Max Li

COEN 241 HW 3

**Task 1**

Confirmed that h1 can ping h8

Text

Description automatically generated with low confidence

Q1: Output of “nodes” and “net”

Text

Description automatically generated

Q2: Output of “h7 ifconfig”

Text

Description automatically generated

**Task 2**

Function Call graph of this controller while in act\_like\_hub mode (sends to all switches)

Text, letter

Description automatically generated

Q2:

A screen shot of a computer

Description automatically generated with medium confidence

Takes 2.328 seconds on average to ping h1 to h2, while the minimum is 1.522 and maximum is 7.671 seconds.

Text

Description automatically generated with low confidence

For h1 ping to h8, it takes 10.040 seconds on average, with minimum of 7.856 seconds and maximum of 17.999 seconds.

It takes considerably longer for h1 to ping to h8 because for h1 to ping h2, only one switch (s3) is involved so response will be fast, but for h1 to ping h8, 5 switches are involved (s3, s2, s1, s5, s7).

Q3: Text

Description automatically generated

iperf is used to test the bandwidth between components, returning the speeds for client and server. The bandwidth for h1 and h2 to server is 6.82 Mbits/s, while for client is 8.09 Mbits/s. For h1 to h8, the bandwidth to server is 2.84 Mbits/s, while the bandwidth to client is 3.48 Mbits/s. Again, the difference is because fewer switches are involved for h1/h2, meaning higher bandwidth is possible per second without the impediment of waiting for more switches.

Q4: Which of the switches observe traffic?

Because the controller is operating as a hub, all the switches will observe traffic as packets are sent to every switch. One way of observing traffic would be to add a counter to each switch (tutorial object) so that when it handles a packet in, it increments the counter by one. Then at the end, iterate through objects and print out all the counts.

**Task 3**

Q1: Describe how the above code works, such as MAC to port.

mac\_to\_port is an internal dictionary variable for each switch that maps an ethernet address (MAC) to a port value. When h1 pings h2, if the switch s3 does not have saved the MAC->port mapping for h1, it adds this to its mac\_to\_port dictionary. Then, if h2 has a known port, the packet is forwarded only to h2 and not to all other components. However, if h2 doesn’t have a port known by this switch yet, then we again resend this packet to all other components.

Q2:

Text

Description automatically generated

For h1 to ping h2, the average was 4.244 seconds, the minimum was 1.677 seconds, and maximum 39.509 seconds. For h1 to ping h8 (below), the average was 16.115 seconds, the minimum was 9.308 seconds, and the maximum was 54.407 seconds.

Text

Description automatically generated with medium confidence

Both of these are considerably slower than the results in task 2 (when controller was operating as a hub). Not exactly sure why this would be the case because the only extra overhead is checking whether the MAC is known in the internal dictionary, which should not be very much overhead. I would imagine that a controller acting as a hub would be slower since extra traffic would be sent to each switch and this would cause crowding issues, perhaps. Whereas when the controller acts as a switch, you would expect clearly defined and precise transmission.

Q3:

Text

Description automatically generated

The bandwidth between h1 and h2 is considerably faster than before, though the bandwidth between h1 and h8 is a bit slower. The bandwidth for h1 and h2 might be better because using controller as a switch here is a more precise way to limit the actions of each switch to only do what is necessary, causing a speedup. Not sure why h1 – h8 would have a lower bandwidth here but it could be random fluctuation and a less noticeable difference since this connection involves more switches.