Computer Networks - Homework 1

JJ McCauley
2/17/25

Chapter 1's Problems: 6,10,11,12,20,31

6 Propagation delay and transmission delay

6.1 Part a

The propagation delay would be as follows:

$$d_{\text{prop}} = \frac{m}{s}$$

where s = Propagation Speed, and m = Distance Between Hosts. This represents the time it takes for the first but of the packet to travel from Host A to Host B.

6.2 Part b

The transmission delay, which is the time required to push all bits of a packet onto the link, can be defined as:

$$d_{\text{trans}} = \frac{L}{R}$$

where L = packet size (in bits) and R = Transmission Rate (in bits per second, bps).

6.3 Part c

The end-to-end delay is the total time it takes for a packet to travel from the **sending host (A)** to the **receiving host (B)**. This can be modeled as:

$$d_{\text{total}} = d_{\text{trans}} + d_{\text{prop}}$$

In the context of L, R, m, and s, this can be denoted as:

$$d_{\text{total}} = \frac{L}{R} + \frac{m}{s}$$

6.4 Part d

At time $t = d_{\text{trans}}$, the last bit of the packet is just leaving Host A and entering the transmission link. Since d_{trans} is the time it takes to push all bits of the packet to the transmission link, when $t = d_{\text{trans}}$, the last bit just started going across the link. The first bit is already traveling towards **Host B**, but the last bit has not yet reached the destination.

6.5 Part e

If $d_{\text{prop}} > d_{\text{trans}}$, then at time $t = d_{\text{trans}}$, the first bit has entered the link, but it has not yet reached **Host B**. This leaves the first bit somewhere in the middle of the link between **Host A** and **Host B**.

6.6 Part f

If $d_{\text{prop}} < d_{\text{trans}}$, then at time $t = d_{\text{trans}}$, the first bit has already been received by **Host B**. Specifically, the first but would reach **Host B** at $t = d_{\text{prop}}$.

6.7 Part g

If we know that $s = 2.5 * 10^8$, $L = 1500$ bytes, and $R = 10$ Mbps, then we can use the values and the equation $d_{\text{prop}} = d_{\text{trans}}$ to find the distance. This can be broken down into the formulas:

$$\frac{m}{s} = \frac{L}{R}$$

with values of:

$$\frac{m}{2.5 * 10^8} = \frac{1500}{10 * 10^6}$$

After solving this equation for m, it is revealed that $m = 300,000$ meters.

10 End-to-End Delay

In order to compute the total End-To-End delay, we can use the equation

$$d_{\text{total}} = d_{\text{trans total}} + d_{\text{prop total}} + d_{\text{proc total}}$$

In order to determine this, we must first look at the known values. We know that $L = 1500 * 8 = 12000$ bits, $s = 2.5 * 10^8$ m/s, $R = 2.5 * 10^6$ bps, and $d_{\text{proc}} = 3 * 10^{-3}$ s. Additionally, there are variable Link distances, with $d_1 = 5,000,000$ m, $d_2 = 4,000,000$ m, and $d_3 = 1,000,000$ m.
Firstly, we can compute the transmission delays with $d_1 = \frac{L}{R} * 3$ (since all three links have the same transmission rate), which turns out to be:

$$d_{\text{trans total}} = \frac{12,000}{2,500,00} * 3 = .0048 \text{ sec} * 3 = 14.4\text{ms}$$

Then, we can calculate the propagation delays ($d_{\text{prop total}}$) by plugging in the various distances in the following equation, and adding them:

$$d_{\text{prop i}} = \frac{distance}{2.5 * 10^8}$$

After plugging in for each value, we get:

$$.02 \text{ sec} + .016 \text{ sec} + .004 \text{ sec} = 40 \text{ ms}$$

Now, knowing that there are two packet switches, we can calculate $d_{\text{proc total}}$ through the following equation:

$$d_{\text{proc total}} = 2 * 3\text{ms} = 6\text{ms}$$

Now that we have all of the information, we can plug each total into the initial equation to get the **total End-to-End Delay**.

$$d_{\text{total}} = 14.4 \text{ ms} + 40 \text{ ms} + 6 \text{ ms} = 60.4 \text{ ms}$$

Therefore, the **total End-to-End Delay is 60.4 ms**.

11 End-to-End Delay Expanded

Under these new assumptions, we can use the equation

$$d_{\text{total}} = d_{\text{trans}} + d_{\text{prop total}}$$

to find the total End-to-End delay. Since this method would only involve the use of one transmission rate, we can use the value of **4.8 ms** for the transmission rate, and we can use the previous total propagation rate of **40 ms**. Under these assumptions, we get:

$$d_{\text{total}} = 4.8\text{ms} + 40\text{ms} = 44.8\text{ms}$$

Therefore, **the new End-to-End Delay would be 44.8 ms**.

12 Queuing Delay

To calculate queuing delay, we must first look at the variables given. We know that $L = 1500$ bytes $= 12,000$ bits, $R = 2.5$ Mbps $= 2.5 * 10^6$ bps, and one packet is halfway transmitted, $\frac{L}{2}$ bits sent. Additionally, four full packets are waiting in the queue.
To determine the **transmission rate**, we can use the following equations:

$$d_{\text{trans}} = \frac{L}{R} = \frac{12,000}{2.5 * 10^6} = .0048 \text{ sec} = 4.8 \text{ ms}$$

To determine the total queuing delay, we can use the equation $d_{\text{queue}} = d_{\text{remaining}} + d_{\text{queued}}$
To get the transit time of the remaining half packet, we can use the following equation:

$$d_{\text{remaining}} = \frac{L - x}{R} = \frac{6,000}{2.5 * 10^6} = .0024 \text{ seconds} = 2.4 \text{ ms}$$

Then, to get the time for the remaining four full packets, we can simply multiply the transit time previously calculated, such that:

$$d_{\text{queued}} = 4 * 4.8 \text{ ms} = 19.2 \text{ ms}$$

Once plugging these values into the final equation, we get

$$d_{\text{queue}} = d_{\text{remaining}} + d_{\text{queued}} = 2.4 \text{ ms} + 19.2 \text{ ms} = 21.6 \text{ ms}$$

Leaving us with a **total queuing delay of 21.6 ms**. Generally, an equation can be derived by combining these calculations, resulting in:

$$d_{\text{queue}} = \frac{L - x}{R} + n * \frac{L}{R}$$

20 Throughput General Expression

This problem involves M client-server pairs, with R_{s} representing the rate for the server link, R_{c} representing the rate for the client link, and R representing the rate for the network link. Additionally, we assume that all other links have abundant capacity (no bottlenecks) and no other traffic is present, other than the M client-server pairs.
We know that throughput is limited by the **slowest link** in the path. Knowing this, we can derive that, with M servers:

The total server-side rate is MR_{s}

The total client-side rate is MR_{c}

Since the network link capacity of R must be divided by M connections, each client-server pair gets at most:

$$\frac{R}{M}$$

Given these three values, we know that the actual throughput will be bottle-necked by the slowest rate, which can be found using

$$T = \min(R_{\text{s}}, R_{\text{c}}, \frac{R}{M})$$

Lastly, since there are M client-server pairs, **the total throughput can be calculated as:**

$$T_{\text{total}} = M * \min(R_{\text{s}}, R_{\text{c}}, \frac{R}{M})$$

31 Segmentation

For this question, we will firstly consider that each message is 10^6 **bits long**, with each link being **5 Mbps**, or $5 * 10^6$ bps

31.1 Part a - Sending Message Without Segmentation

Firstly, we must determine the transmission rate, which can be calculated as:

$$d_{\text{trans}} = \frac{L}{R} = \frac{10^6}{5 * 10^6} = .2 \text{ sec} = 200 \text{ ms}$$

This time (**200 ms**) represents the time it would take for the packet to travel from the source host to the first packet switch. Now, since this uses a **store-and-forward** method, it must completely transfer the packet before forwarding to the next link. Therefore, the calculation is fairly easy:

$$d_{\text{total}} = N * d_{\text{trans}} = 3 * 200 \text{ ms} = 600\text{ms}$$

where N = the number of links.
Thus, the total **end-to-end delay without segmentation would be 600 ms**.

31.2 Part b - First Packets when Segmenting into 100 Packets

Now, we divide the packet into 100 packets such that each packet = $10,000$ bits.

$$d_{\text{trans}} = \frac{L}{R} = \frac{10,000}{5 * 10^6} = .002 \text{ sec} = 2 \text{ ms}$$

This concludes that each packet takes **2 ms** to be transmitted to a link. Therefore, the **first packet finishes transmission at 2 ms**. When the first packet moves to the second link, the second packet begins transmission, meaning that the **second packet finishes at 4 ms** (.2 ms $*$ 2).

31.3 Part c - Total Transmission Time when Segmenting

In order to calculate the total delay, we know that $N = 3$, the total number of links, $M = 100$, the total number of packets, and $d_{\text{trans per packet}} = 2$ ms. Using the following equation to receive the total transmission time:

$$d_{\text{total}} = d_{\text{trans per packet}} + (N - 1) * d_{\text{trans per packet}} + (M - 1) * d_{\text{trans per packet}}$$

We can substitute the known numbers to receive the total delay:

$$d_{\text{total}} = 2 \text{ ms} + (3 - 1) * 2 \text{ ms} + (100 - 1) * 2 \text{ ms} = 204 \text{ ms}$$

This leaves us with a **total delay when using segmentation to be 204 ms**, which is significantly *less* than the 600 ms delay obtained when not using segmentation. This is because the links does not have to wait for the entire packet to be received, instead being able to send it through piece by piece without having to wait.

31.4 Part d - Benefits of Message Segmentation

In addition to **reducing delay**, Message segmentation allows for **improved link utilization**, with each link constantly active with little idle time, and **better error handling**, with errors in transmission only requiring one small packet to be resent rather than the entire large file.

31.5 Part e - Drawbacks of Message Segmentation

When using message segmentation, it is important to consider **increased overhead**, as each packet needs its own metadata which increases the overall total data sent, **increased complexity for end host** as it must reassemble all the packets in the correct order, and **higher packet loss risk** as each packet can individually be lost.