CSC 458/2209: Computer Networking Systems, Winter 2025

Department of Computer Science, University of Toronto

Problem Set 1 Solutions **Date:** Friday February 14, 2025

1a) We have 7000 bytes of data + 20 bytes of header which is less than the MTU of the link A → R1 (which is 8000 bytes) so A will not fragment the packet. R1 → R2 has an MTU of 1500 bytes, which means it will fragment the original IP datagram. The next link has an MTU of 9000 so no fragmentation will happen at R2 → B.

1b) With 7000 bytes of data, A will generate one IP datagram of 7020 bytes total. At R1, this packet will be fragmented into 4 packets of size 1480 (data) + 20 (IP header) + one packet of size 1080 bytes (data) + 20 bytes of IP header for a total of 4x1500 + 1100 =7,100 bytes.

1c) Typically, reassembly of fragments happens at the end-host and not at the intermediate routers or other intermediate devices for the following reasons:

- Reassembling fragments means we need to keep them in a buffer before we can recreate the original datagram. In the middle of the network we are dealing with a large number of flows and extremely high volumes of traffic which means we will need extremely large buffers.
- In general, packets of the same flow do not necessarily take the same route, which means if an intermediate node wants to reassemble the fragments, it might not have access to all fragments. In this example, we have a single path between A and B so we do not have route variations.
- Reassembly of packets in the intermediate node can add significant delay to the end-to-end latency of the traffic.

2a)

Based on the diagram shown in the problem, T1 is equal to 3 times the transmission of a single packet on Host A+ propagation of packet from A to R1.

T1 = 3x1000/400,000 + 0.003 = 0.0105 sec = 10.5ms

T2 = T1 + 1000/800,000 + 0.0015 = 0.01325 sec = 13.25ms

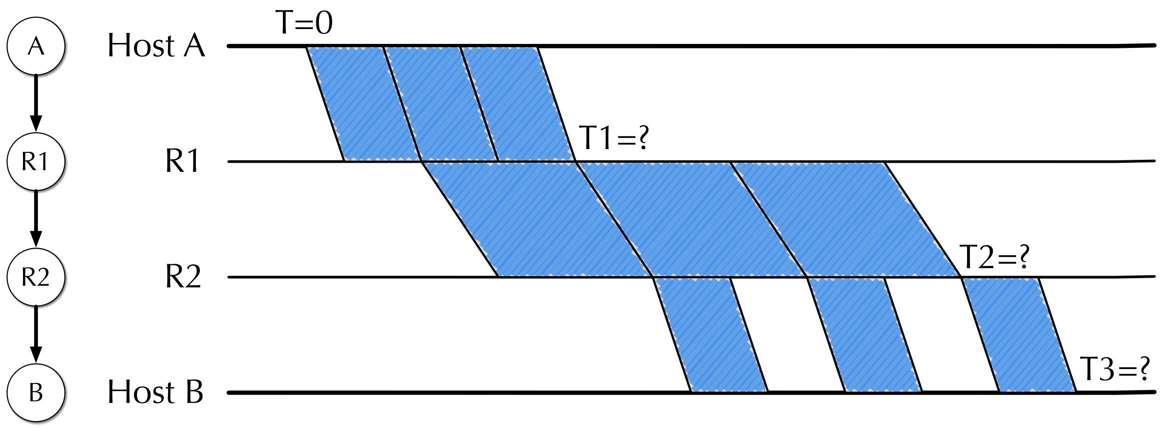
T3 = T2 + 1000/800,000 + 0.001 = 15.5 ms

2b)

T1 and T2 do not change here. For T3, we need to change the transmission time and propagation delay from R2 to B based on the new information.

T3 = T2 + 1000/400,000 + 0.0025 = 0.01825 sec = 18.25 ms

2c) In this scenario, the bottleneck is going to be the link between R1 and R2, and packet transmissions will look like the following figure.



Here, we have:

T1 = 3x1000/800,000 + 0.0015 = 5.25ms

To calculate T2, let us consider the first packet.

Time for the first packet to arrive at R1: 1000/800,000 + 0.0015 =0.00275 second = 2.75 ms
Time for the first packet to arrive at R2: 0.00275 + 1000/400,000 + 0.003 =0.00825 second = 8.25ms.

From this point on every 1000/400,000 seconds one packet arrives at R2. Therefore:

T2 = 0.00825 + 2 * 1000/400,000 = 0.01325 seconds = 13.25ms

Finally, T3 is T2 + the time it takes for one packet to go from R2 to B (please see the figure above).

T3 = T2 + 1000/800,000 + 0.001 = 15.5 ms

Therefore, the overall latency does not change compared to the previous case, but T1 changes here.

3.a) Root = S1

Each node will find its shortest path towards the root. S2, S3, and S5 are directly connected to the root and that's the path they are going to choose to reach the root. S4 can reach S1 either through S3 or S5, but since S3 has a lower ID than S5, the node S4 will use S3 to reach the root. As a result the following links will be discarded: (S5, S2), (S3, S2) and (S4, S5) as the are not on the path to the root for any of the nodes after the STP converges.

3b) When A sends a packet to B, since none of the nodes know where B is they will simply broadcast the packet. As a side-effect, all switches will learn about A. When, D sends packet to A, all switches on the path (S5, S1, and S2) already know about A so they will directly send the packet towards A. Here, S5, S1, and S2 will learn about D. When, B sends a packet to D, the switches on the path (S3, S1, and S5 will learn about B.

Forwarding Table for S1

Destination	Next Hop
A	S2
D	S5
B	S3

Forwarding Table for S2

Destination	Next Hop
A	A
D	S1

Forwarding Table for S3

Destination	Next Hop
A	S1
B	B

3c) Here, when C sends to B, the packet goes through S4, S3, and to B. Therefore, S3 and S4 learn about C. When B sends a packet to C the packet goes from B to S3 to S4 and C. S3 already knows about B so it will not change anything.

Forwarding Table for S1

Destination	Next Hop
A	S2
D	S5
B	S3

Forwarding Table for S2

Destination	Next Hop
A	A
D	S1

Forwarding Table for S3

Destination	Next Hop
A	S1
B	B
C	S4

3d)

C → B: The shortest path from C to B goes through C → S4 → S3 → B which has 3 links. In the spanning tree the same path exists so the stretch factor is 3/3 = 1.

C → A: The shortest path from C to A is C → S4 → S3 → S2 →A (with 4 links). The path over the spanning tree is C → S4 → S3 → S1 → S2 → A (5 links) so the stretch factor is 5/4 =1.25.

C → D: The shortest path from C to D goes has 3 links (C → S4 → S5 → D). The path over the spanning tree is C → S4 → S3 → S1 → S5 → D which has 5 5 links. Stretch factor = 5/3 =1.666.
3e) Changing the ID of one switch can change the spanning tree. For example, if we change the ID of switch 1 to 6 the new spanning tree will be rooted at S2. Or, changing the ID of switch 4, will make this node the root of the tree.

In general, this can lead to significant changes in the stretch-factor of the network, but it turns out in this specific network changing only one ID does not reduce the maximum stretch (which is 1.666 from previous example). For instance, if we change the root to node S4 (by changing the ID to zero) the path from C to D will have a stretch-factor of 1 (unlike previous case which has a stretch factor of 1.666) but on the other hand hand the stretch factor of A to D goes from 1 to 1.666. You can verify that changing the ID of any of the switches will still lead to a maximum stretch factor of 1.666.

4) Please see the example provided in the course handouts for a similar problem.

5)

a) Destination = 140.20.100.15 → B
b) Destination = 140.20.250.1 → A
c) Destination = 140.20.101.200 → C
d) Destination = 141.0.1.5 → D