Homework 3 for Evolving Internet

Gabriele Genovese

07 November 2024

1 Exercise 1

1.1 Point a

1. The initial setup is $L(A) = 0, L(B) = \infty, L(C) = \infty, L(D) = \infty, L(E) = \infty$.
2. Then we add the weight of the nodes reachable from the source A

$$L(B) = 0 + w_{AB}(0) = 0 + 1 = 1$$

$$L(C) = 0 + w_{AC}(0) = 0 + 3 = 3$$

$$L(E) = 0 + w_{AE}(0) = 0 + 9 = 9$$

We update all the node's label because they are all less than ∞. The minimum weight is B.

3. We add the weight of the nodes reachable from B

$$L(C) = 1 + w_{BC}(1) = 1 + 3 = 4$$

$$L(D) = 1 + w_{BD}(1) = 1 + 0 = 1$$

We update only $L(D) = 1$. We take the minimum label which is D.

4. We add the weight of the nodes reachable from D

$$L(C) = 1 + w_{DC}(1) = 1 + 6 = 7$$

$$L(E) = 1 + w_{DE}(1) = 1 + 1 = 2$$

We update only $L(E) = 2$. The minimum weight is E, so we take it.

5. We add the weight of the nodes reachable from E.

$$L(C) = 2 + w_{EC}(1) = 2 + 1 = 3$$

We don't update it because it's already 3. We take the node C, using the edge from A.

The final labels of the routes from A at time 0 will be

$$L(A) = 0, L(B) = 1, L(C) = 3, L(D) = 1, L(E) = 2$$

using links AB, AC, BD, DE.

1.2 Point b

If A want to comunicato to D at time 5 using the previous calculation, it will have to pass through B. But it will have to wait until time 7 to communicate and there might be a better route using C or E. Therefore, the old routes will not hold. We need to do the computation again from time = 5.

- First step:

$$L(B) = 5 + w_{AB}(5) = 5 + 2 = 7$$

$$L(C) = 5 + w_{AC}(5) = 5 + 1 = 6$$

$$L(E) = 5 + w_{AE}(5) = 5 + 4 = 9$$

We take C.

- Second step:

$$L(B) = 6 + w_{CB}(6) = 6 + 0 = 6$$

$$L(D) = 6 + w_{CD}(6) = 6 + 1 = 7$$

$$L(E) = 6 + w_{CE}(6) = 6 + 3 = 9$$

From $L(B) = 6, L(D) = 7, L(E) = 9$ we take B.

- Third step:

$$L(D) = 6 + w_{BD}(6) = 6 + 4 = 10$$

We don't update D's label. From $L(D) = 7, L(E) = 9$ we take D.

Therefore, the final label to reach D from A at time 5 will be 7, passing through C.

2 Exercise 2

2.1 Point 1

We know that Throughput $= \frac{\text{Window Size}}{\text{RTT}}$. So, to find the Window Size = Throughput $\times$ RTT, that is

$$\text{Window Size} = 10 \times 0.2 = 2 \text{ Mb}$$

in bytes: $2 \cdot 1024 \cdot 1024 \cdot \frac{1}{8} = 262144$ bytes.

To write the number 262144 we need 19 bit but the tcp header field of the window field is 16 bits. So it's not enough to announce to the server the required window.

2.2 Point 2

If we use 100% of the downlink bandwidth, we know that in a RTT we transmit 262144 bytes of data. In terms of packets they are:

$$\text{Numeber of Packets} = \frac{262144}{1500} \approx 174 \text{ packets}$$

If we multiply the numeber of packet to 5 (number of chunk of data transmited in one second) we obtain that we can transmit $174 \cdot 5 = 870$ packets in one second. Therefore, the client needs to upload in 870 acknowledgments packets. That's equal to $870 \cdot 60 = 52200$ bytes. 1 Mbps of upload in bytes is equal to $\frac{1 \cdot 1024 \cdot 1024}{8} = 131072$ bytes/second. Lastly,

$$\frac{52200}{131072} \approx 0.3983 = 39.83\%$$

of the uplink bandwidth will be used to transmit acknowledgments packets.

2.3 Point 3

2.3.1 Point a

A packet is lost every 5 seconds. During this period of time we can do $\frac{5}{0.2} = 5 \cdot 5 = 25$ RTT. So we increase the number of packets 25 times. Therefore, W that rappresents congestion window increase corresponds to

$$W = 25 \cdot 1500 = 37500 \text{ bytes}$$

2.3.2 Point b

We recall that Throughput $= \frac{\text{Window Size}}{\text{RTT}}$. The window size is dynamic through the time so we can take the mean windows size to calculate the throughput.

$$\text{Mean window size } = \frac{2W+W}{2} = 1.5W = 1.5 \cdot 37500 = 56250 \text{ bytes/second}$$

Therefore, a way to approximate the throughput is

$$\text{Throughput} = \frac{\text{Window Size}}{\text{RTT}} = \frac{56250}{0.2} = 281250 \text{ bytes/second}$$

Let's compute the real value using the formula

$$\text{Throughput} = \frac{\text{Number of packets transmitted per cycle L}}{\text{Cycle duration S}}$$

Compute L using

$$L = W + (W + 1 \cdot 1500) + (W + 2 \cdot 1500) + ... + (W + 25 \cdot 1500) =$$

$$25 \cdot W + \left(\sum_{i=1}^{25} i \cdot 1500\right) = 25 \cdot W + \frac{25 \cdot (25+1)}{2} \cdot 1500 = 1425000 \text{ bytes}$$

$$S = 5 \text{ seconds}$$

$$\text{Throughput} = \frac{L}{S} = \frac{1425000}{5} = 285000 \text{ bytes/second}$$

These two solutions are similar.

2.4 Point 4

2.4.1 Point a

From the previous exercise we know that Throughput $= \frac{L}{S}$, where we can express the formula in funtion of T.

The number of RTT is $\frac{T}{0.2} = 5 \cdot T$.

Let's also express W in functon of T:

$$W = 5 \cdot T \cdot 1500$$

Then, L now is

$$L = 5 \cdot T \cdot W + \frac{5 \cdot T \cdot (5 \cdot T + 1)}{2} \cdot 1500$$

$$S = T$$

$$X = \left(5 \cdot T \cdot 5 \cdot T \cdot 1500 + \frac{5 \cdot T \cdot (5 \cdot T + 1)}{2} \cdot 1500\right) \cdot \frac{1}{T} =$$

$$= 1500 \cdot 5 \cdot \left(5 \cdot T + \frac{5 \cdot T}{2} + \frac{1}{2}\right) = 1500 \cdot 5 \cdot (7.5 \cdot T + 0.5)$$

X can be seen as a linear function in the form of $y = ax + b$ with $x < 5$. We can use a python program to visualize the function.

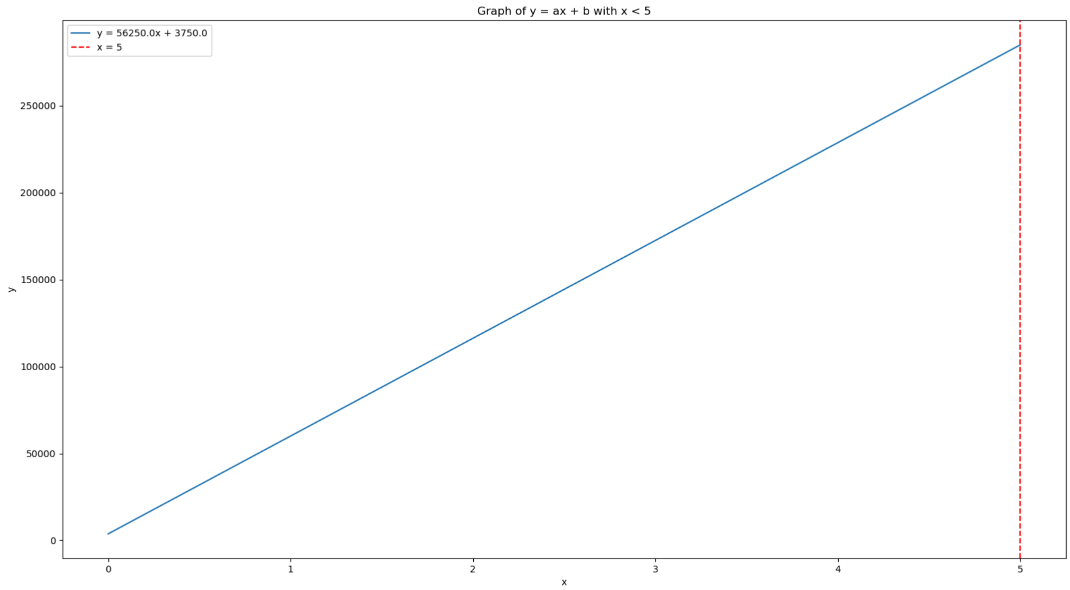

Abb. 1: Graph of the window size in function of T

The value is within the range of $(0, 285000)$ (in bytes) and will have the following values

$$f(1) = 60000 \text{ bytes}$$

$$f(2) = 116250 \text{ bytes}$$

$$f(3) = 172500 \text{ bytes}$$

$$f(4) = 228750 \text{ bytes}$$

2.4.2 Point b

If $T > 5$, the window size increments linearly. We need to find a T such that the value of the equation in a) is equal to 10Mbps that are $\frac{10 \cdot 1024 \cdot 1024}{8} = 1310720$ bytes.

$$1500 \cdot 5 \cdot (7.5 \cdot T_0 + 0.5) = 1310720$$

$$T_0 = 23.235 \text{ seconds}$$

In approximately 23 seconds the windows size will reach the maximum downlink capacity.

2.4.3 Point c

If the window size continues to increase beyond the maximum downlink capacity (that is $T > T_0$), the link will become congested, leading to potential packet loss and increased latency in data transmission.