Rajat Sethi

ECE 4380 – Computer Communications

Machine Problem 3

7/20/2021

**Summary:**

The goal of this lab was to create a network on GENI using OSPF routing. Once created, I was to explore the properties of OSPF routing through several commands, including but not limited to:

* vtysh
* show ip ospf neighbor
* show ip ospf route
* traceroute
* tcpdump
* configure terminal

I was also able to explore a little bit of Quagga, Zebra, and OSPFD and their configurations.

**Implementation:**

The implementation for this project was quite similar to “Machine Problem 2.” Using GENI, I created a network on the Rutgers server with six bridges and six hosts. Through scp, I moved the routeconfig.sh file into each of the bridges and installed Quagga with its dependencies.

For the first question, I used grep with a RegEx to split the output into separate lines. Using grep and echo, I organized and split the lines even further until I had arrays for the ethernet interfaces, IP numbers, and IP subnets.

For the second question, I did the same thing as “Machine Problem 2.” I logged into each bridge and used ifconfig to determine where each ethernet interface went.

For the third question, I took the same link up and down while testing the network using “neighbor” and “ping.” I examined the dead time for the down link, and used ping to see how the routing changed almost instantly when the link fell.

For the fourth question, I used “configure terminal”, “interface”, and “ip ospf cost” commands to change the cost of a link. I then used traceroute and tcpdump to see what changes would happen to the network (especially the hosts that were using that route for the shortest path).

**Questions:**

Q1.)

#!/bin/bash

ETHLINES=$(ip addr show | grep -E "eth[1-9][0-9]\*$")

ETHNUMS=($(echo "${ETHLINES[\*]}" | grep -Eo "eth[1-9][0-9]\*"))

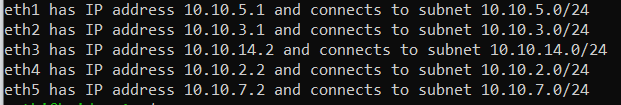
IPNUMS=($(echo "${ETHLINES[\*]}" | grep -Eo "inet [0-9][0-9]\*\.[0-9][0-9]\*\.[0-9][0-9]\*\.[0-9][0-9]\*" | cut -c6-))

IPSUBS=($(echo "${IPNUMS[\*]}" | grep -Eo "[0-9][0-9]\*\.[0-9][0-9]\*\.[0-9][0-9]\*"))

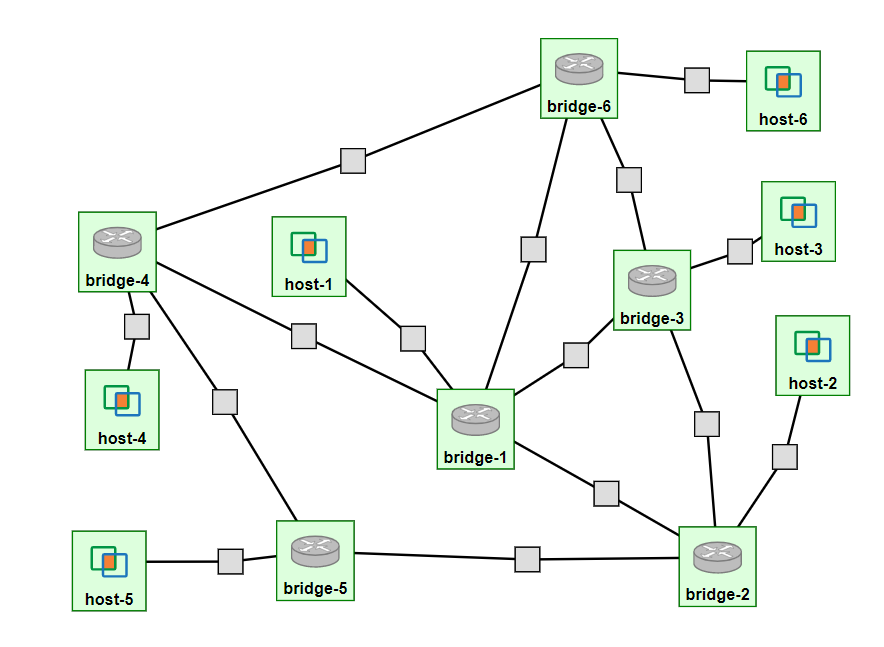
for i in ${!ETHNUMS[@]} ; do

echo "${ETHNUMS[$i]} has IP address ${IPNUMS[$i]} and connects to subnet ${IPSUBS[$i]}.0/24"

done



Q2.)



host6: @pc3:25818

eth1: 10.1

host3: @pc1:25811

eth1: 9.1

bridge6: @pc3:25814

eth1: 8.2

eth2: 1.1

eth3: 2.1

eth4: 10.2

bridge4: @pc3:25812

eth1: 4.1

eth2: 3.2

eth3: 11.2

eth4: 1.2

host1: @pc3:25815

eth1: 14.1

bridge3: @pc1:25810

eth1: 15.1

eth2: 9.2

eth3: 8.1

eth4: 7.1

bridge2: @pc3:25811

eth1: 5.2

eth2: 15.2

eth3: 13.2

eth4: 6.2

bridge1: @pc3:25810

eth1: 5.1

eth2: 3.1

eth3: 14.2

eth4: 2.2

eth5: 7.2

bridge5: @pc3:25813

eth1: 4.2

eth2: 12.2

eth3: 6.1

host2: @pc3:25816

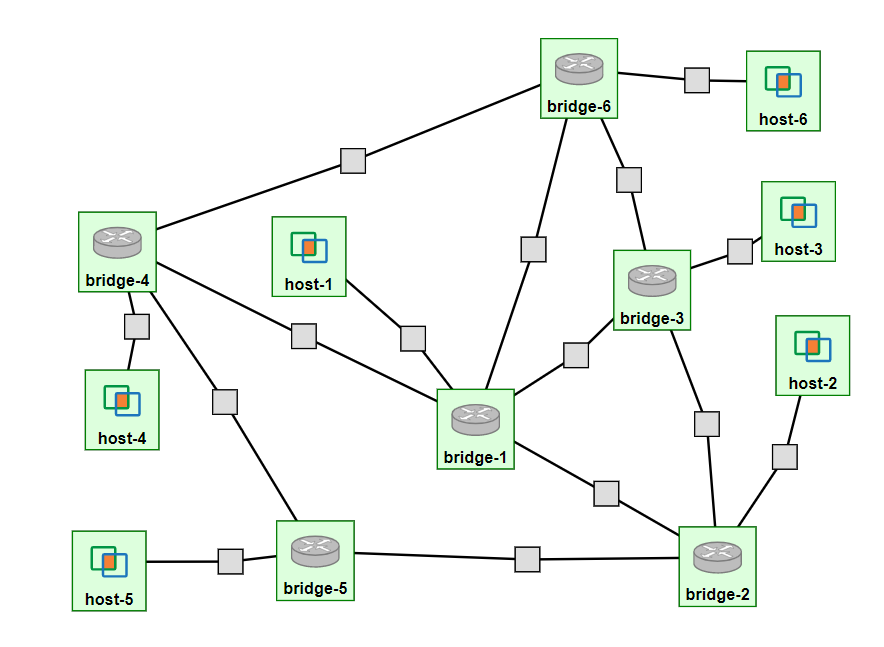
eth1: 13.1

host4: @pc3:25817

eth1: 11.1

host5: @pc1:25812

eth1: 12.1

Q3.)

eth2

eth2

eth1

eth3

eth1

eth4

eth5

eth1

eth1

eth3

eth4

eth3

eth4

eth3

eth1

eth1

eth3

eth1

eth2

eth2

eth1

eth1

eth2

eth1

eth3

eth1

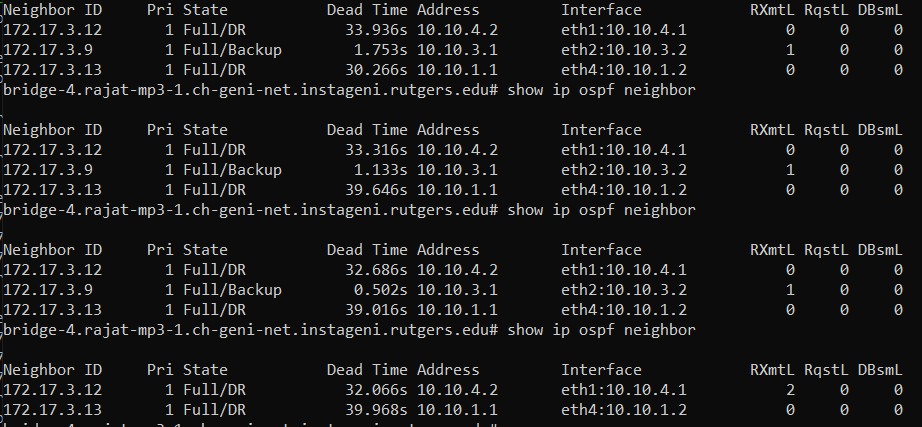
eth2

eth4

eth4

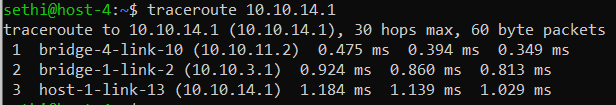
eth1

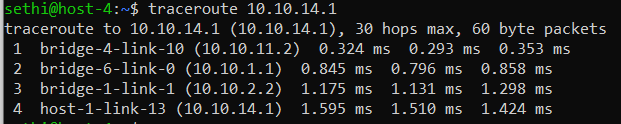
Taking down the link between bridge-1 and bridge-4 (Both are eth2)



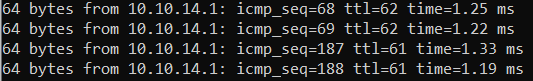
Sending pings from host-4 to host-1

(Traceroute while the link is up)



(Traceroute while the link is down)  


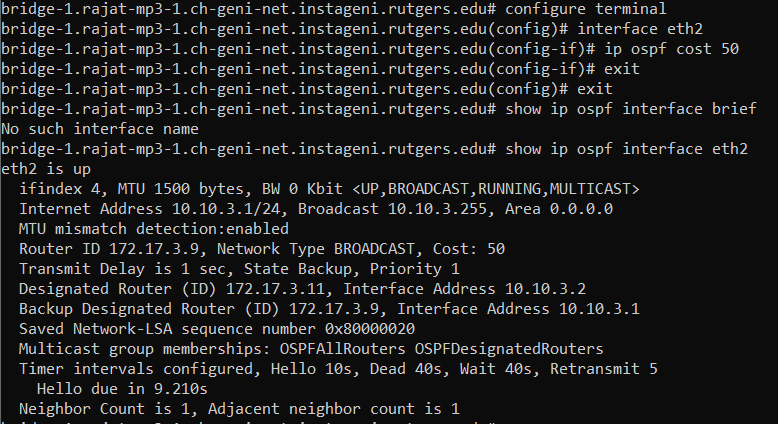
(Change in pings run at 0.05 seconds)



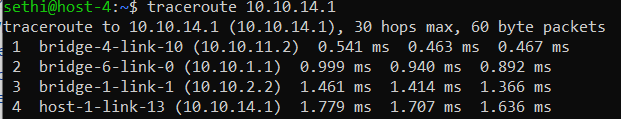
Running the experiment several times, I found that the route would be recalculated in a negligible amount of time (<0.1 seconds). However, if a ping got “stuck” by chance (as no acknowledgement was received), then it would take around 5-10 seconds to fully recalculate.

Q4.)

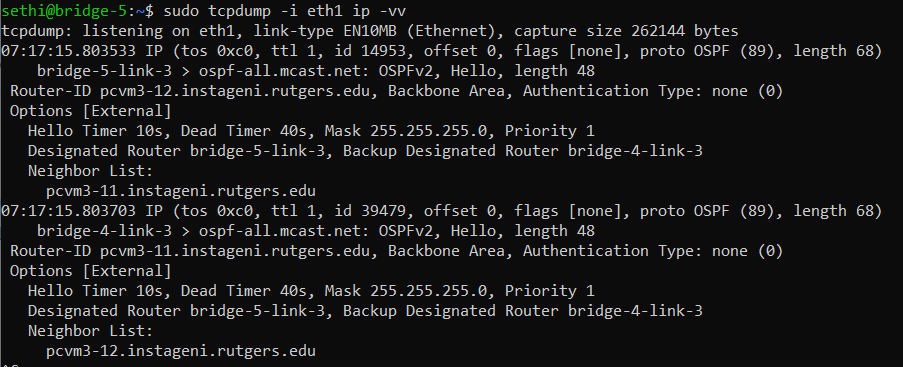
Change in cost to link between bridge-1 and bridge-4 (Applied on both interfaces).



Effects to path when the cost is changed.



tcpdump of bridge-5 (separate from cost change), still receiving LSP packets.



**Conclusion:**

Overall, I learned about the OSPF protocol and the respective commands to configure the network. Surprisingly, I found the hardest part of this lab to be the bash script. I used bash very sparsely in my prior classes, so working with arrays was a little tougher than I expected. The other difficulty I had was with the pinging when taking down a link. At first, I did not think I was doing the question correctly, as the “ttl” and “icmp\_seq” parameters were not changing. I kept changing commands and hosts until I finally got everything to work as expected, which was a good test of patience.