1. What 3 types of policies can be configured into a BGP router, and what might each be used for?

2. Gossip Protocol: (since none of you understood it on the quiz, let’s try again). We want a protocol such that if replica i talks to replica j, i and j “pool their knowledge” so that i won’t need to check with anyone j has recently checked with, and vice versa. Also, there are “updates” which are things like “change memory location 20392819 to 55141”. You can assimilate that into your database, but you need to keep it in its original form until you know all the replicas have seen it. Otherwise, you’ll have to actually send the entire database to a replica that hasn’t seen that update. That’s the purpose of time2...so that you’ll know when it’s safe to discard an update.

Each keep a data structure that looks like:
(replica1, time1, time2)
(replica2, time1, time2)
...
(replican, time1, time2)

If I have time1=x for replica k, that means I have all replica k’s updates before time k.

If I have time2=y for replica k, that means I know that replica k has seen all updates from any replica before time=x.

So, i and j decide to synchronize. They start by exchanging their data structures. Suppose i sends (replica2, 3 PM, 1 PM) and j sends (replica2, 2 PM, noon). What happens with updates that originated at replica2? How will each of i and j update time1 and time2 in (replica2, time1, time2).

So the exchange consists of sending the data structures, sending relevant updates, and updating all the time1’s and time2’s as a result.

After i and j synchronize with each other, which updates can i throw away?

How does it affect things if one replica has its clock set at, say, 2 hours earlier than any of the other replicas?

How does it affect things if one replica is down for say, a day?

What heuristics should a replica use for which other replica it should synchronize with? Suppose there are a bunch of replicas in Europe and a bunch in US. What kind of protocol optimizations can you do that will make it likely that only a single replica will make a transatlantic call? One possibility is to configure only one server to make that call, but instead make the
protocol such that even if a replica is down, someone will make the transatlantic call before the replica groups get too far out of sync.

3. Suppose the prefixes (written in binary) that have been advertised by the routing protocol are *(the zero length prefix), 0111, 1, 110, 110110. Show the TRIE data structure that would result. Look up the following addresses and indicate which prefix they match:
   0101010101, 111000111, 11010011.
   Add the prefix 0 to your data structure.
   Now, to optimize lookup time (at the expense of more memory), change the lookup so that prefixes are looked up 2 bits at a time. What would the TRIE structure look like with the prefixes (0, 0111, 1, 110, 110110) if organized so that 2 bits at a time can be scanned?

   Do the same for binary search (show the data structure that would result from the prefixes (0, 0111, 1, 110, 110110).

   What does the worst case number of comparisons depend on for TRIE? How about binary search?

4. 15-2
5. 15-7
6. (*) (that means it’s hard) 15-8. Basically it’s asking why you need to specify “don’t send me traffic for (S,M)” rather than just specifying “don’t send me traffic for M”. And assume you’re not using IGMPv3, so joins are for M and don’t specify sources.
7. 15-6. Things to think about...reason based on scaling issues such as number of groups, number of receivers per group, distance between members of a group, traffic per group, anything else you think of...
8. What would the spanning tree algorithm do in the following topology? (put X’s on links that would be put into backup state). Show the Hello message (root ID, distance to Root, myID) that would be transmitted on each LAN. Show the sequence of LANs that a packet would travel when S sends a packet to an unknown or multicast destination address. If it branches off into several directions, number all of them the next number. For instance, the LAN on which S appears would be numbered “1” because that’s where the 1st packet transmission happens. Then suppose there are 3 bridges on that LAN that each forward to a different LAN. In that case, number those 3 LANs “2” since they all (pretty much) simultaneously receive a copy of
the packet after the one on the LAN that says “1”. If on the next hop 4 more LANs receive the packet, number them all “3”, and so forth:

9. Summarize Jeff Schiller’s guest lecture.