8**T*p (seconds) (5 points)

Parallel downloads do not make sense in this case (in which transmission time is not negligible), since the parallel connections need to share the bandwidth. (2 points)

Now consider persistent HTTP without pipelining. The total time needed is given by:

(200/150+*T*p + 200/150 +*T*p + 200/150+*T*p + 100,000/150+ *T*p)
+ 10*(200/150+*T*p + 100,000/150+ *T*p)
=7351 + 24**T*p (seconds) (5 points)

Assuming the speed of light is $300*10^6$ m/sec, then Tp=10/($300*10^6$)=0.03 microsec. Tp is therefore negligible compared with transmission delay.

Now consider persistent HTTP with pipelining. The total time needed is given by:
(200/150+200/150+200/150+100000/150)+(200/150+10*100000/150)
= 7339 (seconds) (5 points)
(After receiving the first request from the client, the server can send the first object to the client without waiting for the second request. The server can send the second object right after it finishes sending the first object since it has already received the second request at that time. So the time in the second bracket is the transmission time of the first control packet plus the transmission time of 10 data packets.)

Thus, we see that persistent HTTP is not significantly faster (less than 1 percent) than the non-persistent case with parallel download. (2 points)

To graders: Students can omit Tp from the beginning since it is negligible.

Chapter 5: #27

Problem 27 (14 points)

a) (4 points) The time required to fill $L \cdot 8$ bits is

$$\frac{L \cdot 8}{128 \times 10^3} \text{sec} = \frac{L}{16} m\text{sec}.$$

b) (4 points) For $L = 1{,}500$, the packetization delay is

$$\frac{1500}{16} m\text{sec} = 93.75 m\text{sec}.$$

For $L = 50$, the packetization delay is

$$\frac{50}{16} m\text{sec} = 3.125 m\text{sec}.$$

c) (4 points) Store-and-forward delay $= \frac{L \cdot 8 + 40}{R}$

For $L = 1{,}500$, the delay is

$$\frac{1500 \cdot 8 + 40}{622 \times 10^6} \text{sec} \approx 19.4 \mu\text{sec}$$

For $L = 50$, store-and-forward delay $< 1\mu\text{sec}$.

d) (2 points) Store-and-forward delay is small for both cases for typical link speeds. However, packetization delay for $L = 1500$ is too large for real-time voice applications.