PRACTICE QUESTIONS ON RESOURCE ALLOCATION

QUESTION 1: Internet Versus Station Wagon

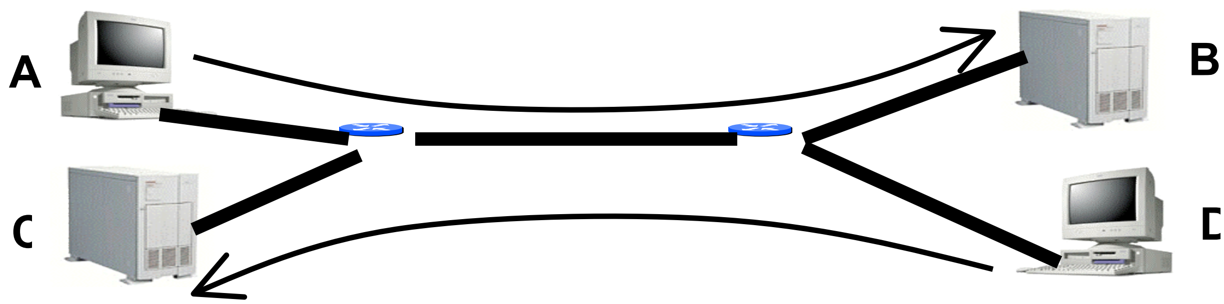
A famous maxim, sometimes attributed to Dennis Ritchie, says "Never underestimate the bandwidth of a station wagon full of tapes." Suppose you need to deliver a large file between two computers that are 200 km apart and connected by a direct link. This question analyzes when it is faster to *drive* the data between the two locations, rather than *transmit* the data?

(1a) Suppose the station wagon drives 100 km/hour and the link has a bandwidth of 800,000 bits/second. And, suppose for simplicity that the data transfer can fully consume the link bandwidth, with no additional overhead (e.g., for headers or the TCP sawtooth). At what file size (in bytes) does the station-wagon solution start delivering the data faster? Show your work.

(1b) Suppose the sender transmits the data as packets with a 20-byte IP header, and 20-byte TCP header, and a maximum segment size of 512 bytes. How much more time (as a fraction) would the data transfer take on the link? (Ignore TCP congestion control and the link-layer header.)

(1c) Suppose that, like the path traversed by the car, the link is 200 km long. The speed of electricity in a copper cable is 200,000,000 meters/second. How big does the receive window need to be to avoid becoming the main constraint on the transfer rate? (Ignore the effects of header sizes, TCP congestion control, and packet loss, and assume that the receiver immediately sends an ACK packet after receiving each data packet and that there is no congestion.)

QUESTION 2: Transmission Overhead



Suppose A has a TCP connection with B, where A sends data packets and B sends ACKs; similarly, suppose D has a TCP connection with C, where D sends data packets and C sends ACKs. Suppose the Maximum Segment Size (MSS) is 472 bytes, and all packets sent by A and D have this size; suppose also that B and C send an ACK in response to each data packet. Suppose that all packets have TCP and IP headers, as well as a 20-byte link-layer header/trailer. Assume the combined data and ACK packets fully utilize the middle link in both directions and no congestion control is applied.

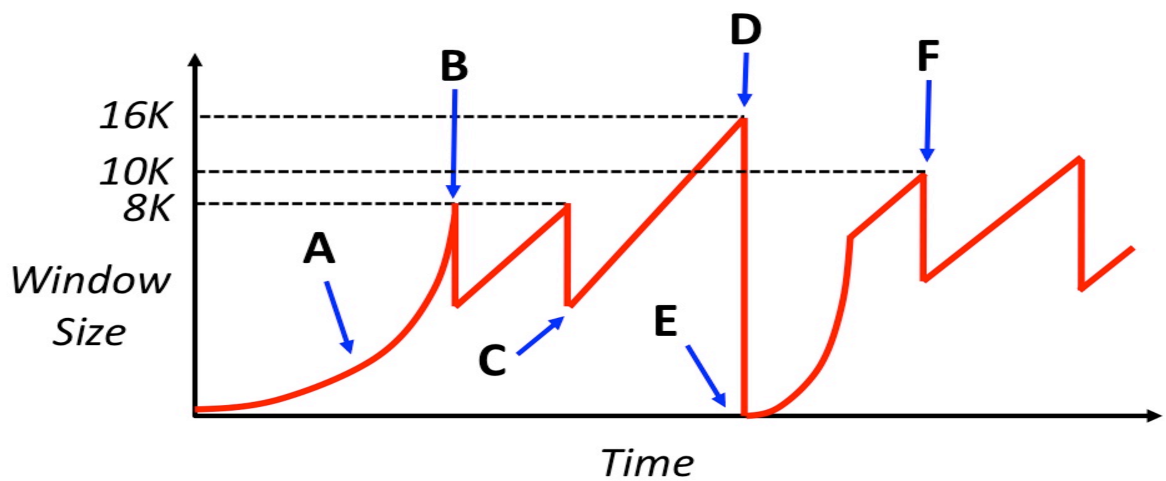
(2a) What fraction of the bandwidth is consumed by data traffic (i.e., the TCP segments, rather than the transport, network, and link-layer information)? Feel free to express your answer as a reduced fraction (e.g., ½ or ¾) rather than a decimal number. Show your work.

(2b) What if the MSS were increased to 1460 bytes? What is the new fraction?

(2c) What if the MSS were increased to 1460 bytes, *and* the receivers apply the delayed-ACK mechanism to send an ACK for *every other* data packet? What is the new fraction?

QUESTION 3: TCP and Congestion Control

Consider the following graph of TCP throughput (**NOT DRAWN TO SCALE**), where the y-axis describes the TCP window size of the sender. We will later ask you to describe what happens on the right side of the graph as the sender continues to transmit.



1. The window size of the TCP sender decreases at several points in the graph, including those marked by B and D.

(a) Name the event at B that occurs that causes the sender to decrease its window.

(b) Does the event at B necessitate that the network discarded a packet (Yes or No)? Why or why not?

(c) Name the event at D that occurs that causes the sender to decrease its window.

(d) Does the event at D necessitate that the network discarded a packet (Yes or No)? Why or why not?

(e) For a lightly-loaded network, is the event at D MORE likely or LESS likely to occur when the sender has multiple TCP segments outstanding? (Write “MORE” or “LESS”)

2. Consider the curved slope labeled by point A. Why does the TCP window behave in such a manner, rather than have a linear slope? (Put another way, why would it be bad if region A had a linear slope?)

3. Assume that the network has an MSS of 1000 bytes and the round-trip-time between sender and receiver of 100 milliseconds. Assume at time 0 the sender attempts to open the connection. Also assume that the sender can "write" a full window's worth of data instantaneously, so the only latency you need to worry about is the actual propagation delay of the network.

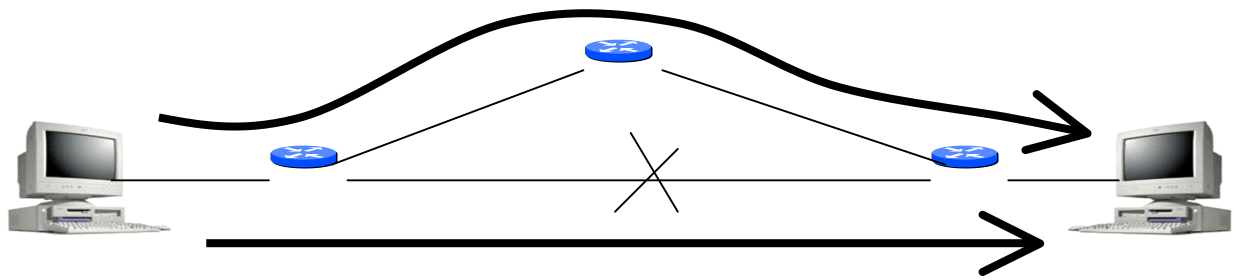
(a) How much time has progressed by point B?

(b) How much time has progressed between points C and D?

(c) How much time has progressed between points E and F?

4. If the sender shares its network with other clients whose traffic traverses the same IP routers, give one explanation for why point D is higher than point B?

QUESTION 4: *Transmission Control Protocol*



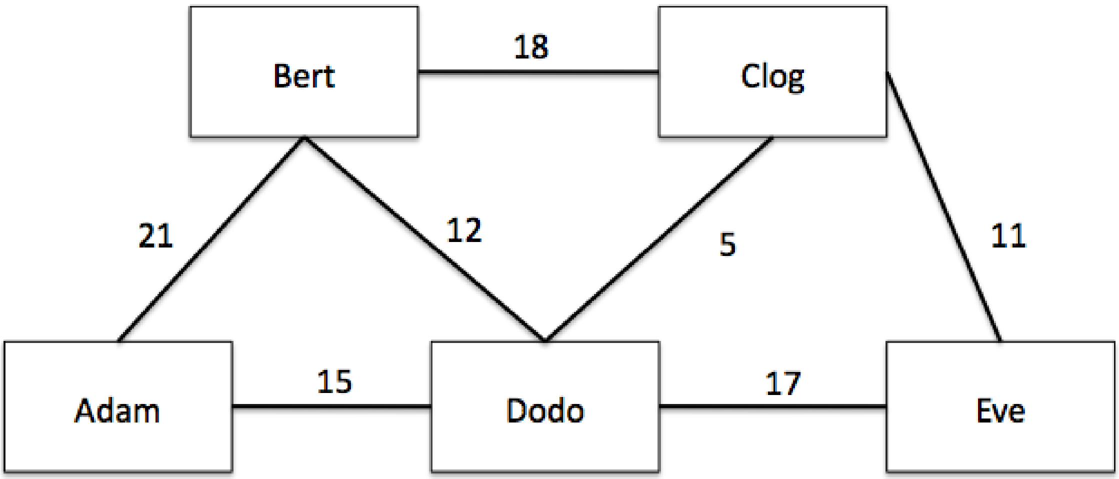
Suppose two hosts have a long-lived TCP session over a path with a 100 msec round-trip time (RTT). Then, a link fails, causing the traffic to flow over a longer path with a 500 msec RTT.

(4a) Suppose the router on the left recognizes the failure immediately and starts forwarding data packets over the new path, without losing any packets. (Assume also that the router on the right recognizes the failure immediately and starts directing ACKs over the new path, without losing any ACK packets.) Why might the TCP sender retransmit some of the data packets anyway?

(4b) Suppose instead that the routers do not switch to the new paths all that quickly, and the data packets (and ACK packets) in flight are all lost. What new congestion window size does the TCP sender use?

QUESTION 5: Distance-Vector Routing

The CS department at Princeton bought new Sun Fire V210 servers. They decided to run a distance-vector protocol for routing between these servers (even though it is a rather small network). They are currently configured as the picture below, with respective edge costs.



The CS staff asked for your help. Write down each step of building the distance-vector routing table for 'Eve' so they can compare it to the output of their implementation. You can use abbreviations e.g., 'A' for Adam and 'E' for Eve.

The initial routing table at node A is:

Destination	*Cost*	*Next Hop*
B	21	B
C	∞	---
D	15	D
E	∞	---

(a) Show the initial routing table of node E:

Destination	*Cost*	*Next Hop*
A		
B		
C		
D		

(b) Show the routing table of node E after one iteration of the algorithm:

Destination	***Cost***	***Next Hop***
A		
B		
C		
D		

(c) Show the routing table of node E after two iterations of the algorithm:

Destination	***Cost***	***Next Hop***
A		
B		
C		
D		

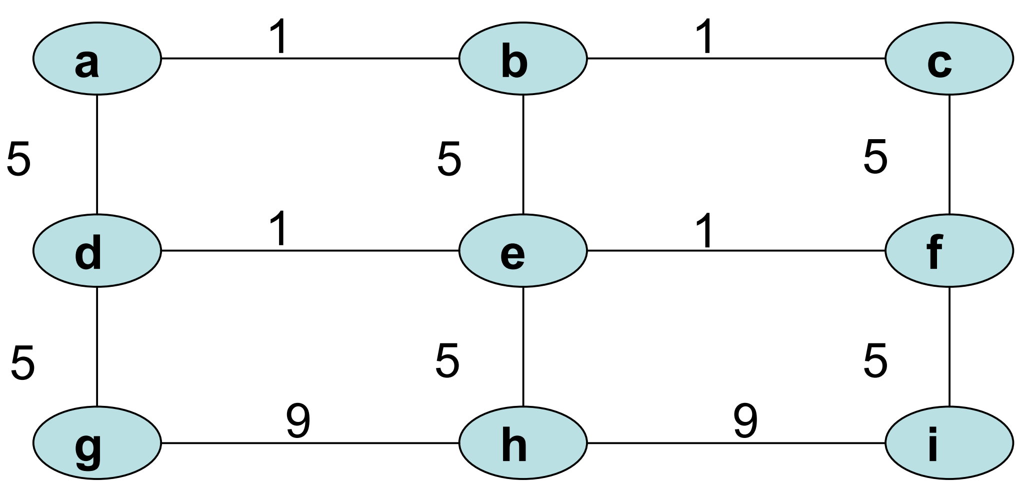
QUESTION 6: *Weighting it Out*

This question explores how to set the (configurable) link weights in link-state routing protocols like OSPF and IS-IS inside a single Autonomous System (AS) to achieve AS-wide goals.

6a) How should the network operators set the link weights if their goal is to minimize the *number of hops* each packet traverses to reach its destination?

6b) How should the operators set the link weights to minimize the *end-to-end delay* the traffic experiences? Assume the network is lightly loaded, so queuing delay is insignificant.

6c) In the picture below, the nodes are routers, the edges are links, and the integers correspond to the link weight on each direction of the link. (That is, the link a-b and the link b-a both have weight 1.) Put arrows on the edges to show the shortest path from every node to the destination node **d**. That is, show the "sink tree" leading to node **d**.



6d) Suppose the link f-e is overloaded with traffic. Identify a single weight change (on *just one link*) that would divert traffic from source f to destination d away from the f-e edge *without affecting the path between* ***any*** *other source-destination pairs*. Avoid any reliance on how routers choose between multiple paths with the same (smallest) cost.