Assignment 1

Joshua Hwang (44302650)

10 March

1 Remote island

It first looks for the first DNS it can find. This will be through C to Switch 1 to Switch 3 then to Node D. From there it gets sent back to Node C. Now knowing where to go we take Node C to S1 to S2 to V and all the way back.

For the sake of easing later calculations we will evaluate the times for each link. The formula for determining the propagation delay time is Length $\div$ Propogation rate = Time to travel.

We do this process for all links in the table.

$$L_1 = 7.5 \times 10^{-7}$$
$$L_2 = 1.2 \times 10^{-1}$$
$$L_3 = 7.5 \times 10^{-7}$$
$$L_4 = 3.0 \times 10^{-2}$$
$$L_5 = 7.5 \times 10^{-7}$$

1.1 to DNS

The delay from C to the first Node (Switch 1). Please note the packet is 100 BYTES while our delays use BITS.

$$\begin{aligned} d_{nodal} &= d_{proc} + d_{queue} + d_{trans} + d_{prop} \\ &= 0 + 0 + \frac{800}{100000000} + 7.5 \times 10^{-7} \\ &= \frac{8}{1000000} + 7.5 \times 10^{-7} \\ &= 8 \times 10^{-6} + 7.5 \times 10^{-7} \\ &= 8.75 \times 10^{-6} \end{aligned}$$

The delay from Switch 1 to Switch 3

$$\begin{aligned} d_{nodal} &= d_{proc} + d_{queue} + d_{trans} + d_{prop} \\ &= 5 \times 10^{-4} + 0 + \frac{800}{100000000} + 3 \times 10^{-2} \\ &= 0.05 \times 10^{-2} + \frac{8}{1000000} + 3 \times 10^{-2} \\ &= 0.05 \times 10^{-2} + 0.0008 \times 10^{-2} + 3 \times 10^{-2} \\ &\approx 3 \times 10^{-2} \end{aligned}$$

The delay from Switch 3 to D

$$\begin{aligned} d_{nodal} &= d_{proc} + d_{queue} + d_{trans} + d_{prop} \\ &= 5 \times 10^{-4} + 0 + \frac{800}{100000000} + 7.5 \times 10^{-7} \\ &= 0.05 \times 10^{-2} + \frac{8}{1000000} + 7.5 \times 10^{-7} \\ &= 0.05 \times 10^{-2} + 0.0008 \times 10^{-2} + 7.5 \times 10^{-7} \\ &\approx 5 \times 10^{-4} \end{aligned}$$

Thus the total time taken comes up to be roughly 3.051×10^{-2} seconds. This aligns with the recieve time of 3.051×10^{-2} seconds as well.

1.2 from DNS

Since the links are bidirectional, it stands to reason that we will get the same results coming from the DNS to C. Thus the total time taken comes up to be roughly 3.051×10^{-2} seconds. Thus, the time at which C recieves the IP is 6.102×10^{-2} seconds.

1.3 to V

The delay from C to the first Node (Switch 1)

$$\begin{aligned} d_{nodal} &= d_{proc} + d_{queue} + d_{trans} + d_{prop} \\ &= 0 + 0 + \frac{800}{100000000} + 7.5 \times 10^{-7} \\ &= \frac{8}{1000000} + 7.5 \times 10^{-7} \\ &= 8 \times 10^{-6} + 7.5 \times 10^{-7} \\ &= 8.75 \times 10^{-6} \end{aligned}$$

The delay from Switch 1 to Switch 2

$$\begin{aligned} d_{nodal} &= d_{proc} + d_{queue} + d_{trans} + d_{prop} \\ &= 5 \times 10^{-4} + 0 + \frac{800}{5000} + 1.2 \times 10^{-1} \\ &= 5 \times 10^{-4} + 0 + 0.16 + 1.2 \times 10^{-1} \\ &= 0.0005 + 0.16 + 0.12 \\ &\approx 0.28 \end{aligned}$$

The delay from Switch 2 to V

$$\begin{aligned} d_{nodal} &= d_{proc} + d_{queue} + d_{trans} + d_{prop} \\ &= 5 \times 10^{-4} + 0 + \frac{800}{100000000} + 7.5 \times 10^{-7} \\ &= 0.05 \times 10^{-2} + \frac{8}{1000000} + 7.5 \times 10^{-7} \\ &= 0.05 \times 10^{-2} + 0.0008 \times 10^{-2} + 7.5 \times 10^{-7} \\ &\approx 5 \times 10^{-4} \end{aligned}$$

Thus the total time taken comes up to be roughly 0.281 seconds. Thus, the time at which V recieves the `TCP SYN` is 0.342 seconds.

1.4 from V

Since the links are bidirectional, it stands to reason that we will get the same results coming from the V to C. Thus the total time taken comes up to be roughly 0.281 seconds. Thus, the time at which C recieves the `TCP SYN ACK` is 0.623 seconds.

1.5 to V (`HTTP`)

The difference in this case is instead of sending a `TCP SYN` we're sending a `HTTP GET` which is 10 times larger.

The delay from C to the first Node (Switch 1)

$$\begin{aligned} d_{nodal} &= d_{proc} + d_{queue} + d_{trans} + d_{prop} \\ &= 0 + 0 + \frac{8000}{100000000} + 7.5 \times 10^{-7} \\ &= \frac{8}{100000} + 7.5 \times 10^{-7} \\ &= 8 \times 10^{-5} + 7.5 \times 10^{-7} \\ &= 8.75 \times 10^{-5} \end{aligned}$$

The delay from Switch 1 to Switch 2

$$\begin{aligned} d_{nodal} &= d_{proc} + d_{queue} + d_{trans} + d_{prop} \\ &= 5 \times 10^{-4} + 0 + \frac{8000}{5000} + 1.2 \times 10^{-1} \\ &= 5 \times 10^{-4} + 0 + 1.6 + 1.2 \times 10^{-1} \\ &= 0.0005 + 1.6 + 0.12 \\ &\approx 1.725 \end{aligned}$$

The delay from Switch 2 to V

$$\begin{aligned} d_{nodal} &= d_{proc} + d_{queue} + d_{trans} + d_{prop} \\ &= 5 \times 10^{-4} + 0 + \frac{800}{100000000} + 7.5 \times 10^{-7} \\ &= 0.05 \times 10^{-2} + \frac{8}{1000000} + 7.5 \times 10^{-7} \\ &= 0.05 \times 10^{-2} + 0.0008 \times 10^{-2} + 7.5 \times 10^{-7} \\ &\approx 5.8075 \times 10^{-4} \end{aligned}$$

Thus the total time taken comes up to be roughly 1.7256 seconds. Thus, the time at which V recieves the `TCP SYN` is ≈ 2.349 seconds.

1.6 from V (`HTTP`)

The problem becomes much more complicated as each of the packets get placed in a queue, waiting for the intial packets to be processed.

From V to S2 the server throws all 5 html and http packets at S2. S2 will take

these and queue each packet. Thus all 5 packets arrive by,

$$\begin{aligned} d_{nodal} &= d_{proc} + d_{queue} + d_{trans} + d_{prop} \\ &= 0 + 0 + \frac{8000}{100000000} + 7.5 \times 10^{-7} \\ &= 8 \times 10^{-5} + 7.5 \times 10^{-7} \\ &= 8.075 \times 10^{-5} \end{aligned}$$

Now we have S2 to S1. Each packet gets processed then sent out when they are at the front of the queue. Each packet will take 1ms to get processed so our final packet will take 5ms. Note, there is now a 1ms distance between each packet along L2 now. Thus the time for the final packet along with line is,

$$\begin{aligned} d_{nodal} &= d_{proc} + d_{queue} + d_{trans} + d_{prop} \\ &= 0.001 + 0.004 + \frac{8000}{5000} + 1.2 \times 10^{-1} \\ &= 0.005 + 1.6 + 0.12 \\ &= 1.725 \end{aligned}$$

Each packet that enters S1 has a processing time of $500\mu s$ but the distance between each packet is 1ms at this point. Thus, each packet will get sent before another one arrives, no queueing occurs. This is the final part of the journey for these packets.

$$\begin{aligned} d_{nodal} &= d_{proc} + d_{queue} + d_{trans} + d_{prop} \\ &= 5 \times 10{-}4 + 0 + \frac{8000}{100000000} + 7.5 \times 10^{-7} \\ &= 0.005 + 8 \times 10^{-5} + 7.5 \times 10^{-7} \\ &= 0.0058075 \end{aligned}$$

Thus, the time the final packet arrives is 1.731. In total the time is now 4.080

1.7 Closing

The client must send acknowledgement of packets received before any connections can be closed. The final acknowledgement will get sent when the final html packet is received. This is equivalent to the SYN packet sent earilier. Thus, the time taken is 0.281. The total time is now 4.361.

Likewise the `TCP FIN` message will also take as long, 0.281. The `TCP FIN/ACK` and the final `TCP ACK` is sent back to C all also take 0.281 seconds. Thus the total time taken by this is 0.843 and the total time is 5.204.

2 Wireshark exercise

2.1 Used application layer for sending

`SMTP` beginning with `no. 5`. From the IETF RFC `SMTP` is used "for the transmission of mail." (RFC5321 pg. 9)

2.2 Mail client `IP` and port

`192.168.0.17:25` This is because this private `IP` is present as source or destination for all internet messages.

2.3 Mail server `IP` and port

`211.29.132.250:50457` It is the only `IP` which `SMTP` gets sent to.

2.4 What mail client is being used

`X-Mailer: Microsoft Outlook 16.0` found in `no. 15` under a `SMTP` header.

2.5 What is the first line of the message

`The Man from Snowy River` found in `no. 15` and was the first non-header data that wasn't `<CRLF>`.

2.6 Max message size

`50` is a `SYN/ACK` message with the option of maximum segment size of 1460 bytes.

2.7 Used application layer for reading

`POP` as stated in the IETF RFC "`POP3` is not intended to provide extensive manipulation operations of mail on the server; normally, mail is downloaded and then deleted." (RFC1939 pg. 2)

2.8 Username and password

`no. 56` and `no. 58` show the user is `nbnbnbnb` and the password is `Coms7201` respectively.

2.9 How many packets used in receiving

4 packets were need to send the body of the message; `no. 68`, `no. 70`, `no. 71`, `no. 73`.

2.10 Size of first email in maildrop

Using `no. 65` where the mail server responds to a previous `LIST` command. Each line of the returned statement gives the index of each email and an exact size for each email (RFC1939 pg. 7). Thus the first message has a size of 2210 octets (bytes).