

Assignment 3 Report

Question 1 - What are the final statistics for the ping command on the loopless topology?

--- 10.0.0.4 ping statistics ---

10 packets transmitted, 10 received, 0% packet loss, time 9015ms

rtt min/avg/max/mdev = 6.988/9.784/20.283/4.594 ms

Question 2 - What are the same statistics on the topology with loops?

--- 10.0.0.4 ping statistics ---

10 packets transmitted, 5 received, +23 duplicates, 50% packet loss, time 9018ms

rtt min/avg/max/mdev = 1249.801/7503.005/14639.488/3684.966 ms, pipe 7

Question 3 - In 1-2 sentences, what causes any differences in the ping results on the loopless vs loop-y topology?

Because each node floods the network with any packet it receives with the dumb controller, the self-loops cause duplicate ACKs to reach the source host. The self-loops also cause some packets to get stuck in an infinite loop until their TTL expires and they never reach the source.

Question 4 - With the way of computing forwarding tables that you implemented, is it possible for a malicious host to intercept packets from someone else? If it is possible, without changing the spanning tree calculation, what could you do to prevent that? If it is not possible, what prevents it?

Yes, it is possible. If a node were to connect to the network somehow, it would start receiving the "flooded" messages from its neighbor. It would be able to read the flooded messages. To prevent this, we can place restrictions on the machines that can join our network.

Question 5 - Describe how your algorithm works. If it has a name, also give the name.

I use Prim's algorithm, but without using any edge weight comparisons. It works by creating a list of all the nodes we have not inspected yet in the graph, which should be all of them since we just started. Now pick a node arbitrarily, and add all its neighbors to a list. For each neighbor, check if we have already visited that node, if yes then ignore it. Otherwise, add the neighbor to the visited nodes list, and add all the neighbor's neighbors that haven't been visited to the list of neighbors we are inspecting. Add the edge between the original node and the neighbor to a list of edges that will define the spanning tree. Repeat until we have visited all nodes.

Question 6 - Show the complexity of your algorithm in Big-O notation. Considered variables n as the # of nodes, m as the # of links, s as the # of switches and d as the maximum number of interfaces in a switch.

Prim's algorithm is known to be $O(n^2)$. You visit all nodes in the graph, which is represented by n , and for each of those nodes you inspect each of their neighbors which accumulates to another n . We multiply the two n 's together and get n^2 .

Question 7 - Give one advantage of a centralized algorithm (like yours) over a distributed algorithm (like STP). Give one disadvantage of centralized over distributed.

A centralized algorithm would have knowledge of the entire topology, which has the advantage of information over a distributed system. This knowledge can be used to plan specific routes that we want to put in place as policy. However, a distributed algorithm takes less overhead space and computation.