University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Sciences

EE122
MIDTERM EXAMINATION
Wednesday, 23 October 2013
Sylvia Ratnasamy

INSTRUCTIONS—READ THEM NOW! This examination is CLOSED BOOK/CLOSED NOTES. You will not require a calculator, iPhone®, laptop computer, or other calculation aid. Please put them away right now! You MAY use one 8.5"x11" double-sided crib sheet, packed with notes, formulas, and diagrams (but no smaller than 8pt font). All work should be done on the attached pages, and there are four blank pages at the end for you to use as scratch paper. Note that the sub-questions within each question are arranged in increasing order of difficulty so if you find yourself stuck, consider moving on to the next question.

In general, if something is unclear, write down your assumptions as part of your answer. If your assumptions are reasonable, we will endeavor to grade the question based on them. If necessary, of course, you may raise your hand, and a TA or the instructor will come to you. Please try not to disturb the students taking the examination around you.

Please write your SID on each page!

Name – Please Print!

(Signature)

SID

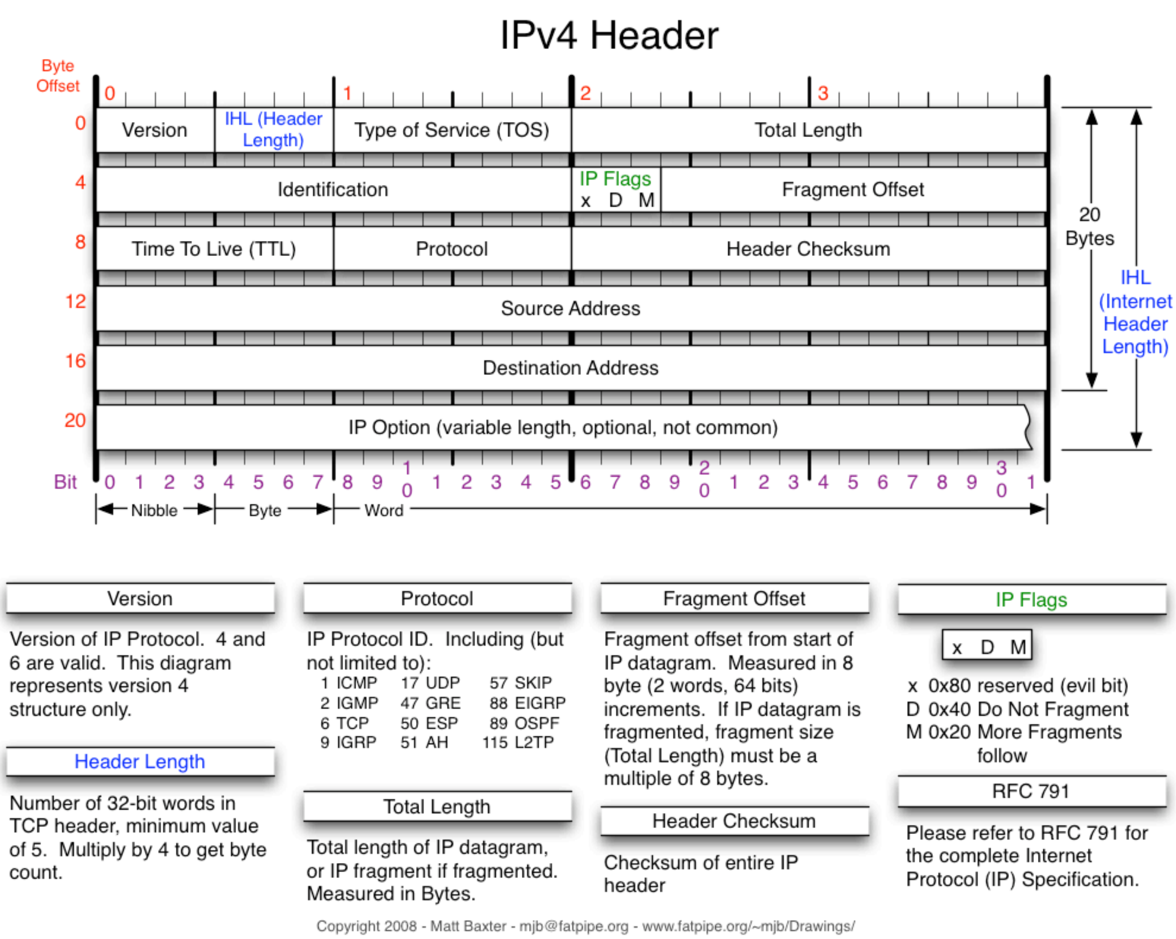
Discussion Section (Day/Time)

Question		Possible Points	Points Obtained
1	General Multiple-Choice	20	
2	Design Decisions	15	
3	BGP	10	
4	Routing (1)	10	
5	Routing (2)	5	
6	Reliable Communication and Congestion Control	15	
Total		**75**	

Fun Facts to Remember

The IP header, without options, is 20 bytes
The TCP header, without options, is 20 bytes
The UDP header is 8 bytes.

1 TB = 10^{12} bytes; 1 GB is 10^{9} bytes; 1 MB is 10^{6} bytes
1Gbps is 10^{9} bits/sec; 1Mbps is 10^{6} bits/sec
1 msec is 10^{-3} seconds



1. General Multiple-Choice [20 points]

[Notes: Correct answer in red.]

Indicate your answer by circling

1. Poison-Reverse eliminates the counting-to-infinity problem (*Circle one*)
   a. True
   b. ***False***

2. The number of addresses represented by the IP prefix 12.3.4.0/24 is: (*Circle one*)
   a. 64
   b. ***256***
   c. 16 million
   d. 6

3. Four flows request bandwidth allocations of 1Mbps, 4Mbps, 8Mbps and 12Mbps. The capacity of the link is 12Mbps. Which of the following allocations are max-min fair?: (*Circle one*)
   a. 3Mbps, 3Mbps, 3Mbps, 3Mbps
   b. 2Mbps, 3Mbps, 3.5Mbps, 3.5Mbps
   c. ***1Mbps, 3.66Mbps, 3.66Mbps, 3.66Mbps***
   d. 1Mbps, 4Mbps, 7Mbps, 0Mbps

4. MIMD (Multiplicative Increase Multiplicative Decrease) always converges to a fair allocation of bandwidth between competing flows: (*Circle one*)
   a. True
   b. ***False***

5. Packet switching requires routers to maintain state for each TCP connection (*Circle one*)
   a. True
   b. ***False***

6. The primary purpose of transport-layer ports is to: (*Circle one*)
   a. Allow the network to demultiplex packets between different endhosts
   b. Allow the network layer at endhosts to demultiplex incoming packets across different transport layer protocols
   c. Speed up packet processing within the operating system (OS) at the endhost
   d. ***Allow the transport layer to demultiplex packets across different applications***

7. Host A is transferring a file to host B over a path with a RTT of 100ms and a bottleneck bandwidth of 100Kbps. A has a fixed sliding window of 1KBytes. Hence, the maximum throughput A can achieve to B (ignoring packet headers) is: (*Circle one*)

a. 10Kbps
b. ***80Kbps***
c. 100Kbps
d. 10/8 Kbps

8. It is possible to build a reliable file transfer application on top of UDP (*Circle one*)
   a. ***True***
   b. False

9. Routing protocols such as RIP (distance-vector) and OSPF (link-state) are typically implemented at: (*Circle one*)
   a. ***The control processor of each router in a domain***
   b. The control processor at only the border routers of a domain
   c. The line-cards of each router in a domain
   d. None of the above

10. In IPv4 routers may be expected to fragment packets that exceed the link MTU (max. transmission unit). In IPv6, however, routers do not implement fragmentation. The rationale for this change is an example of which principle in action: (*Circle one*)
   a. Modularity through layering
   b. ***The end-to-end argument***
   c. Robustness by eliminating state in routers

11. The primary benefit of the narrow waist of the Internet is: (*Circle one*)
   a. Modularity
   b. Scalability
   c. ***Interoperability***
   d. Efficiency

12. If routers implement fair-queuing, there is no additional benefit to implementing congestion control at the end-hosts (*Circle one*)
   a. True
   b. ***False***

13. The problem with input queued routers is: (*Circle one*)
   a. They require an interconnect fabric with capacity N times the output line rate (where N is the number of router ports)
   b. ***They suffer from head-of-line blocking***
   c. They require extra memory for buffering packets
   d. They require complicated packet classification capabilities

14. Which of the statements below are true (*Circle ALL that apply*)

a. Layering advocates implementing reliability at the ends
b. ***Layering trades off modularity for efficient implementations***
c. Layering aims to improve routing convergence
d. ***Layering simplifies the development of applications***

15. A client running TCP Vegas can establish a connection to a server running TCP newReno because: (*Circle one*)
    a. IETF protocol specifications spell out the requirements for interoperability
    b. TCPs Vegas and newReno only differ in how the receiver computes the advertised window
    c. ***The actions of the receiver are the same in TCP Vegas and newReno***
    d. There's no difference between TCP Vegas and newReno

16. In the Internet's layered architecture, network layer functionality is implemented at: (*Circle one*)
    a. Routers
    b. Endhosts
    c. ***Both routers and endhosts***

17. Which of the following is true? The aggregation of multiple IP addresses into a single prefix: (*Circle one*)
    a. ***Reduces the number of routing entries but complicates the route lookup process***
    b. Reduces the number of routing entries and simplifies the route lookup process
    c. Increases the number of routing entries but simplifies the route lookup process
    d. Reduces the number of routing entries and has no effect on the route lookup process

18. Which of the following is true? (*Circle one*)
    a. Multi-homing aids address aggregation
    b. ***Multi-homing complicates address aggregation***
    c. Multi-homing has no impact on address aggregation

19. A routing change has the potential to adversely affect which of the following TCP mechanisms. (*Assume no packets are dropped due to the routing change. Circle all that apply*)
    a. ***RTT estimation***
    b. Flow control
    c. ***Congestion control***
    d. Initial Sequence Number exchange

20. Assuming circuit switching, if the network has allocated exactly enough bandwidth to handle the application's peak bandwidth P, and the application has average bandwidth A, the level of utilization of that application's circuit is given by (where / denotes division):
    a. ***A/P***
    b. P/A
    c. (P - A)/A
    d. (P - A)/P

2. Design Decisions [15 points]

[Notes: We assigned points for correctly selecting or omitting each option as appropriate. Additional explanation inserted as needed.]

A challenge in networking is that networks are built by, and for, parties with very different concerns and incentives. In this question, we'll consider system design decisions from the viewpoint of each party. In each case, circle the design decision you'd agree with.

2a) Viewpoint: Content Provider

You are the provider of a news channel that serves short 100Byte messages to a user population that is evenly spread across the globe. You plan to build a service in which a client opens a connection to one of your servers, requests one 100Byte message and terminates the connection after it has received the message. You get lots of (sometimes conflicting) advice on how to build a system that delivers messages both reliably and quickly. Assume the TCP MSS and network MTU are greater than 100Byte and that the size of the request is less than one MSS. Circle the advice that you agree with: (*Circle all that apply*)

1) ***It is preferable to invest in k servers spread across the globe each with a bandwidth serving capacity of C, than in a single server with a bandwidth serving capacity of kC***
   [This is a good idea because, for short messages, performance will be limited by latency not throughput and hence distributing servers makes sense.]

2) Sign up with multiple provider ISPs; for each user U, route messages to U via the provider ISP that advertises the shortest BGP path to U. This is guaranteed to ensure packets reach U with the shortest delay.
   [No. BGP path length needn't correspond to delay.]

3) Your users will enjoy faster downloads if you make the following adjustment to your TCP implementation: trigger fast retransmits after one (instead of the standard three) duplicate ACK
   [No. We only have one packet to transfer per connection, so there's no possibility of duplicate ACKs and hence this can't help.]

4) ***The default value for a TCP timeout is 500ms. Setting the default to 100ms can only improve a user's download speed. (Ignore any additional overhead introduced at the server due to the potentially larger number of timeouts)***
   [Yes. If we ignore any potential overhead at the server side, there's no downside and can only improve performance for the downloading clients since lost packets will be

retransmitted faster. Again, because there's only one packet to send, timeouts are the only mechanism that will trigger retransmissions and hence this will help.]

5) ***The packet loss rate of the network is less than 1%. So, don't bother with TCP and instead just build your application on top of UDP. Clients send one UDP packet to request a message and your server responds by transmitting the requested message thrice (using three separate UDP packets). If the client does not receive the message, it will simply resend its request. On average, this approach will result in faster downloads for users and will transmit no more packets than TCP would.***
[Yes. Performance for the client is likely to improve since the client effectively experiences lower loss (due to the 3x copies) and faster downloads (since the UDP-based exchange avoids the RTT due to the TCP connection establishment). This will require no more packets than TCP once you factor in the handshake packets (SYN, SYN-ACK) required for TCP connection establishment.

2b) <u>Viewpoint: Application Developer</u>

Your music collection is stored on a computer in your parents' house. You are developing an application that streams audio from this computer (at your parents' house) to your laptop in 155 Dwinelle. Given your now considerable expertise with networking, the following are some design possibilities that come to mind. After more careful consideration, which ones do you think are correct? (*Circle all that apply. Ignore any issues with firewalls. Remember that streaming audio applications don't require reliable delivery.*)

1) Build your application on top of UDP. Your application simply packetizes the audio stream and sends each packet over UDP. You don't need to worry about picking a rate to transmit at because ISPs today deploy fair queuing and this ensures no flow consumes more than its fair share of network bandwidth.
[No. ISPs today do not apply fair-queuing to individual flows for regular Internet traffic. Moreover, even if fair-queuing were deployed for such use, it does not by itself solve the question of what rate the client should select. Fair-queuing will limit the flow to its fair share of the link but with no feedback in the protocol, any traffic the client sends in excess of its fair-share will be dropped.]

2) ***Build your application on top of UDP. Your application simply packetizes the audio stream and sends each packet over UDP. You do need to figure out what rate to transmit at but the TCP throughput equation (with perhaps some tips from your helpful EE122 TAs on how to dynamically measure network loss and delay) will help you determine a good rate.***

[Yes. We discussed this exact use case in class ☺]

3) Instead of messing with equations, build your application on top of TCP. Since you don't need the reliability, simply modify the receiving end of the application to not transmit any acknowledgements.
[No! Without ACKs, TCP throughput will effectively drop to zero.]

4) Cache a copy of your music collection on your EECS instructional machine (the machine admins approve of this use!) and stream it from there instead of your parents' machine. The path between your EECS machine and Dwinelle has higher bandwidth and lower RTT than that between your parents' house and Dwinelle. On average, this will improve the performance of your application and will certainly make your parents happier.
[Yes. Most students got this right.]

5) You occasionally experience 30 second periods where your application receives no audio packets. You discover this is due to BGP problems along the path between your parents' machine and 155 Dwinelle. You thus decide to cache a copy of your music collection as per the previous recommendation (#4) and switch to downloading from your instructional machine during periods of high loss. This will fix the problem.

[Yes because your instructional machine and Dwinelle are in the same UCB domain and hence won't suffer BGP-related problems.]

2c) <u>Viewpoint: Router vendor</u>

You are charged with developing the next-generation high-speed router. Consider each of the proposed architectural changes listed below and determine whether it makes your task easier, harder, or has no impact on your task. (Circle one option for each.)

1) Eliminate options from the IP header [***Easier***] [Harder] [Doesn't matter]
[Easier because less (currently troublesome) processing for the router.]

2) Switch to a rate-based congestion control protocol such as RCP [Easier] [***Harder***] [Doesn't matter]
[Harder because routers have to calculate per-flow fair-share allocations.]

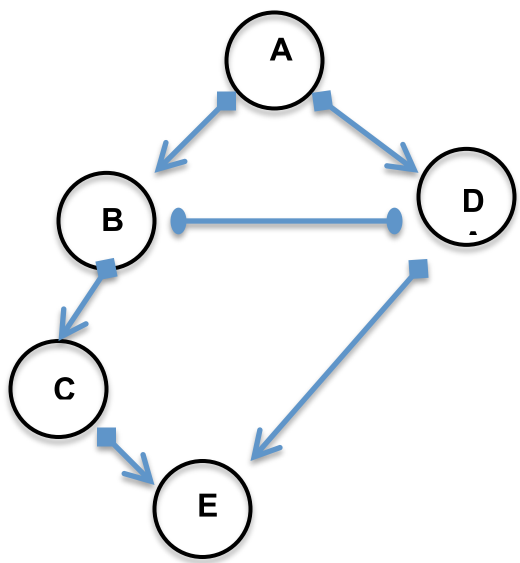
3) Endhosts have infinite buffer memory at the transport layer [Easier] [Harder] [***Doesn't matter***]
[Doesn't matter because it doesn't impact any functionality currently implemented in routers.]

4) Replace UDP for TCP as the de-facto transport protocol [Easier] [Harder] **[*Doesn't matter*]**
[Doesn't matter because it doesn't impact any functionality currently implemented in routers.]

5) Packets carry a "source route". A source route is a fixed-length ordered list of all the routers along the path between a source and destination. [*Hint: consider how a router performs a forwarding lookup given a source route.*] **[*Easier*]** [Harder] [Doesn't matter]
[Easier because the next-hop lookup operation is simpler with source-routes (requires simply indexing into the packet header) than with current IP addresses (longest-prefix match on a large routing table stored in the router).]

3. BGP [10 points]

Consider the following network of Autonomous Systems. Each circle represents an AS. Providers are higher than customers and ASes B and D are peers. Assume the standard Gao-Rexford peering arrangements hold and assume that if an AS has multiple routes through customers that it will choose the one with the least number of ASes along the path.



3a) For destination E, which (if any) path would be advertised by B?

B-C-E

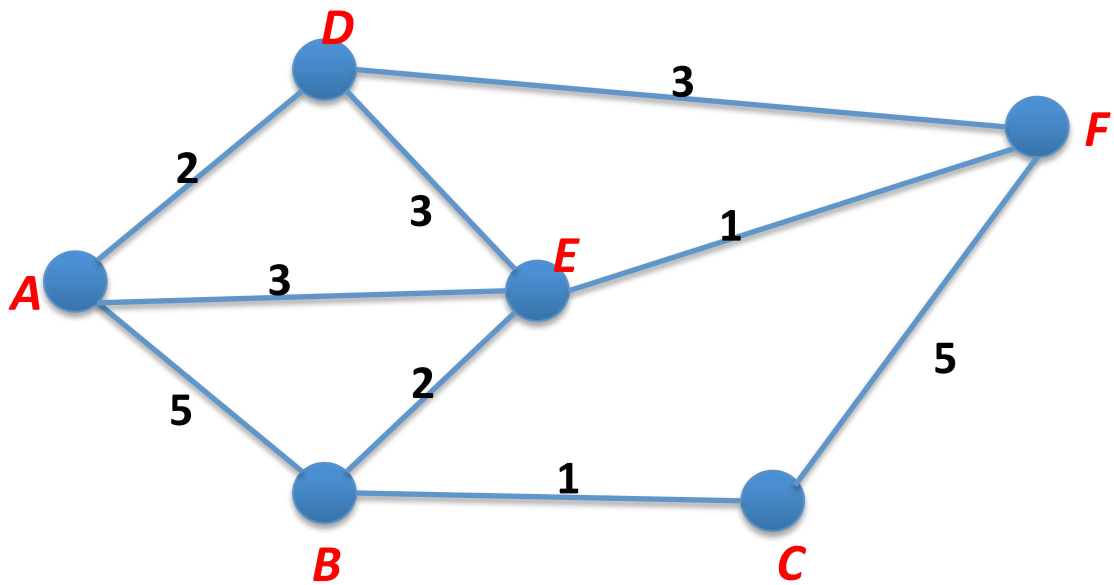
3b) What path will be used by A to reach E? Why might this path not be the shortest path in terms of number of router level hops?

A-D-E, Shortest in terms of #ASes need not mean shortest in #router-hops.

3c) If link D-E fails, what path will D use to reach E?
D-B-C-E

3d) If link D-E fails, what path will A use to reach E?
A-B-C-E
3e) If both A-D and B-D fail, can C use the path C-E-D to reach D? Why or why not?
No, not valley free; E won't advertise D

4. Routing [10 points]



Consider the network depicted above.

Assume that a simple distance vector protocol (without poison reverse) is used to route packets and that it has converged to the correct topology. The table below shows the distance and next-hop to F from each node at this point.

	Before increase (distance, next hop)
A to F	4, E
B to F	3, E
C to F	4, B
D to F	3, F
E to F	1, F

Assume that routing messages and forwarding table updates are synchronous (i.e. they happen at the same time on all nodes). Remember that while the protocol is converging, the forwarding tables will not necessarily be correct. If you want partial credit for incorrect answers be sure to explain how you arrived at your result. Assume routers break ties between equal cost paths by picking the next-hop router with the lower ID.

4.1) Using distance vector <u>without</u> Poison Reverse, what is E's second best path to F?

E-B-E-F

4.2) Now the link cost between nodes E and F increases from 1 to 6. Once E detects the change in link cost, what distance to F does E advertise? Fill in the routing entry (distance and next-hop) that E's neighbors compute for destination F after processing E's update.

E's advertised distance to F is ___**5**_

	After E's update (distance, next hop)
A to F	***5,D***
B to F	***5,C***
D to F	***3, F***

4.3) Continuing with the previous question, what distance does B advertise for destination F in the next iteration? Fill in the routing entry (distance and next-hop) that B's neighbors compute for destination F after processing B's update.

B's advertised distance to F is ___**5**_

	After B's update (distance, next hop)
A to F	***5,D***
C to F	***5,F***
E to F	***6,D***

4.4) Continuing with the previous question, what distance to F does C now advertise? Fill in the routing entry (distance and next-hop) that C's neighbors compute for destination F after processing C's update.

C's advertised distance to F is _**5**___

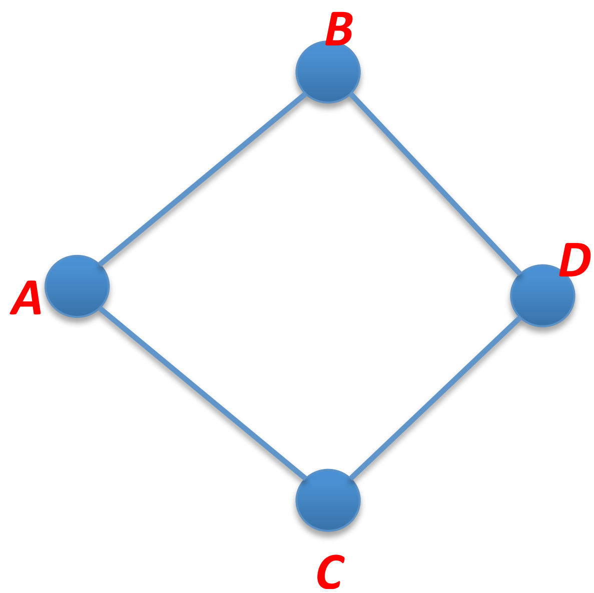
	After B's update (distance, next hop)
B to F	***6, C***

Are all routers now following the correct shortest path? ***Yes*** No

4.5) Will the routing tables converge faster if we use Poison Reverse?

Yes ***No*** *[Try it with Poison Reverse to see why!]*

5. Load-Sensitive Routing [5 points]



Consider the above network, where A sends traffic to D. Assume the routing protocol computes the shortest path, where the cost of a link is based on its traffic load (i.e., the amount of traffic on the link). The traffic from A to D is the only traffic in the network. Which route is traffic from A to D going to use? Is the route stable? Explain why or why not.
[*While the initial starting point does not affect your answer, you may assume that the initial link costs (i.e., before traffic flows) equal one and that A breaks ties between equal cost paths by routing through the neighbor with the lowest ID (alphabetical order). You may further assume that the interval between routing updates (and hence changes in routes) is much longer than the path RTTs.*]

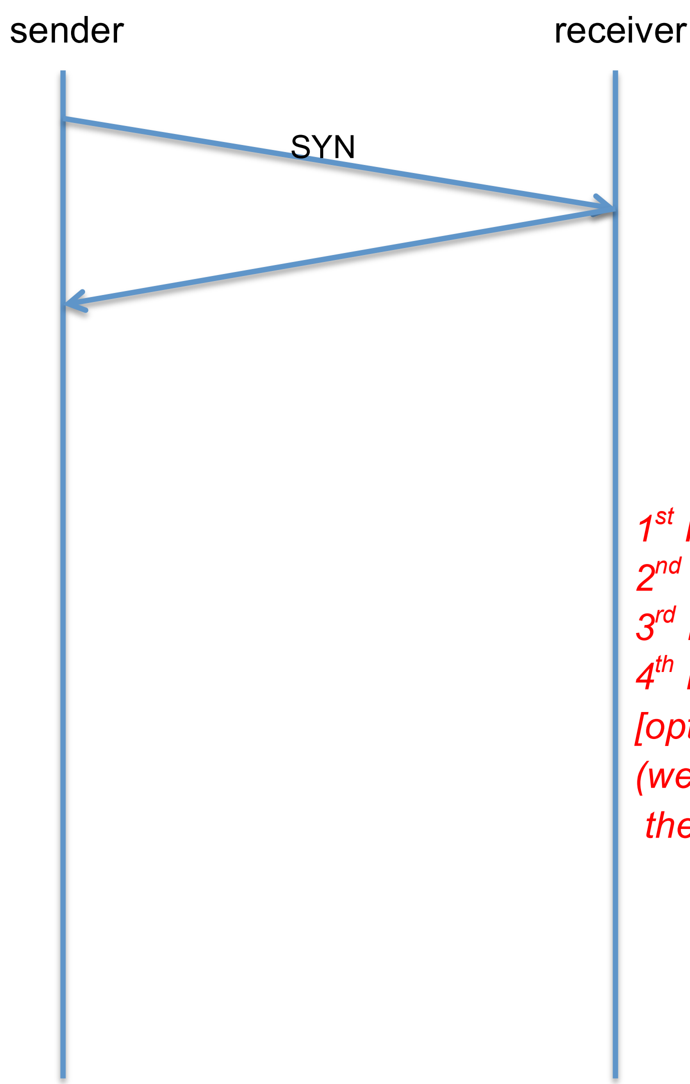
Not Stable. Route will oscillate between A-B-D and A-C-D because the metric itself is dynamic, dependent on load.

6. Reliable Communication and Congestion Control [15 points]

[Tip: No detailed numerical calculations are required to answer any of the questions in this problem!]

6a) A sender and a receiver are communicating using a TCP-based protocol in which data is sent immediately after connection establishment (i.e., the first data payload is sent with the final ACK in the three-way handshake of connection establishment). The bandwidth of the link is 100Mbps, the round-trip time is 10ms and the maximum packet/segment size is 1000 bytes (ignore header overhead). Assume the initial congestion window is set to one MSS, that the receiver has ample buffer space and the initial value of SSTHRESH is high (e.g., 64,000Bytes). Assume default values for TCP parameters.

Draw a packet exchange diagram showing how many RTTs it takes to transfer 7,000 bytes. (To get you started, we've drawn out the first packet sent below.)



1st RTT: SYN+SYN-ACK
2nd RTT: 1000B of data + ACK
3rd RTT: 2000B of data + ACKs
4th RTT: 4000B of data + ACKs
[optional] 5th RTT: FIN handshake
(we gave full credit whether or not the FIN handshake was included)

6b) The fast retransmit optimization is used to avoid a timeout. Consider the TCP transfer in the previous question (6a).

If packets were lost during that transfer could fast retransmit ever come into play? (*Circle one*)

YES NO

Possible for the 4th data packet (4000-5000) because the CWND=4, there are 3 packets in flight after this packet.

The fast recovery optimization that temporarily inflates the congestion window is designed to allow TCP to keep packets flowing after a loss. Could fast recovery ever improve performance for the above transfer? (*Circle one*)

YES ***NO***

No, there aren't enough packets in flight to actually inflate the CWND (and no additional packets to send)

6c) Suppose after sending a series of packets,1..n, TCP does not receive any acknowledgements for an extended period of time. Normally, after a certain timeout period is reached, TCP will reset the congestion window to 1, and will retransmit the first packet (packet 1). Suppose instead, that TCP were modified to send the $(n+1)^{th}$ packet in this case. Under what conditions would this be a good idea and why? (Be specific.) Under what circumstances would this be a bad idea and why? (Be specific.)

Good idea: If packet one (and perhaps later packets as well) were received. It would prevent unnecessary retransmission and the next ACK would indicate which, if any, packet actually needed retransmission. Possible if the reverse path saw high loss (of ACKs) and fewer ACKs need to be lost to allow a timeout to occur.

Bad idea: If packet one was actually lost and the receiver buffer is full, the (n+1)th packet will have to be dropped.

[Note: terser answers – e.g., "all ACKs were lost" are fine. We gave full credit to many variants that got the general gist.]