CSCI 356 – Computer Networking – Final Exam Review – Spring 2017

Topics: Everything we have covered, from top to bottom.

- Upper layers & applications: Sockets, clients & servers, Web, HTTP, DNS, caching, replication, performance
- Transport layers: UDP, TCP, and all related mechanisms
- IP layer: addresses, classful routing, CIDR, distance vector & link state routing protocols, BGP & hierarchical routing, etc.
- Link & physical layers: Ethernet, mac addresses, CSMA/CD, frames & headers, Manchester encoding, repeaters, hubs, switches, broadcasts, forwarding tables, collisions, etc.

However, the exam will be heavily biased towards more recent material, i.e. IP and lower layers of the protocol stack, along with firewalls and some of the security topics we covered.

You do not need to memorize exact numeric details (e.g. the layout of header fields, or the specific numeric parameter values used for EWMA). You will be given a reference sheet showing the header layout for TCP segments, UDP datagrams, IP packets, and Ethernet frames.

Below are a few review questions culled from various sources. I do not have solutions for these, but am happy to discuss by email or in person before the exam. For additional exercises, see the review sheets for both earlier exams and the textbook. This review sheet is not comprehensive.

[Some questions courtesy Lynchburg, UCSD, others.]

(1) Which of the following are guaranteed under IP?
- All packets sent will arrive.
- Assuming packets arrive, they will arrive in the same order that they are sent.
- Packets will not be duplicated, i.e. a packet sent once will not arrive more than once.
- IP uses the IP address stored in each datagram to route the packet.
- IP packets will contain both source and destination IP addresses.
- IP uses the port number stored in each datagram to route the packet.
- Packets contain uncorrupted data.
- Arriving packets (assuming uncorrupted) will have the same size as when they were sent.
- After the connection is established, all packets for the connection all take the same route.

(2) Same question as (1), but for TCP.

(3) Is there a maximum number of routers that a packet can pass through on its way to its destination? Why or why not?

(4) How can we tell a class B network address from a class C network address?

(5) Suppose that we want to split the 123.132.23.0/24 network into 16 subnets. How many hosts could we put on each subnet? What would the addresses of the subnets be?

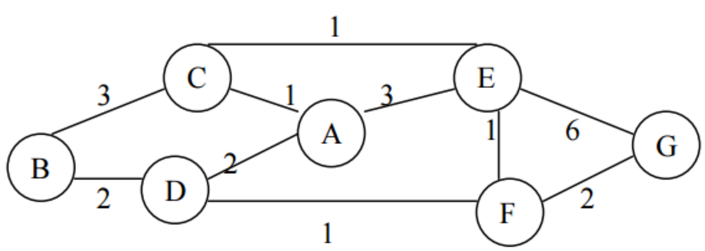
(6) Each of the following IP addresses is special or unusual in some way. Explain how.
- 0.0.0.0

- 127.0.0.1
- 0.0.0.18
- 255.255.255.255
- 161.115.255.255

(7) Suppose host 161.115.144.19/16 wants to send a message to host 161.115.144.120/16. What is the minimum number of routers the packets have to pass through? How is this determined?

(8) For each of the following, say whether it occurs (or may occur) as part of distance vector routing, link state routing, and/or flooding
- Perform Dijkstra's all-pairs shortest path algorithm
- Shortest paths are always followed.
- Longest paths are always followed.
- Only neighbors exchange information about their routes.
- Non-neighbors exchange information about their routes.
- Routes are determined by tables maintained in routers.
- Each router learns the complete topology of the network.
- The count to infinity problem might be encountered.
- The procedure must be repeated multiple times to generate good results.
- Neighbors might lie about the best routes they know.

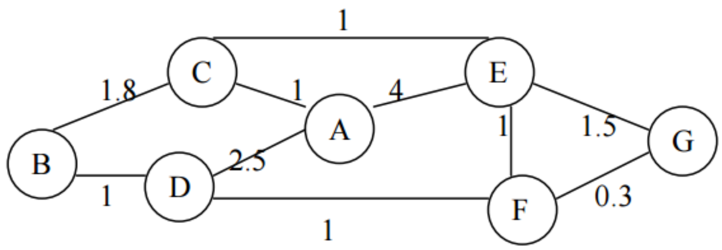
(9) Consider the following network…



a.) Show the data that node A will receive on the first iteration of the distance vector routing algorithm.

b.) Show the routing table for node A after the first iteration of the algorithm has been completed.

(10) In the following network, use Dijkstra's all pairs shortest paths algorithm to determine the shortest paths to node A. Record the order in which the nodes are made permanent along with the next hop and the distance to the destination.



(11) For the network shown, complete the table below showing how the link-state algorithm builds the routing table for node D.