

Last time: Packet Switching

S — r1 — X — r2 — R

Propagation Delay:

$$\left(\frac{l_1 = \text{distance of the link from S to X}}{c_1 = \text{speed of light (in this medium)}} \right)$$

+ Serialization Delay:

$$\left(\frac{p = \text{size of the packet}}{r_1 = \text{link rate (bits/second)}} \right)$$

----- How long does it take for a packet to arrive from sender S to router X -----

+ Queueing Delay (waiting at the out-going queue):

$$\left(\sum_{i \in \text{out-going queue}} \text{serializationdelay}(i) = \frac{\sum_{i \in \text{out-going queue}} \text{size}_i}{r_2 = \text{link rate of the out-going link}} \right)$$

----- After it arrives at X, how long does it take for it to leave X -----

$$\frac{l_2}{c_2} + \frac{p}{r_2}$$

----- After it leaves X, how long does it take to arrive R -----

Q: Multiplexing?

A: The underlying links are multiplexed across different flows. Each packet is an atomic unit.

Q: How does a router know a flow ends?

A: The router knows nothing about the flow. It only cares about each individual packet.

Gradescope Problem:

S ----- L1 ----- A ----- L2 ----- R

L1: $r_1 = 1 \text{ Mbits/s}$, $l_1 = 200 \text{ Mm}$, $c_1 = 200 \text{ Mm/s}$

L2: $r_2 = 1 \text{ kbits/s}$, $l_2 = 100 \text{ Mm}$, $c_2 = 200 \text{ Mm/s}$

At time 0, the sender S send two packages of the same length 10 kbits. Question: When will the LAST bit of the second packet arrive at R?

- For the first packet

The whole packet arrives at A:

$$\frac{l_1}{c_1} + \frac{p}{r_1} = \frac{200 \text{ Mm}}{200 \text{ Mm/s}} + \frac{10 \text{ kbits}}{1 \text{ Mbits/s}} = 1 \text{ second} + 0.01 \text{ second} = 1010 \text{ ms}$$

The first packet is sent without any queueing delay, and therefore it is fully sent from A

$$\text{at: } 1010 \text{ ms} + \frac{p}{r_2} = 1010 \text{ ms} + \frac{10 \text{ kbits}}{1 \text{ kbits/s}} = 1010 \text{ ms} + 10 \text{ second} = 11010 \text{ ms}$$

- For the second packet

The second packet leaves S after the first packet leaves S, and then it arrives at A:

$$\frac{p}{r_1} + \frac{l_1}{c_1} + \frac{p}{r_1} = 0.01 \text{ second} + 1 \text{ second} + 0.01 \text{ second} = 1020 \text{ ms}$$

(Note that you should not need to calculate this when you do the problem, since the second packet arrives at A way before the first packet leaves A, and you should be able to guess that looking at the rate difference between L1 and L2.

The second packet leaves A after the first packet is fully sent from A, and therefore it arrives at R:

$$11010 \text{ ms} + \frac{l_2}{c_2} + \frac{p}{r_2} = 11010 \text{ ms} + 0.5 \text{ second} + 10 \text{ second} = 21510 \text{ ms}$$

Next: Congestion Control

As you see in the Gradescope problem, the performance is bottlenecked at leaving A, because L2 is much slower than L1. What happens if more and more packets are waiting for the same out-going queue? That could potentially cause a buffer overflow at A. The throughput of sending from S to R is bottlenecked by the slowest link on the path. Is there anything smart we could do to prevent more and more packets waiting at A?