1) (4 points) In class we talked about the 3 army problem with regard to TCP close and how it would require an infinite number of fin messages to be sure both sides agreed to close. The same problem occurs with regard to TCP opens, yet a 3 way handshake has been chosen as a practical solution. A two way handshake could be potentially even more practical in terms of saving packets. Think about routing on the internet and suggest why a 3 way handshake is used rather than a 2 way or 4 way handshake.

One reason is that the return route might not be good and with a 2 way handshake, the contacted host might start sending data on a connection that is not up because the originating host has not received a response.

2) (5 points) We discussed on the class email list the fact that the “all zeros” address of any subnet is not allowed as a valid host address since it used to be a broadcast address. Since IP address allocation is very tight, you have the idea of changing the rule. You do some testing and find that most systems can handle such a host address. But how many IP addresses would you reclaim? Outline a procedure to determine how many that would be. Would you be able to count all of them?

The issue is that old style broadcasts and the new style broadcast addresses are both being reserved, i.e. not allowed as host addresses. Since everything uses the new style, you could potentially put the old style broadcast addresses back to work as host addresses. To figure out how many host addresses you would reclaim, you need to figure out how the address space is split up and how many networks there are on the Internet, since you will gain one host address per subnet. Under the old class based routing, that would be easy. But with classless routing, it is harder. One place to start would be the size of the BGP table.

4) (6 points) A computer is sending full TCP windows of 65535 bytes over a 40 Mbit DS3 connection that has a 20 msec one-way delay. What is the maximum throughput achievable? What is the line efficiency?

The simple solution is: It sends 65,535 bytes, which takes \((8 \times 65,535)/40,000\) (kbit/sec) = 13 ms followed by a 20 ms Ack. For 13 of every 33 ms the link is sending data, which is a utilization of 39%.
The more complex answer would be to add in sliding window delays like in the homework problem since I did say TCP windows although I did say “full TCP windows.” Generally for my exams, provide the simple answer or ask me in class.

5) (10 points) For the homework problem about finding fully redundant routes, I provide the solution “run Djikstra, mark those routes unavailable and run Djikstra again.” For the following network, this would not provide two full routes from A to F, although it is clearly possible. Describe an improvement, which will handle similar situations. Would this be possible to run on a large network?

Do something like find all routes A to F, then find the disjoint sets which have no common elements. This would be hard in a large network due to the additional calculation beyond dijkstra.

6) (12 points) Answer the following questions concerning the differences between Link State, Vector and Vector Path routing protocols
Which generates more network traffic in a large network? Link State

Which protocol uses the least router memory? Vector

Which protocol handles link additions better than failures? Vector

Which protocol is typically used for 802.3 LAN networks? Vector

7) Some True False questions about TCP flow and routing

8) A design problem on what routing protocols to use for various networks.