

# System Setup

---

## GENI Project

1. Registered for GENI project and upload SSH key
2. Created a project with a slice
3. Add Resource: uploaded the Rspec XML to it
4. Reserved resource for the experiment

## Modify Script

1. Changed all usernames for ssh connection to *hc50*
2. Changed all ports to the resources assigned to my slice
3. Configured the ip addresses for all nodes.
4. Added aliases for each routers in */etc/hosts*

## Result

### Script Execution

```
bash l3_setup.sh
router1.hc50.ch-geni-net.instageni.illinois.edu
127.0.0.1      localhost loghost localhost.hc50.ch-geni-net.ins
router2.hc50.ch-geni-net.geni.it.cornell.edu
127.0.0.1      localhost loghost localhost.hc50.ch-geni-net.gen
router3.hc50.ch-geni-net.instageni.idre.ucla.edu
127.0.0.1      localhost loghost localhost.hc50.ch-geni-net.ins
node1-1.hc50.ch-geni-net.instageni.illinois.edu
127.0.0.1      localhost loghost localhost.hc50.ch-geni-net.ins
node1-2.hc50.ch-geni-net.instageni.illinois.edu
127.0.0.1      localhost loghost localhost.hc50.ch-geni-net.ins
node2-1.hc50.ch-geni-net.geni.it.cornell.edu
127.0.0.1      localhost loghost localhost.hc50.ch-geni-net.gen
node2-2.hc50.ch-geni-net.geni.it.cornell.edu
127.0.0.1      localhost loghost localhost.hc50.ch-geni-net.gen
node3-2.hc50.ch-geni-net.instageni.idre.ucla.edu
127.0.0.1      localhost loghost localhost.hc50.ch-geni-net.ins
node3-1.hc50.ch-geni-net.instageni.idre.ucla.edu
127.0.0.1      localhost loghost localhost.hc50.ch-geni-net.ins
```

```
node1-1.hc50.ch-genet.net.instageni.illinois.edu
node1-2.hc50.ch-genet.net.instageni.illinois.edu
node2-1.hc50.ch-genet.net.geni.it.cornell.edu
node2-2.hc50.ch-genet.net.geni.it.cornell.edu
node3-2.hc50.ch-genet.net.instageni.idre.ucla.edu
node3-1.hc50.ch-genet.net.instageni.idre.ucla.edu
router1.hc50.ch-genet.net.instageni.illinois.edu
router2.hc50.ch-genet.net.geni.it.cornell.edu
router3.hc50.ch-genet.net.instageni.idre.ucla.edu
```

## Ping in router1

Can ping to *router2* and *router3* successfully. This is because there is a connection (in the same subnet) between *router1* and *router2*, as well as between *router1* and \**router3*\*.

```
hc50@router1:~$ ping router2
PING router2 (10.10.101.2) 56(84) bytes of data.
64 bytes from router2 (10.10.101.2): icmp_seq=1 ttl=64 time=47.1
64 bytes from router2 (10.10.101.2): icmp_seq=2 ttl=64 time=21.9
64 bytes from router2 (10.10.101.2): icmp_seq=3 ttl=64 time=22.1
^C
--- router2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 21.960/30.422/47.178/11.849 ms
hc50@router1:~$ ping router3
PING router3 (10.10.100.3) 56(84) bytes of data.
64 bytes from router3 (10.10.100.3): icmp_seq=1 ttl=64 time=98.1
64 bytes from router3 (10.10.100.3): icmp_seq=2 ttl=64 time=47.6
64 bytes from router3 (10.10.100.3): icmp_seq=3 ttl=64 time=47.6
^C
--- router3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 47.619/64.473/98.119/23.792 ms
hc50@router1:~$
```

See the route:

```
hc50@router1:~$ ip route show
default via 172.16.0.1 dev eth0 proto dhcp src 172.17.2.9 metric
10.10.0.0/24 dev eth1 proto kernel scope link src 10.10.0.1
10.10.100.0/24 dev eth3 proto kernel scope link src 10.10.100.1
10.10.101.0/24 dev eth4 proto kernel scope link src 10.10.101.1
```

```
172.16.0.0/12 dev eth0 proto kernel scope link src 172.17.2.9
172.16.0.1 dev eth0 proto dhcp scope link src 172.17.2.9 metric
```

## Ping in node1-1

Cannot ping to *router2* or *node2-1* because there is no connection directly reachable to these two hosts.

```
hc50@node1-1:~$ ping 10.10.1.21
PING 10.10.1.21 (10.10.1.21) 56(84) bytes of data.
^C
--- 10.10.1.21 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4098ms

hc50@node1-1:~$ ping 10.10.1.1
PING 10.10.1.1 (10.10.1.1) 56(84) bytes of data.
^C
--- 10.10.1.1 ping statistics ---
11 packets transmitted, 0 received, 100% packet loss, time 10237
```

Check the route:

```
hc50@node1-1:~$ ip route show
default via 172.16.0.1 dev eth0 proto dhcp src 172.17.1.5 metric
10.10.0.0/24 dev eth1 proto kernel scope link src 10.10.0.11
172.16.0.0/12 dev eth0 proto kernel scope link src 172.17.1.5
172.16.0.1 dev eth0 proto dhcp scope link src 172.17.1.5 metric
```

## Task1

---

In this task, static routing table has been set up manually and separately for routers and hosts.

The result of *ping*, which proves the accessibility between nodes in **site 1** and nodes in **site 2**, after setting up route table from *node1-1* is:

```
hc50@node1-1:~$ ping 10.10.1.22
PING 10.10.1.22 (10.10.1.22) 56(84) bytes of data.
64 bytes from 10.10.1.22: icmp_seq=1 ttl=62 time=24.2 ms
64 bytes from 10.10.1.22: icmp_seq=2 ttl=62 time=23.3 ms
```

```

64 bytes from 10.10.1.22: icmp_seq=3 ttl=62 time=23.3 ms
64 bytes from 10.10.1.22: icmp_seq=4 ttl=62 time=23.3 ms
64 bytes from 10.10.1.22: icmp_seq=5 ttl=62 time=23.8 ms
^C
--- 10.10.1.22 ping statistics ---
6 packets transmitted, 5 received, 16% packet loss, time 5007ms
rtt min/avg/max/mdev = 23.301/23.613/24.242/0.368 ms
hc50@node1-1:~$ ping 10.10.1.1
PING 10.10.1.1 (10.10.1.1) 56(84) bytes of data.
64 bytes from 10.10.1.1: icmp_seq=1 ttl=63 time=23.0 ms
64 bytes from 10.10.1.1: icmp_seq=2 ttl=63 time=23.3 ms
64 bytes from 10.10.1.1: icmp_seq=3 ttl=63 time=22.6 ms
64 bytes from 10.10.1.1: icmp_seq=4 ttl=63 time=23.4 ms
^C
--- 10.10.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 22.696/23.126/23.409/0.297 ms

```

## Routing Table

For the detail of routing table on other machines, please check the script.

- *router1*

```

hc50@router1:~$ ip r
default via 172.16.0.1 dev eth0 proto dhcp src 172.17.2.9 metric
10.10.0.0/24 dev eth1 proto kernel scope link src 10.10.0.1
10.10.1.0/24 via 10.10.101.2 dev eth4
10.10.100.0/24 dev eth3 proto kernel scope link src 10.10.100.1
10.10.101.0/24 dev eth4 proto kernel scope link src 10.10.101.1
172.16.0.0/12 dev eth0 proto kernel scope link src 172.17.2.9
172.16.0.1 dev eth0 proto dhcp scope link src 172.17.2.9 metric
192.168.0.0/24 via 10.10.100.3 dev eth3

```

- *node1-1*

```

hc50@node1-1:~$ ip r
default via 172.16.0.1 dev eth0 proto dhcp src 172.17.1.5 metric
10.10.0.0/24 dev eth1 proto kernel scope link src 10.10.0.11
10.10.0.0/16 via 10.10.0.1 dev eth1
172.16.0.0/12 dev eth0 proto kernel scope link src 172.17.1.5
172.16.0.1 dev eth0 proto dhcp scope link src 172.17.1.5 metric

```

# Task2

---

## Direct Routing

**site 1 <-----> site 2**

```
hc50@node1-1:~$ traceroute 10.10.1.1
traceroute to 10.10.1.1 (10.10.1.1), 30 hops max, 60 byte packet
 1  10.10.0.1 (10.10.0.1)  2.878 ms  2.755 ms  2.666 ms
 2  * 10.10.1.1 (10.10.1.1)  50.061 ms *
hc50@node1-1:~$ ping 10.10.1.1
PING 10.10.1.1 (10.10.1.1) 56(84) bytes of data.
64 bytes from 10.10.1.1: icmp_seq=1 ttl=63 time=23.3 ms
64 bytes from 10.10.1.1: icmp_seq=2 ttl=63 time=23.2 ms
64 bytes from 10.10.1.1: icmp_seq=3 ttl=63 time=23.2 ms
64 bytes from 10.10.1.1: icmp_seq=4 ttl=63 time=23.0 ms
64 bytes from 10.10.1.1: icmp_seq=5 ttl=63 time=23.0 ms
^C
--- 10.10.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 23.062/23.182/23.310/0.216 ms
```

## Indirect Routing

**site 1 <----> site 3 <----> site 2**

Please run [detour.sh](#) to make achieve the goal of task2. Obviously, the delay increased.

```
hc50@node1-1:~$ ping 10.10.1.1
PING 10.10.1.1 (10.10.1.1) 56(84) bytes of data.
64 bytes from 10.10.1.1: icmp_seq=1 ttl=62 time=228 ms
64 bytes from 10.10.1.1: icmp_seq=2 ttl=62 time=111 ms
64 bytes from 10.10.1.1: icmp_seq=3 ttl=62 time=111 ms
64 bytes from 10.10.1.1: icmp_seq=4 ttl=62 time=111 ms
64 bytes from 10.10.1.1: icmp_seq=5 ttl=62 time=111 ms
64 bytes from 10.10.1.1: icmp_seq=6 ttl=62 time=112 ms
^C
--- 10.10.1.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 111.711/131.289/228.316/43.392 ms
hc50@node1-1:~$ traceroute 10.10.1.1
```

```
traceroute to 10.10.1.1 (10.10.1.1), 30 hops max, 60 byte packet
 1  10.10.0.1 (10.10.0.1)  1.945 ms  1.816 ms  1.670 ms
 2  10.10.100.3 (10.10.100.3)  50.428 ms  50.312 ms  50.177 ms
 3  * 10.10.1.1 (10.10.1.1)  115.395 ms  115.204 ms
```

If you want to go back to normal routing(without "detour"), please run [cleanRouting.sh](#) and [l3\\_setup.sh](#).

## Task3

---

### Preparation

1. First, we need to config the routing table of hosts in **site 3**. Let traffic going to 10.10.0.0/16 be routed to 192.168.0.1
2. Then, copy the program to the nodes:

```
# copy from my machine to node1-1
scp -P 26810 -r ./src/py/ hc50@pc1.instageni.illinois.edu:~/ns/
# copy from my machine to node3-1
scp -P 28610 -r ./src/py/ hc50@pc1.instageni.idre.ucla.edu:~/ns
```

3. After that, config the iptables rule on router3:

```
sudo iptables -t nat -A POSTROUTING -s 10.10.0.0/24 -o eth3 -j M
sudo iptables -t nat -A POSTROUTING -s 10.10.1.0/24 -o eth4 -j M
sudo iptables -t nat -A PREROUTING -p tcp --dport 12345 -j DNAT
```

4. Last, run the program on client and server.

### Result

Using TCP protocol and port 12345, the server and client can communicate.

#### Client Side *node1-1*

```
hc50@node1-1:~/ns$ ./netster.py 10.10.100.3
INFO:client: Hello, I am a client...
INFO:client: Connect to server 10.10.100.3 success!
```

```
>hello
INFO:client: Server says: world

>aisudgasuigdasui
INFO:client: Server says: aisudgasuigdasui

>exit
INFO:client: Server says: ok
```

### Server Side *node3-1*

```
hc50@node3-1:~/ns$ ./netster.py
INFO:server: Hello, I am a server...
INFO:server: Service is on! hostname: 192.168.0.32, port: 12345
INFO:server: Service is using UDP: False
INFO:server: New Thread for: ('10.10.0.11', 52050)
INFO:server: New message from ('10.10.0.11', 52050): hello

INFO:server: New message from ('10.10.0.11', 52050): aisudgasuig

INFO:server: New message from ('10.10.0.11', 52050): exit

INFO:server: Got an exit signal, terminating myself...
```

### Router Side *router3*

```
hc50@router3:~$ sudo iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source                destination
DNAT       tcp  --  anywhere              anywhere            tc

Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source                destination
MASQUERADE all  --  10.10.0.0/24          anywhere
```

```
MASQUERADE all -- 10.10.1.0/24 anywhere
hc50@router3:~$ sudo tcpdump -i eth1 -n
tcpdump: verbose output suppressed, use -v or -vv for full proto
listening on eth1, link-type EN10MB (Ethernet), capture size 262
19:27:33.462597 IP 10.10.0.11.52050 > 192.168.0.32.12345: Flags
19:27:33.463301 IP 192.168.0.32.12345 > 10.10.0.11.52050: Flags
19:27:33.512907 IP 10.10.0.11.52050 > 192.168.0.32.12345: Flags
19:27:35.424832 IP 10.10.0.11.52050 > 192.168.0.32.12345: Flags
19:27:35.425747 IP 192.168.0.32.12345 > 10.10.0.11.52050: Flags
19:27:35.427367 IP 192.168.0.32.12345 > 10.10.0.11.52050: Flags
19:27:35.475781 IP 10.10.0.11.52050 > 192.168.0.32.12345: Flags
19:27:36.945276 IP 10.10.0.11.52050 > 192.168.0.32.12345: Flags
19:27:36.947840 IP 192.168.0.32.12345 > 10.10.0.11.52050: Flags
19:27:36.995992 IP 10.10.0.11.52050 > 192.168.0.32.12345: Flags
19:27:38.601929 ARP, Request who-has 192.168.0.32 tell 192.168.0
19:27:38.602567 ARP, Reply 192.168.0.32 is-at 02:46:b1:ce:35:80,
19:27:39.006392 IP 10.10.0.11.52050 > 192.168.0.32.12345: Flags
19:27:39.008819 IP 192.168.0.32.12345 > 10.10.0.11.52050: Flags
19:27:39.009319 IP 192.168.0.32.12345 > 10.10.0.11.52050: Flags
19:27:39.057102 IP 10.10.0.11.52050 > 192.168.0.32.12345: Flags
19:27:39.057445 IP 10.10.0.11.52050 > 192.168.0.32.12345: Flags
19:27:39.058153 IP 192.168.0.32.12345 > 10.10.0.11.52050: Flags
```