

CSE310 - Assignment 1

Ivan Lin

Problem 1

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 8×10^6 bits long that is to be sent from the 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 the 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 to destination host?

Assuming all other forms of delay are negligible, the total delay time, T_{total} will be equal to the transmission delay, T_t .

$T_1 = T_t = \frac{S}{R} = \frac{8 \times 10^6 \text{ Mb}}{2 \times 10^6 \text{ Mbps}} = 4 \text{ seconds}$, where R is the data link rate and S is the size of the message. It takes 4 seconds for the message to move to the first packet switch.

Since the system is store-and-forward, the first switch will wait to receive the entire message before sending it to the next one. Considering that the message has to cross 3 links, $T_{total} = 3 * T_1 = 3 * 4s = 12 \text{ seconds}$. It takes the message 12 seconds to reach the destination.

- 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?

Assuming all other forms of delay are negligible, the total delay time, T_{total} will be equal to the transmission delay, T_t . We will use t to denote the time to transmit a single packet.

$$t_1 = t_t = \frac{S}{R} = \frac{s}{R} = \frac{1 \times 10^4}{2 \times 10^6} = 0.005 \text{ seconds}$$

It will take 0.01 seconds for the first packet to move from the source host to the first switch. Once the first packet has been sent, the source can begin sending the second packet. It will take the same amount of time to send, so the second packet will be fully received at the first switch at time $2 * t_1 = 2 * 0.01s = 0.02 \text{ seconds}$

- 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.

The total time it takes to transmit the file from the source host to the destination is the time it takes to transmit one packet from the source to the destination, plus the time it takes to transmit the $n - 1$ packets from over one data link.

$$T_{total} = t_{total} + (n - 1)t_1 = (3 * t_1) + (800 - 1) = 3 * 0.005 + 799 * 0.005 = 4.01 \text{ seconds}$$

It takes 4.01 seconds to transmit the entire message using message segmentation as opposed to 12 seconds. This is almost 8 seconds faster, or nearly a 67% increase in transmission speed, which is quite significant.

Problem 2

Suppose two hosts, A and B, are separated by 20,000 kilometers and are connected by a direct link of $R = 2 \text{ Mbps}$. Suppose the propagation speed over the link is $2.5 * 10^8$ meters/sec.

- a. Calculate the bandwidth-delay product, $R * d_{prop}$.

$$R * d_{prop} = R \frac{\text{distance}}{\text{speed}} = 2 \text{ Mbps} \frac{20,000,000 \text{ m}}{2.5 * 10^8 \text{ m/s}} = 160,000 \text{ bits or } 160 \text{ kilobits.}$$

- b. Consider sending a file of 800,000 bits from Host A to Host B. Suppose the file is sent continuously as one large message. What is the maximum number of bits that will be in the link at any given time?

The maximum number of bits that could be in the link at any given time would be the amount of data the source can transmit during the propagation time of the first packet.

$$R * d_{prop} = 160 \text{ kb}$$

- c. Provide an interpretation of the bandwidth-delay product.

The bandwidth-delay product is, as discovered in part (b), the maximum amount of bits that will be in the link at any given time.

- d. What is the width (in meters) of a bit in the link? Is it longer than a football field?

I'm assuming that the width would refer to the length of the link divided by the amount of bits in the link to determine the amount of "space" allocated to each bit.

$$\frac{\text{distance}}{R * d_{prop}} = \frac{20,000,000 \text{ m}}{160,000 \text{ bits}} = 125 \frac{\text{m}}{\text{b}}$$

Since a football field is roughly 48.5m by 109.1m according to Google, a bit would be wider than a football field.

- e. Derive a general expression for the width of a bit in terms of the propagation speed s , the transmission rate R , and the length of the link m .

$$\frac{\text{distance}}{R * d_{prop}} = \frac{m}{R * d_{prop}} = \frac{m}{R \frac{m}{s}} = \frac{ms}{Rm} = \frac{s}{R}$$

Problem 3

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 $u=300\text{Kbps}$, 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.

Client-Server Distribution			
Peers(N) \ up speed(u)	10	100	1000
300Kbps	7500s	50000s	500000s
700Kbps	7500s	50000s	500000s
2Mbps	7500s	50000s	500000s

$$D_{C-S} \geq \max\left\{\frac{NF}{u_s}, \frac{F}{d_{min}}\right\}$$

$$N = 10 : \max\left\{\frac{10*15Gb}{30Mbps} = 5000s, \frac{15Gb}{2Mbps} = 7500s\right\} = 7500s$$

$$N = 100 : \max\left\{\frac{100*15Gb}{30Mbps} = 50000s, \frac{15Gb}{2Mbps} = 7500s\right\} = 50000s$$

$$N = 1000 : \max\left\{\frac{1000*15Gb}{30Mbps} = 500000s, \frac{15Gb}{2Mbps} = 7500s\right\} = 500000s$$

In the client-server model, the client upload speed, u has no effect on the overall distribution time. Only the server upload speed u_s does.

P2P Distribution			
Peers(N) \ up speed(u)	10	100	1000
300Kbps	7500s	25000s	45455s
700Kbps	7500s	18750s	28302s
2Mbps	7500s	7500s	7500s

$$D_{P2P} \geq \max\left\{\frac{F}{u_s}, \frac{F}{d_{min}}, \frac{NF}{u_s + \sum u_i}\right\}$$

$$u = 300Kbps, N = 10 : \max\left\{\frac{15Gb}{30Mbps} = 500s, \frac{15Gb}{2Mbps} = 7500s, \frac{10*15Gb}{30Mbps+10*300Kbps} = 4545s\right\} = 7500s$$

$$u = 300Kbps, N = 100 : \max\left\{\frac{15Gb}{30Mbps} = 500s, \frac{15Gb}{2Mbps} = 7500s, \frac{100*15Gb}{30Mbps+100*300Kbps} = 25000s\right\} = 25000s$$

$$u = 300Kbps, N = 1000 : \max\left\{\frac{15Gb}{30Mbps} = 500s, \frac{15Gb}{2Mbps} = 7500s, \frac{1000*15Gb}{30Mbps+1000*300Kbps} = 45455s\right\} = 45455s$$

$$u = 500Kbps, N = 10 : \max\left\{\frac{15Gb}{30Mbps} = 500s, \frac{15Gb}{2Mbps} = 7500s, \frac{10*15Gb}{30Mbps+10*500Kbps} = 4286s\right\} = 7500s$$

$$u = 500Kbps, N = 100 : \max\left\{\frac{15Gb}{30Mbps} = 500s, \frac{15Gb}{2Mbps} = 7500s, \frac{100*15Gb}{30Mbps+100*500Kbps} = 18750s\right\} = 18750s$$

$$u = 500Kbps, N = 1000 : \max\left\{\frac{15Gb}{30Mbps} = 500s, \frac{15Gb}{2Mbps} = 7500s, \frac{1000*15Gb}{30Mbps+1000*500Kbps} = 28302s\right\} = 28302s$$

$$u = 2Mbps, N = 10 : \max\left\{\frac{15Gb}{30Mbps} = 500s, \frac{15Gb}{2Mbps} = 7500s, \frac{10*15Gb}{30Mbps+10*2Mbps} = 3000s\right\} = 7500s$$

$$u = 2Mbps, N = 100 : \max\left\{\frac{15Gb}{30Mbps} = 500s, \frac{15Gb}{2Mbps} = 7500s, \frac{100*15Gb}{30Mbps+100*2Mbps} = 6522s\right\} = 7500s$$

$$u = 2Mbps, N = 1000 : \max\left\{\frac{15Gb}{30Mbps} = 500s, \frac{15Gb}{2Mbps} = 7500s, \frac{1000*15Gb}{30Mbps+1000*2Mbps} = 7389s\right\} = 7500s$$

Problem 4

In this assignment, you will develop a simple Web server in Python that is capable of processing only one request. Specifically, your Web server will (i) create a connection socket when contacted

by a client (browser); (ii) receive the HTTP request from this connect; (iii) parse the request to determine the specific file being requested; (iv) get the requested file from the server's file system; (v) create an HTTP response message consisting of the requested file preceded by header lines; and (vi) send the response over the TCP connection to the requesting browser. If a browser requests a file that is not presetn in your server, your server should return a "404 Not Found" error message.

* NOTE: Python 2.7.1 was used