QUESTION 5: Wireless (10 points)
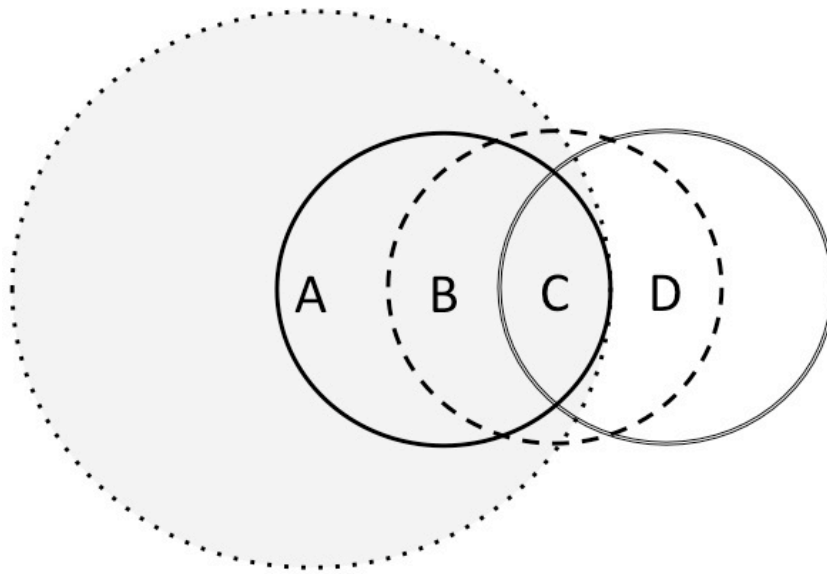

Consider the wireless topology above, comprised of 4 nodes. A has a large transmission range (shown in the dotted, shaded circle), while B, C, and D have smaller transmission ranges (shown by the solid, dashed, and double lines, respectively). Assume that if the transmissions of two nodes' will interfere at a location if and only if they transmit at the same time and their transmission areas overlap. In these problems, assume that losses only occur due to collisions.

A) When node A transmits to node B, list the potential hidden terminals (in either direction, i.e., those who might clobber A's transmission or those who A's transmission might clobber) and exposed terminals.

Transmition A->B, Terminal T
Hidden: T is not in A's range but there exists a node in the ranges of A and T
Exposed: T is in the range of A but B is not in T's transmit range.

Hidden terminals: D – Transmit from D->C will be clobbered by A->B

Exposed terminals: None

B) What about when node B transmits to node C?

Hidden terminals: ==D==

Exposed terminals: ==None==

C) Suppose A is sending data to B and C is sending data to D, both at a constant bit rate equal to the physical capacity of the wireless channel ("as fast as they can").

     i) Assume that no mechanism is used to detect or avoid collisions. What is the throughput of each transfer as a fraction of its send rate? (No math required.)
==A->B: 0 , no throughput, collision from C==
==C->D: 100 %==

     ii)   Now suppose that each node uses CSMA/CA. Again, express the throughput of each transfer as a fraction of its send rate. (No math required.)
==A->B: 100%   A would never sense C==
==C->D: 0%==

     iii) Now assume that we also use an RTS/CTS scheme (like one we learned in class): An RTS is sent if and only if no other RTS or CTS has been heard recently, and the same goes for a CTS. Assume that RTS/CTS exchanges are small compared to data packets and have negligible overhead. Again, express the approximate throughput of each transfer as a fraction of its send rate. (No math required.)
==A->B: 50%==
==C->D: 50%==

QUESTION 2 : TCP and Congestion Control (19 POINTS)

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.
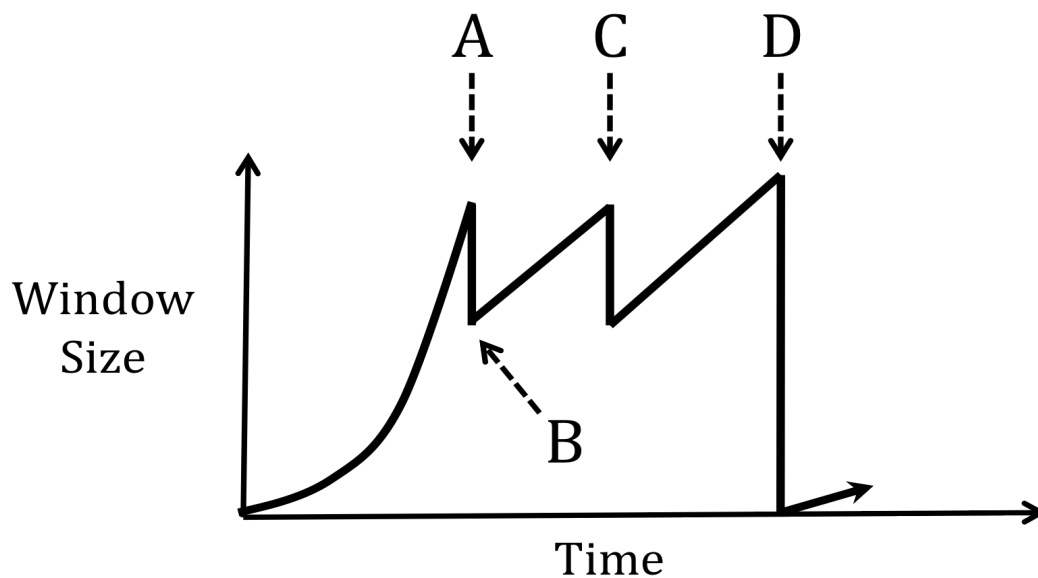
1. [3 points] The window size of the TCP sender decreases at the points marked by A, C, and D on the graph, which happens when a packet belonging to the stream is lost. Name the event which occurs that causes the sender to decrease its window.

    A: Triple Duplicate ACK

    C: Triple Duplicate ACK

D: **Timeout**

2. [4 points] Assume that the network has an MSS of 1000 bytes. If point A occurs 1 second after the sender begins its transfer, and the sender has written 15,000 bytes to the network by that time, what is the round-trip-time (RTT) of the network? 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.

First, calculate the number of RTTs needed. 1 RTT for client handshake (SYN → SYN-ACK). Then window size grows during slow start: 1 MSS, 2 MSS, 4 MSS, 8 MSS = 15 MSS = 15,000 bytes. Note that A occurs at the point of detecting triple duplicate ACK, and therefore the client must have seen at least some ACKs from the 8 MSS windows.

   Hence a total of 5 RTTs over 1 second = 200 ms

3. [4 points] What is the sender's window size (in bytes) at point B?

After triple duplicate ACK, the sender halves its window size (part of "multiplicative decrease" of AIMD), thus leaving us at ½ cwnd of A = 4 MSS = 4,000 bytes

4. [4 points] If point C occurs 2 seconds after point B, what is the sender's window size (in bytes) at point C?

Network RTT = 200 ms, thus 2 seconds = 10 RTT. The sender's window is currently growing linearly in the "additive increase" part of AIMD, and hence increases by 1MSS every RTT.

4

Therefore, our prior cwnd was 4 MSS + 10 MSS = 14 MSS = 14,000 bytes

5. [4 points] Assume that the window size at point D is 16,000 bytes, at which time it drops to 1,000 bytes. How much time will it take for the sender to return to a window size of 16,000 bytes (assuming that there is no further packet loss in the network)? You can express your answer in terms of the number of RTTs, as opposed to the number of seconds.

After a timeout, the window drops to 1 MSS. It then proceeds to grow via slow-start (i.e., doubling each RTT) until it reaches ½ of the cwnd at the time it detected the timeout. Afterwards, it grows linearly (+1 MSS / RTT).

Thus, we experience slow-start until 8 MSS, which takes 3 RTT. We then increase linearly until 16 MSS, which takes an additional 8 RTT, giving us a total of 11 RTTs.

QUESTION 8: Short Answer (29 points)

1) HTTP and content distribution networks

(a) Which scenario benefits the most from using persistent HTTP connections, as opposed to establishing a new connection per HTTP request: Web pages with large objects or pages with small objects? Why? Be specific.

 Pages with small object - % overhead

(b) Most browsers don't have pipelining turned on by default. Ignoring implementation complexity reasons, describe why pipelining might not always give better performance. Be specific.

Browsers wait for stuff in-order causes. head of line blocking

(c) When browsing web sites that use a CDN such as Akamai, most users at Princeton notice better performance. But a friend from Stanford who is visiting Princeton reports that these same web sites appear SLOWER than many other sites, and he's confounded as to the reason. What's the likely culprit of this performance?

He has a static DNS entry for Stanford that redirects him to Akamai servers by Stanford and not by Princeton

3) Using TCP, a sender has sent out packets 1 through 20. The sender receives an ACK for packet 10 (signifying that the destination received packet 10), then triple duplicate ACKs for packet 11. Given this result:

(a) Which of the 20 packets can the sender assume are definitely lost, if any?
11

(b) Which of the 20 packets can the sender assume were definitely received?
1-10

Now, the same sender resends the next missing packet and receives an ACK for packet 16 in response. Based on this information:

(c) Which of the original 20 packets can the sender assume are still definitely lost, if any?
You don't know which ones were lost

(d) Which of the original 20 can the sender assume were definitely received?
1-16

QUESTION 1: Internet Versus Station Wagon (15 POINTS)

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.

*By car, travel time is 2 hours (i.e., 200 km * 1hr/100km). During two hours, the link could transfer 800,000 bits/sec * 60 sec/min * 60 min/hour * 2hr * 1byte/8bits for **720,000,000 bytes**.*

(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.)

*An MSS of 512 bytes and two 20-byte headers leads to a packet size of 552 bytes. So, only 512/552 of the bandwidth is used to transmit the actual data, so the transfer takes 552/512 of the time, or 40/512 times longer. This reduces to a factor of **5/64** longer time.*

(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.)

*The receive window needs to be large enough to accommodate a round-trip time of data. The one-way delay is 1sec/200,000km * 200km, or 0.001 seconds (or 1 msec). So, the round-trip time is 0.002 seconds (or 2 msec). Since the link bandwidth is 800,000 bits/sec, the total number of bytes transmitted during a round-trip time is 0.002 sec * 800,000bits/sec * 1byte/8bits, or a grand total of **200 bytes**.*

QUESTION 4: A routed and bridged network (14 POINTS)
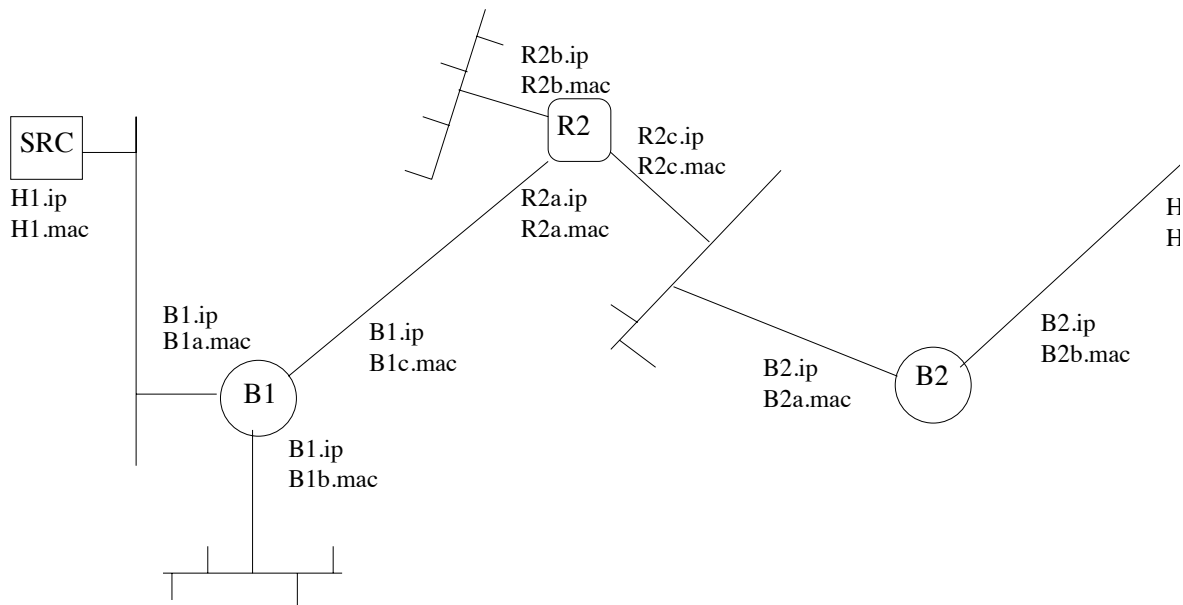
Above is a picture of a network with 2 bridges and 1 router. Each interface is labeled with both an IP address and a MAC address. Imagine that host H1 is sending a packet to host H2. Please answer the following questions about this figure:

4A. How many (datalink) networks are shown above?

3 -- The networks hanging off of R2

4B. Just before the packet reaches bridge B1, what is its layer 2 destination?

R2a.mac

4C. Just before the packet reaches bridge B2, what is its layer 2 source?

R2c.mac

4D. Just after the packet leaves router R2, what is its layer 3 source?

H1.ip

4E. When H1 sends out an ARP query, what is the reply to that query?

R2a.mac has R2a.ip (or just "R2a.mac")

4F. Does the entry B2a.mac appear in B1's forwarding table?

No. B2a is on a different datalink network than B1.

4G. Will SRC's Ethernet device receive packets with destination R2a.mac? Why / why not?

This turned out to be a more confusing question than we intended, depending on how you interpreted the question. We accepted both "yes" and "no" answers here, but your explanation why had to be correct:

No -- because B1 is a (learning) bridge and any packets to R2a sent by nodes in on the "B1b.mac" or "B1c.mac" broadcast domains will be filtered by B1, so will only be re-broadcast out the B1c link.

Yes -- if the sender is in the same local broadcast domain (the "B1a.mac" domain) as SRC, then SRC will receive those packets.

Yes -- if B1 or R2 has just come online and B1 hasn't yet learned local MAC addresses of nodes, then it could forward packets out all interfaces.