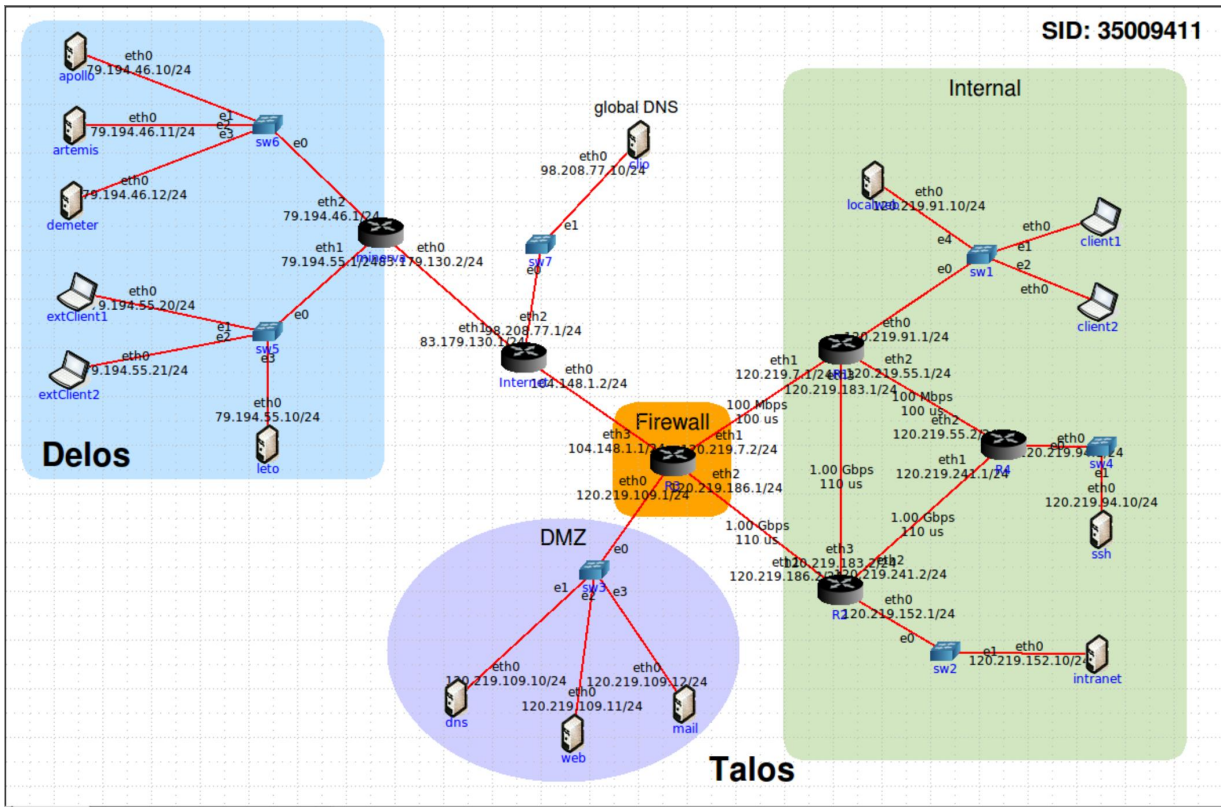


Content

Task A

- Part one: Configuration
  - R1
  - R2
  - R3
  - R4
  - Minerva
  - Internet
- Part two: Testing

Task B

- Part one: Configuration

        - Minerva
        - Clients of delos
    - Part two: Testing

Task C

- Default
- Anywhere to DMZ
- DMZ to external
- Internal to DMZ
- Internal to internal
- Internal to external
- Clients to R3
- R3 ICMP

Task A: Routing

Part one: configuration

A.1.1 R1:

a. Configuration

1. Default configuration

Use *ip route* command to check the existing routing table

```
root@R1:/tmp/pycore.53692/R1.conf# ip route
120.219.7.0/24 dev eth1  proto kernel  scope link  src 120.219.7.1
120.219.55.0/24 dev eth2  proto kernel  scope link  src 120.219.55.1
120.219.91.0/24 dev eth0  proto kernel  scope link  src 120.219.91.1
120.219.183.0/24 dev eth3  proto kernel  scope link  src 120.219.183.1
```

Subnet 120.219.7.0/24 is directly connected via interface eth1

Subnet 120.219.55.0/24 is directly connected via interface eth2

Subnet 120.219.91.0/24 is directly connected via interface eth0

Subnet 120.219.183.0/24 is directly connected via interface eth3

2. Changes I made

i) Static routing

Add static routing to subnet 120.219.94.0/24 via R4's interface eth2:

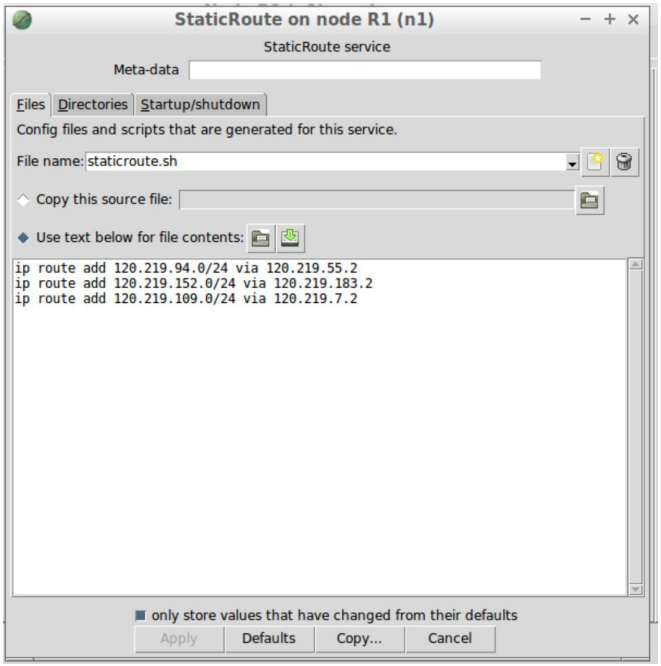
ip route add 120.219.94.0/24 via 120.219.55.2

Add static routing to subnet 120.219.152.0/24 via R2's interface eth3:

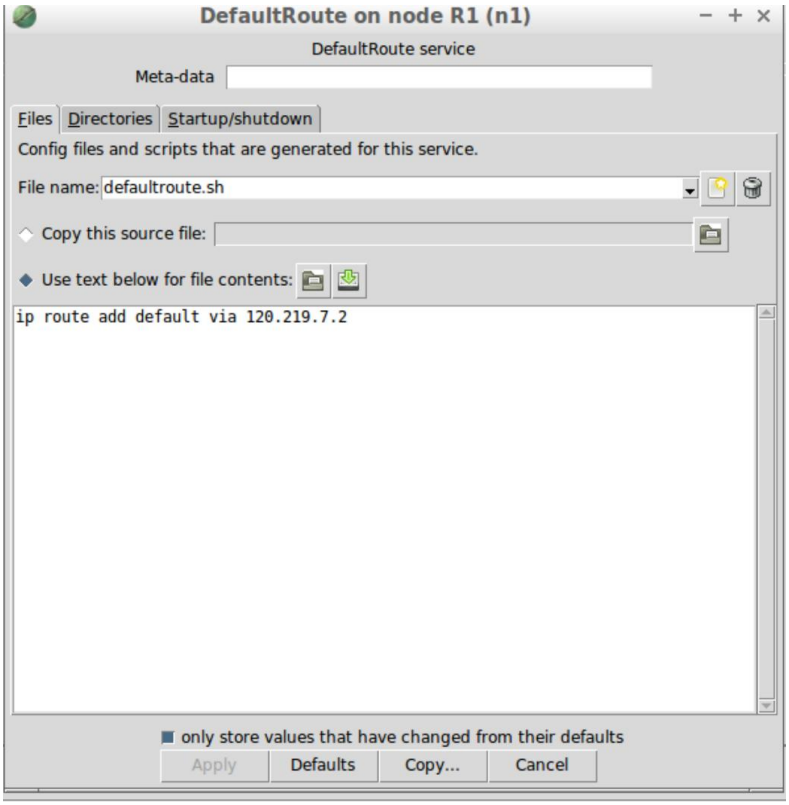
ip route add 120.219.152.0/24 via 120.219.183.2

Add static routing to subnet 120.219.109.0/24 via R3's interface eth1:

ip route add 120.219.109.0/24 via 120.219.7.2



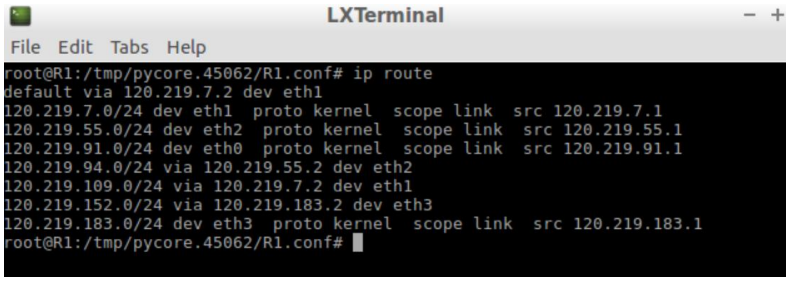
ii)Default gateway



Add default gateway via R3's interface eth1:

ip route add default via 120.219.7.2

3. Results



b. Reasons for changes

i) Static routing set up

R1 needs to have all subnets of the same organisation talos in its routing table, that is, 120.219.91.0/24, 120.219.94.0/24, 120.219.109.0/24 and 120.219.152.0/24. In order to forward message to these subnets, R1 has to choose the optimal routing path.

Factors to consider are first the bandwidth and second the link delay. Generally, link delay has a greater impact on path selection in that bandwidth mainly determines the maximum transmission speed, but the latter directly affects the time of the whole transmission without relation to speed. That means however fast the speed is, it still takes time to transmit due to the link delay. As a result, link delay plays a primary role especially in real-time senarios such as video-conferencing and web-browsing. Last but not least, the link delay can be accumulated across the path on each node, so in principle there should be as few nodes as possible in the routing route.

Although bandwidth has some impact on latency, it is only in single-digit nanoseconds while the link delay has effect of more than 100 microseconds as showed in the layout. Significantly, bandwidth plays a negligible role in our consideration compared with link delay. And we can even come to the conclusion that the performance of network in our design is proportional to the number of nodes since each node will add up at least 100 us of delay.

As for R1

The optimal path to 120.219.91.0/24 is via eth0 which is directly connected and has no other possible path.

The optimal path to 120.219.94.0/24 is via R4's interface eth2 so that the number of node passing through is 1 and the latency is appropriately 100us.

The optimal path to 120.219.109.0/24 is via R3's interface eth1, the number of node is 1 and the latency is appropriately 100us.

The optimal path to 120.219.152.0/24 is via R2's interface eth3, the number of node is 1 and the latency is appropriately 110us.

ii) Default gateway

As required by the specification, R3 must be the default gateway of R1, that is R3's interface eth1.

A.1.2 R2

a. Configuration

1. Default configuration

```
root@R2:/tmp/pycore.45064/R2.conf# ip route
120.219.152.0/24 dev eth0  proto kernel  scope link  src 120.219.152.1
120.219.183.0/24 dev eth3  proto kernel  scope link  src 120.219.183.2
120.219.186.0/24 dev eth1  proto kernel  scope link  src 120.219.186.2
120.219.241.0/24 dev eth2  proto kernel  scope link  src 120.219.241.2
```

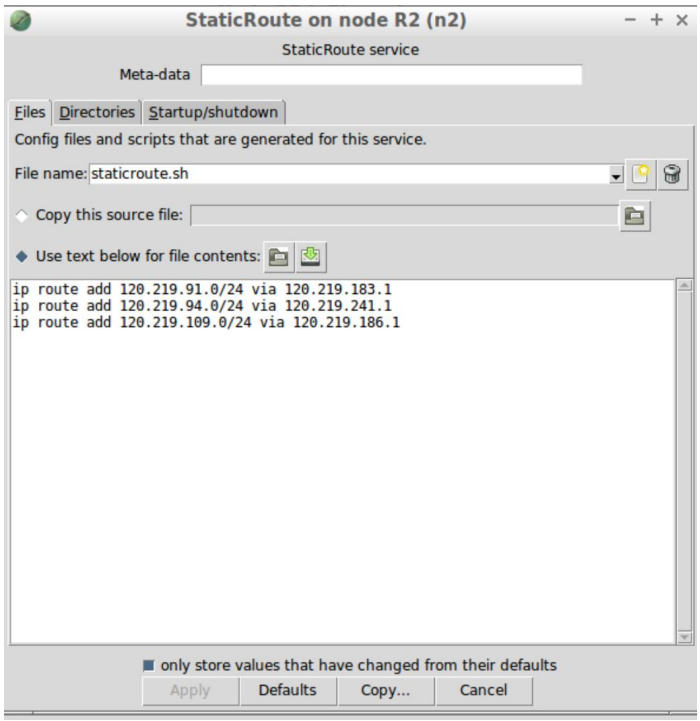
2. Changes I made

i) Static routing

ip route add 120.219.91.0/24 via 120.219.183.1

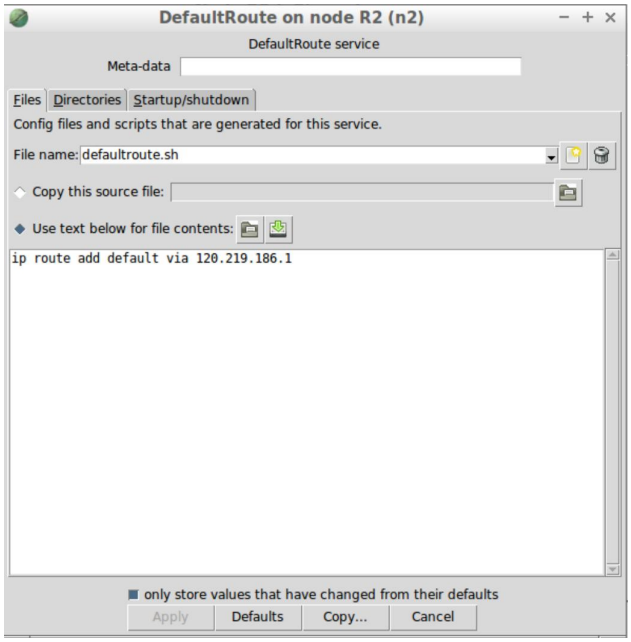
ip route add 120.219.94.0/24 via 120.219.241.1

ip route add 120.219.109.0/24 via 120.219.186.1



ii) Default gateway

ip route add default via 120.219.186.1



3. Results

```
root@R2:/tmp/pycore.45070/R2.conf# ip route
default via 120.219.186.1 dev eth1
120.219.91.0/24 via 120.219.183.1 dev eth3
120.219.94.0/24 via 120.219.241.1 dev eth2
120.219.109.0/24 via 120.219.186.1 dev eth1
120.219.152.0/24 dev eth0  proto kernel  scope link  src 120.219.152.1
120.219.183.0/24 dev eth3  proto kernel  scope link  src 120.219.183.2
120.219.186.0/24 dev eth1  proto kernel  scope link  src 120.219.186.2
120.219.241.0/24 dev eth2  proto kernel  scope link  src 120.219.241.2
```

b. Reasons for changes

i) Static routing

Similar to R1, the fewest node and the smallest link delay

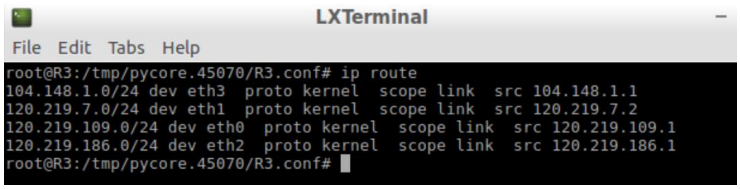
ii) Default gateway

As required by the specification, R3 should be R2's default gateway.

A.1.3 R3

a. Configuration

1. Default configuration



2. Changes I made

i) Static routing

ip route add 120.219.91.0/24 via 120.219.7.1

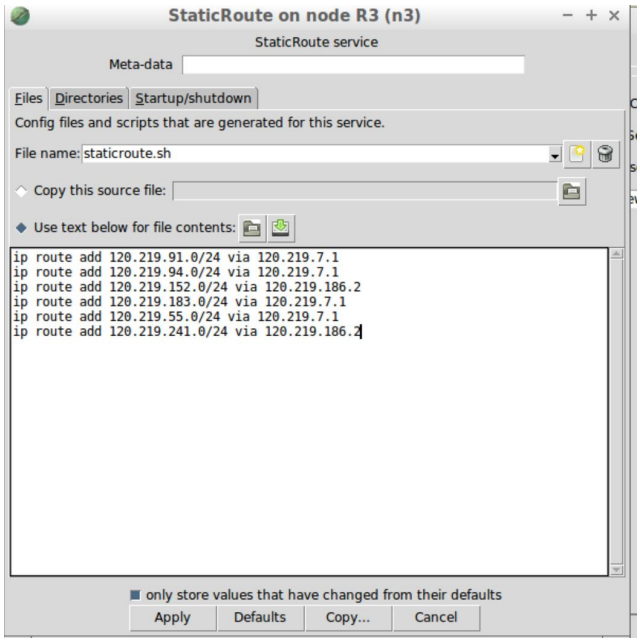
ip route add 120.219.94.0/24 via 120.219.7.1

ip route add 120.219.152.0/24 via 120.219.186.2

ip route add 120.219.183.0/24 via 120.219.7.1

ip route add 120.219.55.0/24 via 120.219.7.1

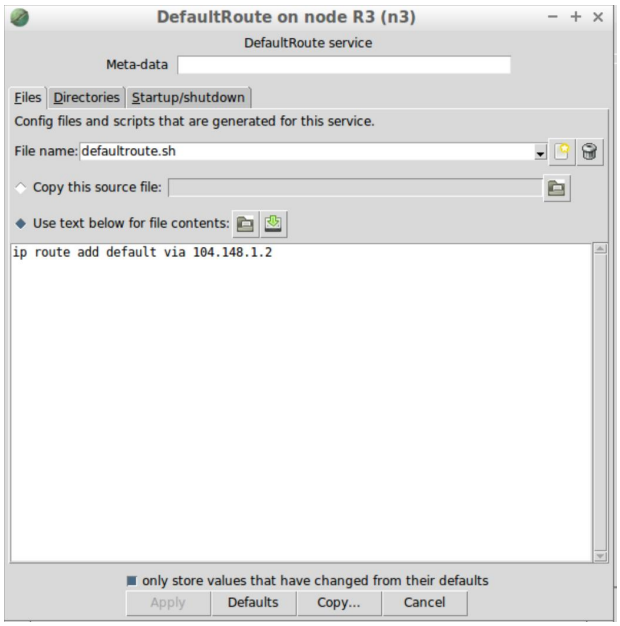
ip route add 120.219.241.0/24 via 120.219.186.2



The last 3 static routings are added to ensure R3 can reach R1's and R2's intermediate interface, as well as R4's interfaces. Without these, there will be a loop between R3 and Internet if R3 tries to ping R1's and R2's intermediate interface or ping R4.

ii) Default gateway

ip route add default via 104.148.1.2



3. Results

```
root@R3:/tmp/pycore.45072/R3.conf# ip route
default via 104.148.1.2 dev eth3
104.148.1.0/24 dev eth3  proto kernel  scope link  src 104.148.1.1
120.219.7.0/24 dev eth1  proto kernel  scope link  src 120.219.7.2
120.219.91.0/24 via 120.219.7.1 dev eth1
120.219.94.0/24 via 120.219.7.1 dev eth1
120.219.109.0/24 dev eth0  proto kernel  scope link  src 120.219.109.1
120.219.152.0/24 via 120.219.186.2 dev eth2
120.219.186.0/24 dev eth2  proto kernel  scope link  src 120.219.186.1
root@R3:/tmp/pycore.45072/R3.conf#
```

b. Reasons for changes

i) Static routing

The reasons for choosing optimal pahts for subnets 120.219.91.0/24 and 120.219.152.0/24

are similar to R1 and R2 as there is only one node passing through.

For R3 to connect to 120.219.94.0/24, it's better to go through R1 instead of R2. Although the number of nodes is same, both are 2, the total latency via R1 is about 100 + 100 = 200 us, while the route through R2 will result in a latency of 110 + 110 = 220 us. The effect of bandwidth is negligible here in terms of latency, as a 1 Gbps bandwidth would only reduce the delay by 9 nanoseconds compared to a 100 Mbps bandwidth.

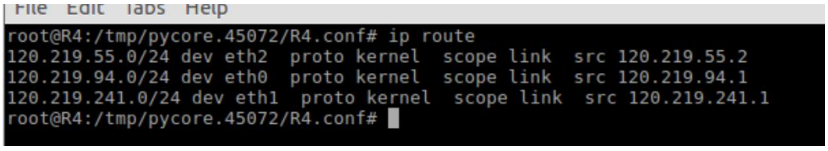
ii) Default gateway

As is required by the specification, Internet should be R3's default gateway, that is via 104.148.1.2.

A.1.4 R4

a. Configuration

1. Default configuration



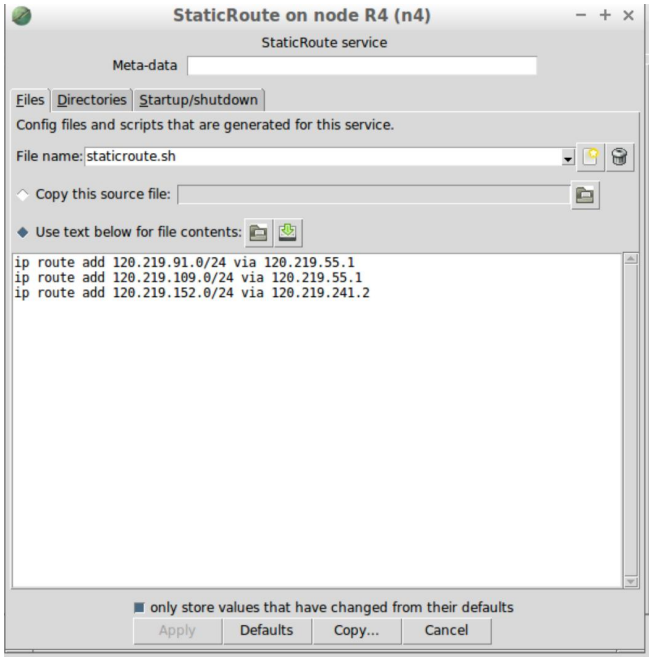
2. Changes I made

i) Static routing

ip route add 120.219.91.0/24 via 120.219.55.1

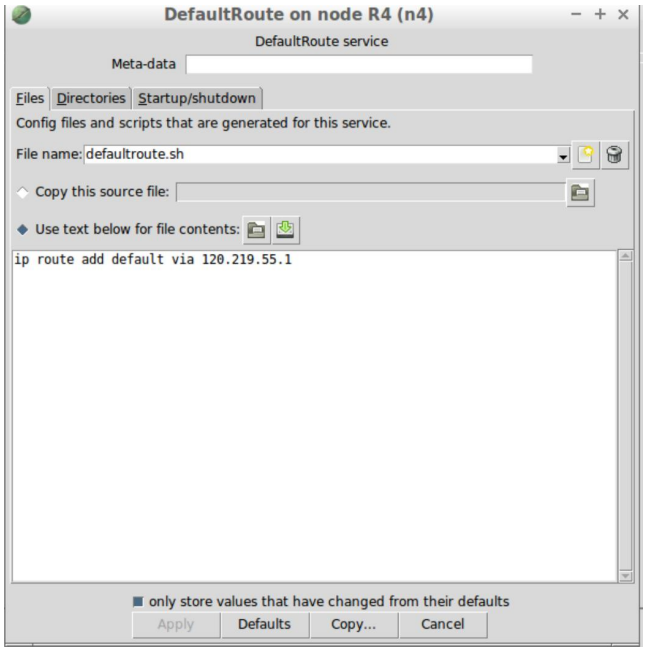
ip route add 120.219.109.0/24 via 120.219.55.1

ip route add 120.219.152.0/24 via 120.219.241.2



ii) Default gateway

ip route add default via 120.219.55.1



3. Results

```
root@R4:/tmp/pycore.45074/R4.conf# ip route
default via 120.219.55.1 dev eth2
120.219.55.0/24 dev eth2  proto kernel  scope link  src 120.219.55.2
120.219.91.0/24 via 120.219.55.1 dev eth2
120.219.94.0/24 dev eth0  proto kernel  scope link  src 120.219.94.1
120.219.109.0/24 via 120.219.55.1 dev eth2
120.219.152.0/24 via 120.219.241.2 dev eth1
120.219.241.0/24 dev eth1  proto kernel  scope link  src 120.219.241.1
root@R4:/tmp/pycore.45074/R4.conf#
```

b. Reasons for changes

i) Static routing

For R4 to connect to 120.219.91.0/24 and 120.219.152.0/24, the primary consideration is to pass through fewest nodes. In this case, each can go through only one node.

For R4 to connect to 120.219.109.0/24, it can go through R1 or R2 and then go through R3. R1 is a better choice due to a smaller latency of 200 us than R2's 220 us.

ii) Default gateway

Similar reason as the subnet 120.219.109.0/24, it's a better choice to go through R1 to reach R3 due to smaller latency.

A.1.5 Minerva

a. Configuration

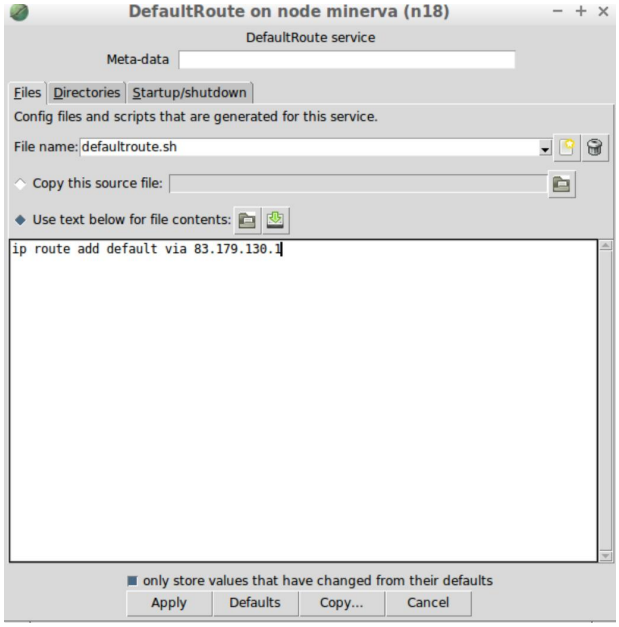
1. Default configuration

```
root@minerva:/tmp/pycore.45074/minerva.conf# ip route
79.194.46.0/24 dev eth2  proto kernel  scope link  src 79.194.46.1
79.194.55.0/24 dev eth1  proto kernel  scope link  src 79.194.55.1
83.179.130.0/24 dev eth0  proto kernel  scope link  src 83.179.130.2
root@minerva:/tmp/pycore.45074/minerva.conf#
```

2. Changes I made

i) Default gateway

ip route add default via 83.179.130.1



3. Results

```
root@minerva:/tmp/pycore.45076/minerva.conf# ip route
default via 83.179.130.1 dev eth0
79.194.46.0/24 dev eth2  proto kernel  scope link  src 79.194.46.1
79.194.55.0/24 dev eth1  proto kernel  scope link  src 79.194.55.1
83.179.130.0/24 dev eth0  proto kernel  scope link  src 83.179.130.2
root@minerva:/tmp/pycore.45076/minerva.conf#
```

b. Reasons for changes

All subnets in Minerva's network are directly connected to it, so I do not have to set the static routing but only the default gateway. As is required by the specification, Internet should be Minerva's default gateway, that is via 83.179.130.1.

A.1.6 Internet

a. Configuration

1. Default configuration

```
root@Internet:/tmp/pycore.45076/Internet.conf# ip route
83.179.130.0/24 dev eth1  proto kernel  scope link  src 83.179.130.1
98.208.77.0/24 dev eth2  proto kernel  scope link  src 98.208.77.1
104.148.1.0/24 dev eth0  proto kernel  scope link  src 104.148.1.2
```

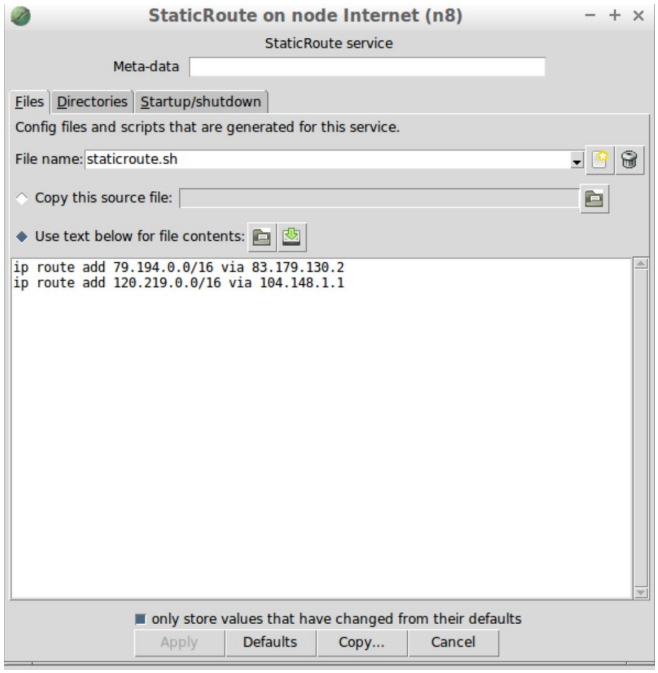
2. Changes I made

i) Static routing

Using subnet mask /16 to get the network address of the delos and talos.

ip route add 79.194.0.0/16 via 83.179.130.2

ip route add 120.219.0.0/16 via 104.148.1.1



3. Results

```
root@Internet:/tmp/pycore.45078/Internet.conf# ip route
79.194.0.0/16 via 83.179.130.2 dev eth1
83.179.130.0/24 dev eth1  proto kernel  scope link  src 83.179.130.1
98.208.77.0/24 dev eth2  proto kernel  scope link  src 98.208.77.1
104.148.1.0/24 dev eth0  proto kernel  scope link  src 104.148.1.2
120.219.0.0/16 via 104.148.1.1 dev eth0
root@Internet:/tmp/pycore.45078/Internet.conf#
```

b. Reasons for changes

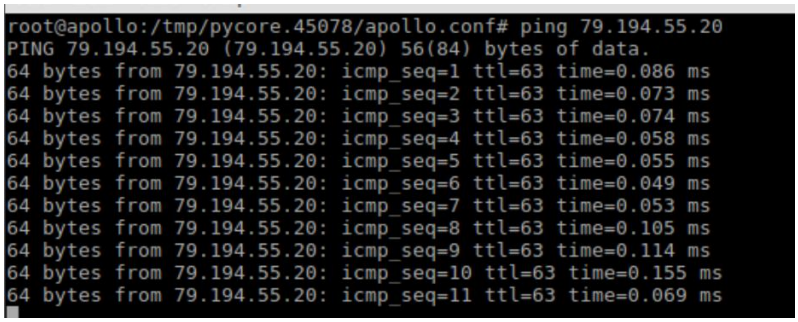
Internet will forward packets to the corresponding organisation according to the organization's ip address of the whole network, that is the common part of all its subnets. For

talos, the common part is 120.219.0.0/16, and for delos, it is 79.194.0.0/16. Internet cannot have a default gateway otherwise all traffic from the whole world may go to that ip address.

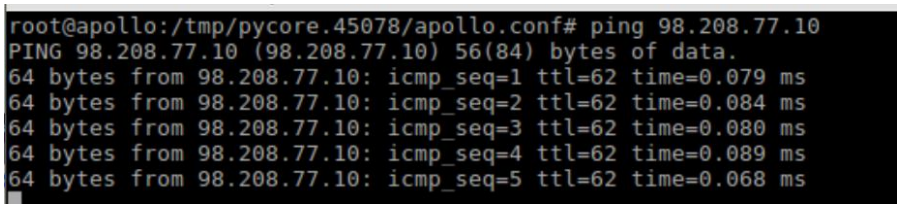
Part two: testing

I will use *ping* command to test the connection between nodes. Only one host in the same subnet will be tested. Since ICMP includes both a request and a reply, there is no need to test inversely if the signal successfuly arrived from one side to the other. For example, if 1.1.1.1 can reach 1.1.2.1, it means 1.1.2.1 can reach 1.1.1.1 too.

1. From 79.194.46.0/24 to 79.194.55.0/24(vice versa)



2. From 79.194.46.0/24 to 98.208.77.0/24(vice versa)



3. From 79.194.46.0/24 to 120.219.91.0/24(vice versa)

```
root@apollo:/tmp/pycore.45078/apollo.conf# ping 120.219.91.10
PING 120.219.91.10 (120.219.91.10) 56(84) bytes of data.
64 bytes from 120.219.91.10: icmp_seq=1 ttl=60 time=0.953 ms
64 bytes from 120.219.91.10: icmp_seq=2 ttl=60 time=6.59 ms
64 bytes from 120.219.91.10: icmp_seq=3 ttl=60 time=1.46 ms
64 bytes from 120.219.91.10: icmp_seq=4 ttl=60 time=14.1 ms
64 bytes from 120.219.91.10: icmp_seq=5 ttl=60 time=1.68 ms
64 bytes from 120.219.91.10: icmp_seq=6 ttl=60 time=7.06 ms
64 bytes from 120.219.91.10: icmp_seq=7 ttl=60 time=1.55 ms
64 bytes from 120.219.91.10: icmp_seq=8 ttl=60 time=1.45 ms
64 bytes from 120.219.91.10: icmp_seq=9 ttl=60 time=1.93 ms
64 bytes from 120.219.91.10: icmp_seq=10 ttl=60 time=2.59 ms
```

4. From 79.194.46.0/24 to 120.219.94.0/24(vice versa)

```
root@apollo:/tmp/pycore.45078/apollo.conf# ping 120.219.94.10
PING 120.219.94.10 (120.219.94.10) 56(84) bytes of data.
64 bytes from 120.219.94.10: icmp_seq=1 ttl=59 time=5.17 ms
64 bytes from 120.219.94.10: icmp_seq=2 ttl=59 time=5.86 ms
64 bytes from 120.219.94.10: icmp_seq=3 ttl=59 time=3.03 ms
64 bytes from 120.219.94.10: icmp_seq=4 ttl=59 time=7.93 ms
64 bytes from 120.219.94.10: icmp_seq=5 ttl=59 time=5.56 ms
64 bytes from 120.219.94.10: icmp_seq=6 ttl=59 time=8.45 ms
64 bytes from 120.219.94.10: icmp_seq=7 ttl=59 time=15.7 ms
64 bytes from 120.219.94.10: icmp_seq=8 ttl=59 time=16.2 ms
```

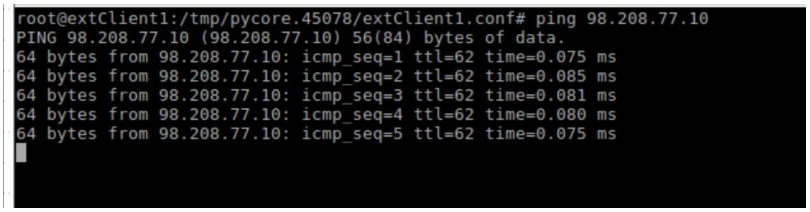
5. From 79.194.46.0/24 to 120.219.109.0/24(vice versa)

```
File  Edit  Tabs  Help
root@apollo:/tmp/pycore.45078/apollo.conf# ping 120.219.109.10
PING 120.219.109.10 (120.219.109.10) 56(84) bytes of data.
64 bytes from 120.219.109.10: icmp_seq=1 ttl=61 time=0.075 ms
64 bytes from 120.219.109.10: icmp_seq=2 ttl=61 time=0.098 ms
64 bytes from 120.219.109.10: icmp_seq=3 ttl=61 time=0.084 ms
64 bytes from 120.219.109.10: icmp_seq=4 ttl=61 time=0.223 ms
64 bytes from 120.219.109.10: icmp_seq=5 ttl=61 time=0.094 ms
64 bytes from 120.219.109.10: icmp_seq=6 ttl=61 time=0.052 ms
```

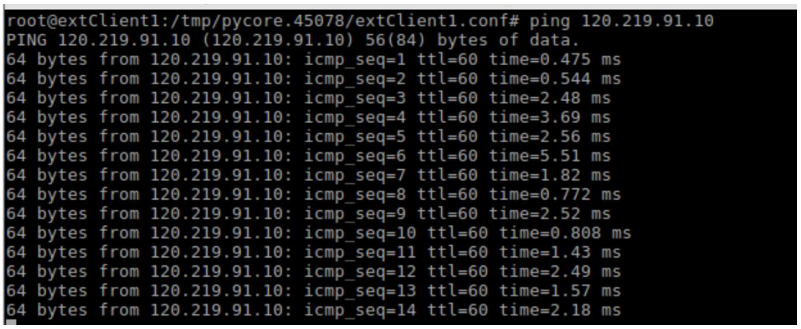
6. From 79.194.46.0/24 to 120.219.152.0/24(vice versa)

```
File  Edit  Tabs  Help
root@apollo:/tmp/pycore.45078/apollo.conf# ping 120.219.152.10
PING 120.219.152.10 (120.219.152.10) 56(84) bytes of data.
64 bytes from 120.219.152.10: icmp_seq=1 ttl=60 time=1.86 ms
64 bytes from 120.219.152.10: icmp_seq=2 ttl=60 time=1.70 ms
64 bytes from 120.219.152.10: icmp_seq=3 ttl=60 time=4.13 ms
64 bytes from 120.219.152.10: icmp_seq=4 ttl=60 time=7.89 ms
64 bytes from 120.219.152.10: icmp_seq=5 ttl=60 time=1.40 ms
64 bytes from 120.219.152.10: icmp_seq=6 ttl=60 time=4.74 ms
64 bytes from 120.219.152.10: icmp_seq=7 ttl=60 time=1.53 ms
```

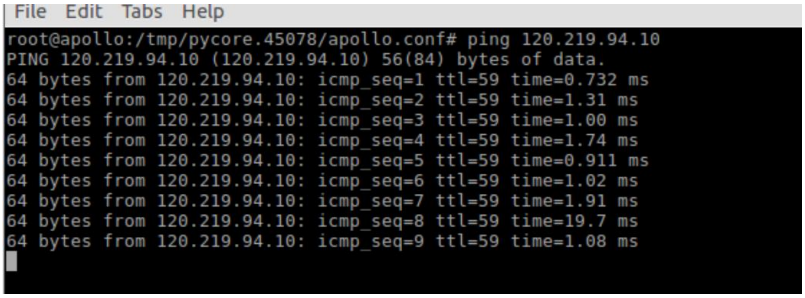
7. From 79.194.55.0/24 to 98.208.77.0/24(vice versa)



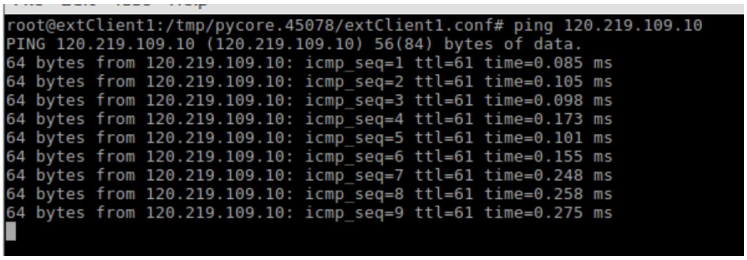
8. From 79.194.55.0/24 to 120.219.91.0/24(vice versa)



9. From 79.194.55.0/24 to 120.219.94.0/24(vice versa)



10. From 79.194.55.0/24 to 120.219.109.0/24(vice versa)



11. From 79.194.55.0/24 to 120.219.152.0/24(vice versa)

```
File  Edit  Tabs  Help
root@extClient1:/tmp/pycore.45078/extClient1.conf# ping 120.219.152.10
PING 120.219.152.10 (120.219.152.10) 56(84) bytes of data.
64 bytes from 120.219.152.10: icmp_seq=1 ttl=60 time=0.608 ms
64 bytes from 120.219.152.10: icmp_seq=2 ttl=60 time=0.903 ms
64 bytes from 120.219.152.10: icmp_seq=3 ttl=60 time=10.1 ms
64 bytes from 120.219.152.10: icmp_seq=4 ttl=60 time=0.578 ms
64 bytes from 120.219.152.10: icmp_seq=5 ttl=60 time=3.87 ms
64 bytes from 120.219.152.10: icmp_seq=6 ttl=60 time=2.18 ms
64 bytes from 120.219.152.10: icmp_seq=7 ttl=60 time=6.20 ms
64 bytes from 120.219.152.10: icmp_seq=8 ttl=60 time=1.30 ms
```

12. From 98.208.77.0/24 to 120.219.91.0/24(vice versa)

```
File  Edit  Tabs  Help
root@clio:/tmp/pycore.45078/clio.conf# ping 120.219.91.10
PING 120.219.91.10 (120.219.91.10) 56(84) bytes of data.
64 bytes from 120.219.91.10: icmp_seq=1 ttl=61 time=0.494 ms
64 bytes from 120.219.91.10: icmp_seq=2 ttl=61 time=1.62 ms
64 bytes from 120.219.91.10: icmp_seq=3 ttl=61 time=5.15 ms
64 bytes from 120.219.91.10: icmp_seq=4 ttl=61 time=1.69 ms
64 bytes from 120.219.91.10: icmp_seq=5 ttl=61 time=3.07 ms
64 bytes from 120.219.91.10: icmp_seq=6 ttl=61 time=2.48 ms
64 bytes from 120.219.91.10: icmp_seq=7 ttl=61 time=1.32 ms
```

13. From 98.208.77.0/24 to 120.219.94.0/24(vice versa)

```
root@clio:/tmp/pycore.45078/clio.conf# ping 120.219.94.10
PING 120.219.94.10 (120.219.94.10) 56(84) bytes of data.
64 bytes from 120.219.94.10: icmp_seq=1 ttl=60 time=2.67 ms
64 bytes from 120.219.94.10: icmp_seq=2 ttl=60 time=2.79 ms
64 bytes from 120.219.94.10: icmp_seq=3 ttl=60 time=2.37 ms
64 bytes from 120.219.94.10: icmp_seq=4 ttl=60 time=2.68 ms
64 bytes from 120.219.94.10: icmp_seq=5 ttl=60 time=6.98 ms
64 bytes from 120.219.94.10: icmp_seq=6 ttl=60 time=6.56 ms
64 bytes from 120.219.94.10: icmp_seq=7 ttl=60 time=7.20 ms
64 bytes from 120.219.94.10: icmp_seq=8 ttl=60 time=4.50 ms
```

14. From 98.208.77.0/24 to 120.219.109.0/24(vice versa)

```
File  Edit  Tabs  Help
root@clio:/tmp/pycore.45078/clio.conf# ping 120.219.109.10
PING 120.219.109.10 (120.219.109.10) 56(84) bytes of data.
64 bytes from 120.219.109.10: icmp_seq=1 ttl=62 time=0.070 ms
64 bytes from 120.219.109.10: icmp_seq=2 ttl=62 time=0.086 ms
64 bytes from 120.219.109.10: icmp_seq=3 ttl=62 time=0.129 ms
64 bytes from 120.219.109.10: icmp_seq=4 ttl=62 time=0.086 ms
64 bytes from 120.219.109.10: icmp_seq=5 ttl=62 time=0.225 ms
64 bytes from 120.219.109.10: icmp_seq=6 ttl=62 time=0.209 ms
64 bytes from 120.219.109.10: icmp_seq=7 ttl=62 time=0.176 ms
```

15. From 98.208.77.0/24 to 120.219.152.0/24(vice versa)

```
root@clio:/tmp/pycore.45078/clio.conf# ping 120.219.152.10
PING 120.219.152.10 (120.219.152.10) 56(84) bytes of data.
64 bytes from 120.219.152.10: icmp_seq=1 ttl=61 time=0.515 ms
64 bytes from 120.219.152.10: icmp_seq=2 ttl=61 time=0.783 ms
64 bytes from 120.219.152.10: icmp_seq=3 ttl=61 time=8.24 ms
64 bytes from 120.219.152.10: icmp_seq=4 ttl=61 time=1.74 ms
64 bytes from 120.219.152.10: icmp_seq=5 ttl=61 time=0.980 ms
64 bytes from 120.219.152.10: icmp_seq=6 ttl=61 time=0.744 ms
```

16. From 120.219.91.0/24 to 120.219.94.0/24(vice versa)

```
File  Edit  Tabs  Help
root@localweb:/tmp/pycore.45078/localweb.conf# ping 120.219.94.10
PING 120.219.94.10 (120.219.94.10) 56(84) bytes of data.
64 bytes from 120.219.94.10: icmp_seq=1 ttl=62 time=0.570 ms
64 bytes from 120.219.94.10: icmp_seq=2 ttl=62 time=1.14 ms
64 bytes from 120.219.94.10: icmp_seq=3 ttl=62 time=1.27 ms
64 bytes from 120.219.94.10: icmp_seq=4 ttl=62 time=3.43 ms
64 bytes from 120.219.94.10: icmp_seq=5 ttl=62 time=1.30 ms
64 bytes from 120.219.94.10: icmp_seq=6 ttl=62 time=19.3 ms
64 bytes from 120.219.94.10: icmp_seq=7 ttl=62 time=2.09 ms
```

17. From 120.219.91.0/24 to 120.219.109.0/24(vice versa)

LXTerminal

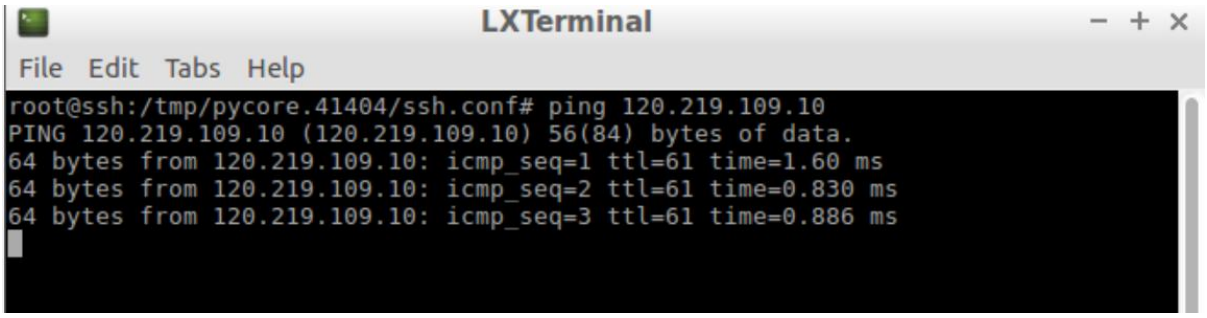
```
File  Edit  Tabs  Help
root@localweb:/tmp/pycore.41404/localweb.conf# ping 120.219.109.10
PING 120.219.109.10 (120.219.109.10) 56(84) bytes of data.
64 bytes from 120.219.109.10: icmp_seq=1 ttl=62 time=7.50 ms
64 bytes from 120.219.109.10: icmp_seq=2 ttl=62 time=3.65 ms
64 bytes from 120.219.109.10: icmp_seq=3 ttl=62 time=3.64 ms
64 bytes from 120.219.109.10: icmp_seq=4 ttl=62 time=0.860 ms
```

18. From 120.219.91.0/24 to 120.219.152.0/24(vice versa)

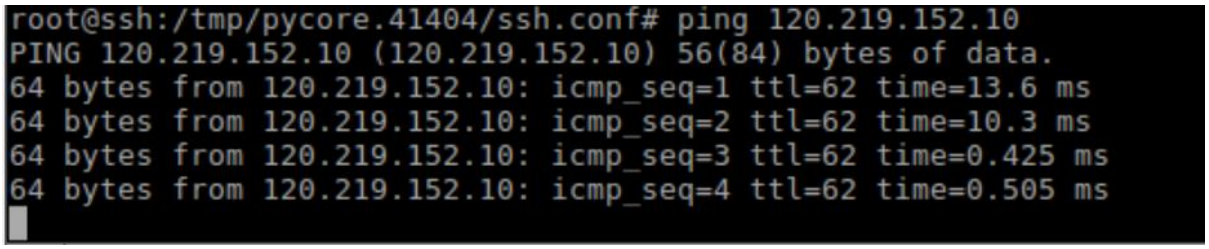
LXTerminal

```
File  Edit  Tabs  Help
root@localweb:/tmp/pycore.41404/localweb.conf# ping 120.219.152.10
PING 120.219.152.10 (120.219.152.10) 56(84) bytes of data.
64 bytes from 120.219.152.10: icmp_seq=1 ttl=62 time=1.95 ms
64 bytes from 120.219.152.10: icmp_seq=2 ttl=62 time=0.697 ms
64 bytes from 120.219.152.10: icmp_seq=3 ttl=62 time=0.398 ms
```

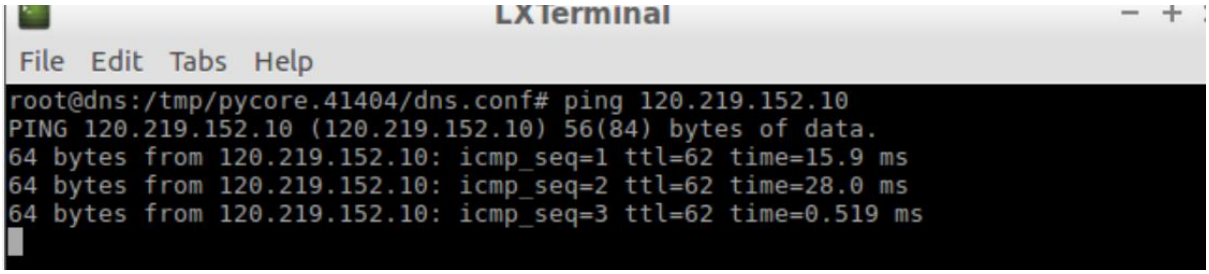
19. From 120.219.94.0/24 to 120.219.109.0/24(vice versa)



20. From 120.219.94.0/24 to 120.219.152.0/24(vice versa)



21. From 120.219.109.0/24 to 120.219.152.0/24(vice versa)



Test passed!

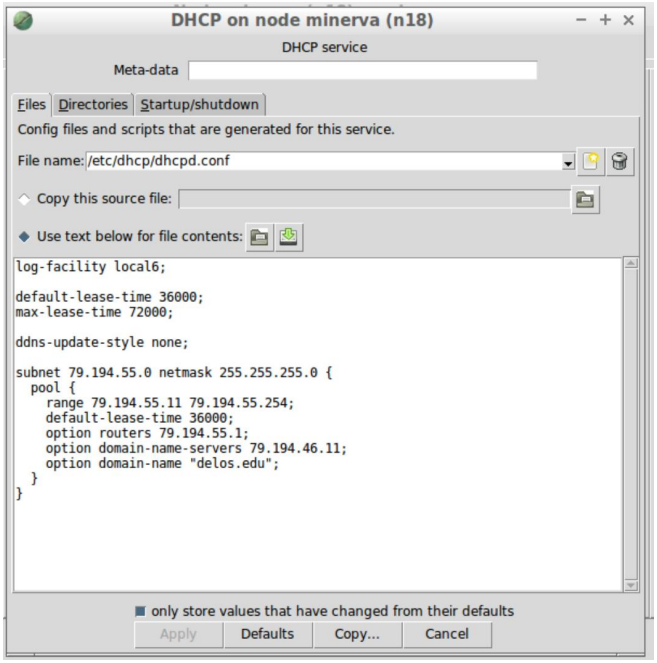
Task B: DHCP Server

Part one: configuration

B.1.1 Minerva

1. Changes

```
log-facility local6;
default-lease-time 36000;
max-lease-time 72000;
ddns-update-style none;
subnet 79.194.55.0 netmask 255.255.255.0 {
   pool {
      range 79.194.55.11 79.194.55.254;
      default-lease-time 36000;
      option routers 79.194.55.1;
      option domain-name-servers 79.194.46.11;
      option domain-name "delos.edu";
   }
}
```



2. Reasons for changes

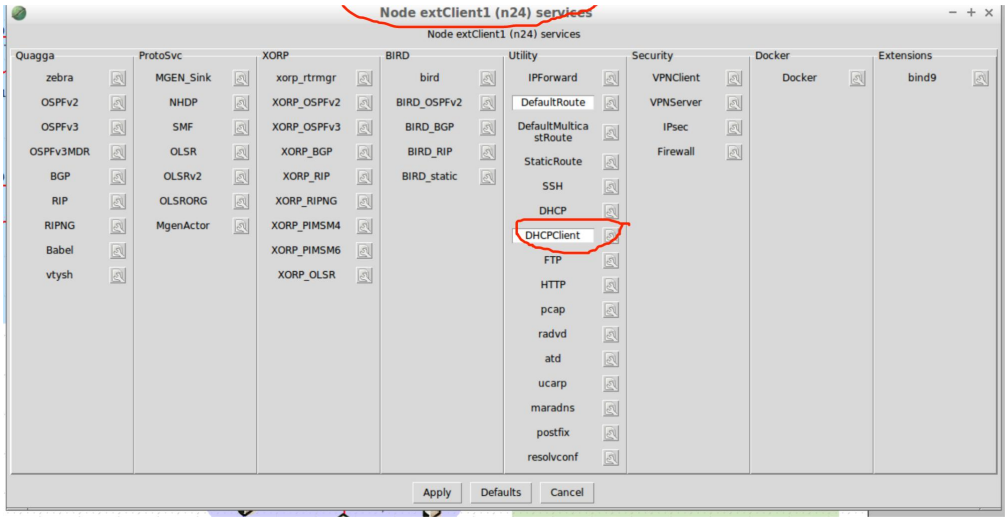
The first line describes the logging facility of the DHCP server. *Default-lease-time 36000* specifies the default time allocated to any client to use the dynamic ip address. *Max-lease-time 72000* means a client cannot ask to rent an ip address for more than 72000 seconds. The fourth line forbids dynamic dns updates. 79.194.55.0 is the network address of the subnet, and 255.255.255.0 is its subnet mask.

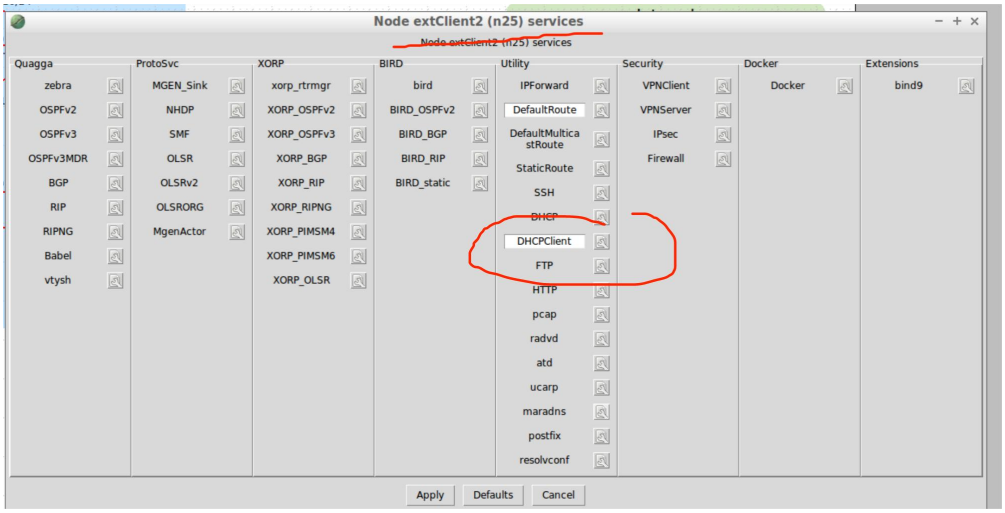
Within the pool block, the first line determines the range of ip address to be assigned. I begin from 79.194.55.11 because 79.194.55.1 and 79.194.55.10 are already used by Minerva and leto. 79.194.55.255 is the broadcast address which cannot be used either. *Option routers 79.194.55.1* is the default gateway of DHCP clients. 79.194.46.11 is the ip address of artemis which is the dns server. The last line records the domain name for the DHCP clients.

B.1.2 Clients of delos

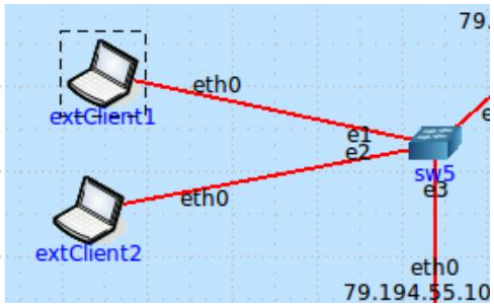
1. Changes

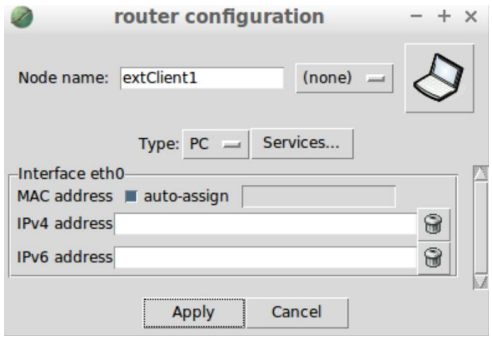
i) Open dhcp client service

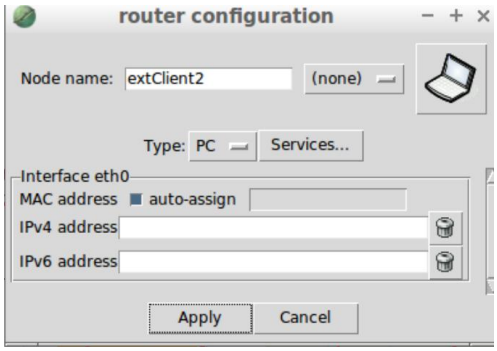




ii) Delete clients's static ip address:







2. Reasons for changes

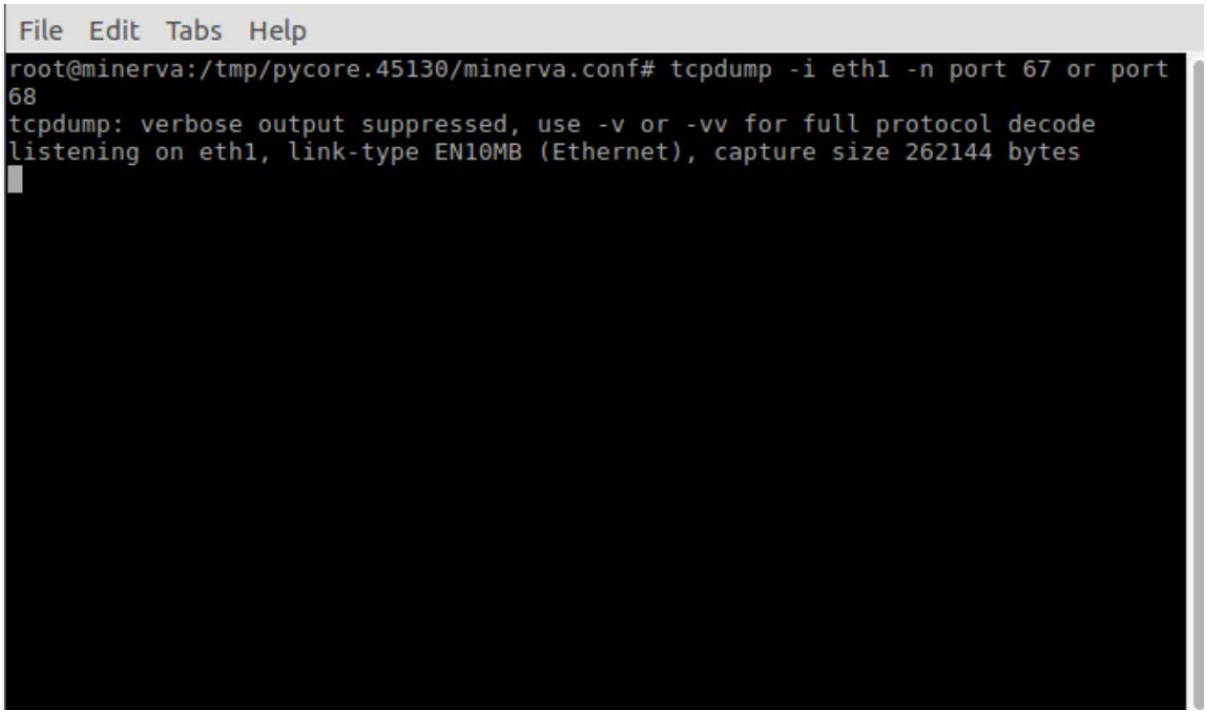
To ensure clients can receive ip address dynamically as DHCP clients.

Part two: testing

I will send a DHCP request to the DHCP server from a specific DHCP client and then use tcpdump command to capture the DHCP request and reply on DHCP server.

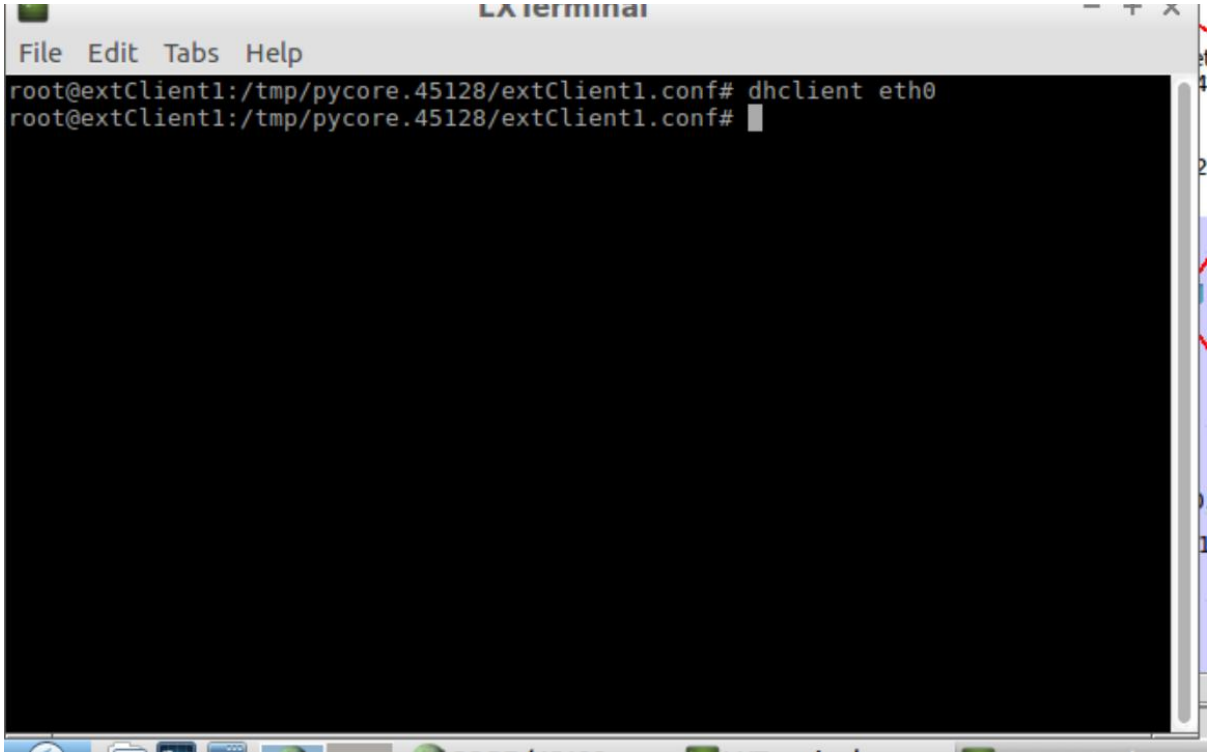
First step:

Catch DHCP packets on DHCP server.



Second step:

Request an ip address on extClient1.



Third step:

The result on DHCP server.

The display information shows extClient1 has been assigned a new ip address of 79.194.55.13, so it turns out the DHCP configuration works.

Test passed!

Task C: Firewall

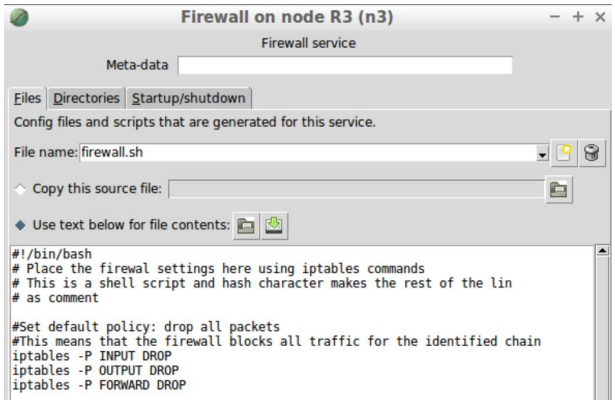
0. Default policy

A. Configuration

iptables -P INPUT DROP

iptables -P OUTPUT DROP

iptables -P FORWARD DROP

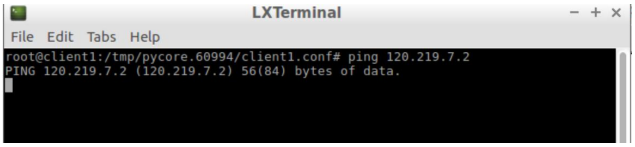


Drop all input, output and forward traffic on R3.

B. Testing

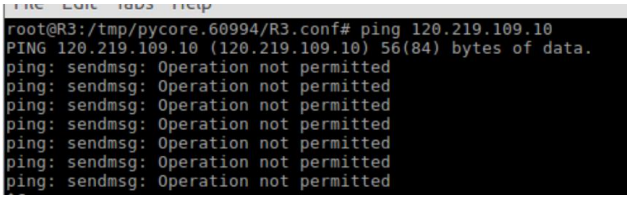
i) Input

Ping firewall from client1 of talos:



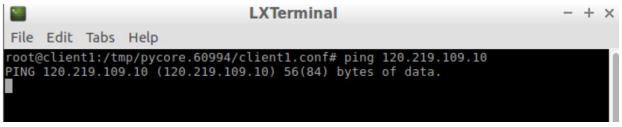
ii) Output

From firewall ping talos' dns server:



iii) Forward

From talos' client1 ping talos' dns server through firewall:



Test passed!

1. Anywhere to DMZ

A. Configuration

#web

iptables -A FORWARD -o eth0 -d 120.219.109.11 -p tcp --dport 80 -j ACCEPT

iptables -A FORWARD -i eth0 -s 120.219.109.11 -p tcp --sport 80 -j ACCEPT

#dns

iptables -A FORWARD -o eth0 -d 120.219.109.10 -p udp --dport 53 -j ACCEPT

iptables -A FORWARD -i eth0 -s 120.219.109.10 -p udp --sport 53 -j ACCEPT

#mail

iptables -A FORWARD -o eth0 -d 120.219.109.12 -p tcp --dport 25 -j ACCEPT

iptables -A FORWARD -i eth0 -s 120.219.109.12 -p tcp --sport 25 -j ACCEPT

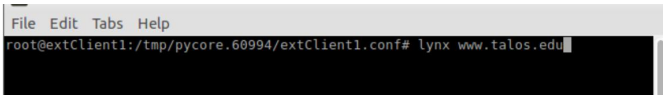
```
#Anywhere to DMZ
#web
iptables -A FORWARD -o eth0 -d 120.219.109.11 -p tcp --dport 80 -j ACCEPT
iptables -A FORWARD -i eth0 -s 120.219.109.11 -p tcp --sport 80 -j ACCEPT
#dns
iptables -A FORWARD -o eth0 -d 120.219.109.10 -p udp --dport 53 -j ACCEPT
iptables -A FORWARD -i eth0 -s 120.219.109.10 -p udp --sport 53 -j ACCEPT
#mail
iptables -A FORWARD -o eth0 -d 120.219.109.12 -p tcp --dport 25 -j ACCEPT
iptables -A FORWARD -i eth0 -s 120.219.109.12 -p tcp --sport 25 -j ACCEPT
```

-i stands for input interface, -o for output interface, -d for destination ip address, -s for source ip address. -p stands for protocol. Tcp and udp are two different transport layer protocols, and tcp is used where network stabiliby is required. --dport represents destination port, --sport is source port. Port 80 is reserved to web service, 53 for dns service and 25 for mail service.

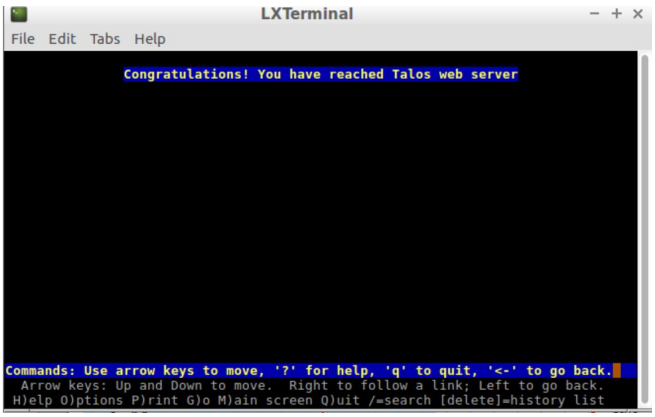
B. Testing

From extClient1 use different commands to connect to dns, web and mail respectively.

i) Dns



If dns service is accessible, it will translate this domain name into an ip address of a web server so that extClient1 can reach the corresponding web server.



ii) Web

iii) Mail

Telnet command can be used to test whether a specific port on a remote server is accessible.

Here we choose port 25 to test mail service.

```
root@extClient1:/tmp/pycore.60994/extClient1.conf# telnet 120.219.109.12 25
```

```
root@extClient1:/tmp/pycore.60994/extClient1.conf# telnet 120.219.109.12 2
Trying 120.219.109.12...
Connected to 120.219.109.12.
Escape character is '^]'.
220 mail ESMTP Postfix (Ubuntu)
```

Test passed!

2. DMZ to external

Web server in DMZ does not need to connect to the external servers in that internal http request will go to external web server directly without going through the web server in DMZ.

Dns server in DMZ should have access to the global dns, because internal dns request needs to go through DMZ's dns to reach global dns in order to access the ip address of www.delos.edu. Internal dns request cannot go directly to external dns server.

Mail server in DMZ needs to connect to the mail server demeter of delos, because mails need to exchange between the two organisations. Clients will use their own mail server and exchange mails between the two servers.

A. Configuration

#dns

iptables -A FORWARD -i eth0 -o eth3 -s 120.219.109.10 -d 98.208.77.10 -p udp --dport 53 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A FORWARD -i eth3 -o eth0 -s 98.208.77.10 -d 120.219.109.10 -p udp --sport 53 -m state --state ESTABLISHED -j ACCEPT

#mail

iptables -A FORWARD -i eth0 -o eth3 -s 120.219.109.12 -d 79.194.46.12 -p tcp --dport 25 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A FORWARD -i eth3 -o eth0 -s 79.194.46.12 -d 120.219.109.12 -p tcp --sport 25 -m state --state ESTABLISHED -j ACCEPT

```
#DMZ to external
#dns
iptables -A FORWARD -i eth0 -o eth3 -s 120.219.109.10 -d 98.208.77.10 -p udp --d
port 53 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i eth3 -o eth0 -s 98.208.77.10 -d 120.219.109.10 -p udp --s
port 53 -m state --state ESTABLISHED -j ACCEPT
#mail
iptables -A FORWARD -i eth0 -o eth3 -s 120.219.109.12 -d 79.194.46.12 -p tcp --d
port 25 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i eth3 -o eth0 -s 79.194.46.12 -d 120.219.109.12 -p tcp --s
port 25 -m state --state ESTABLISHED -j ACCEPT
```

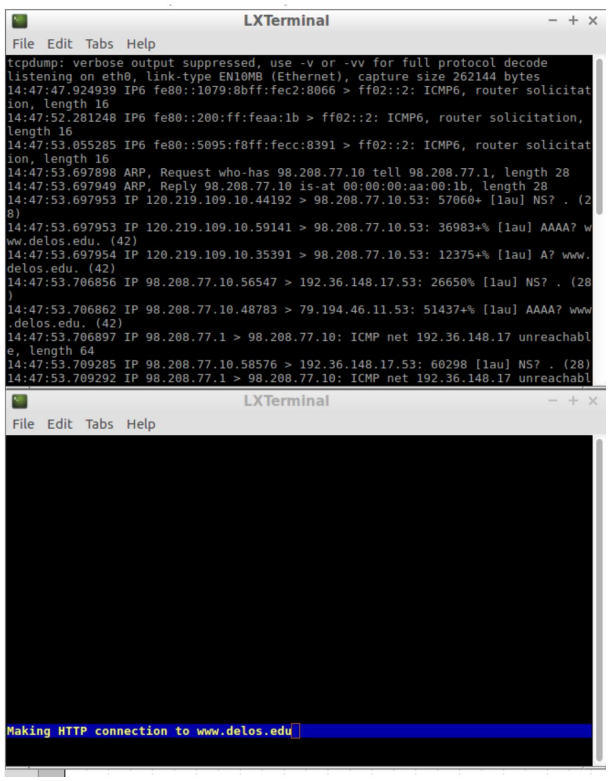
-m state is a stateful inspection, and by using established I limit the traffic from external to those as a reply for the request from DMZ.

B. Testing

i) Dns

Try to connect to a web, and dns server will send a dns request to the global dns clio

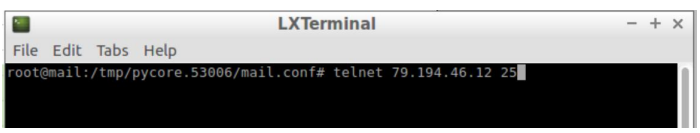
Use tcpdump command on clio to capture the traffic.

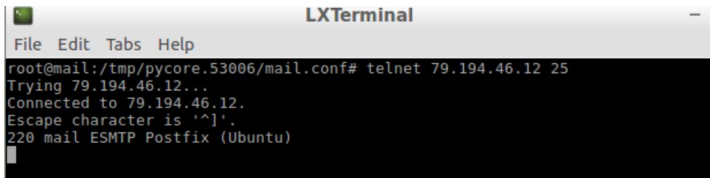


Clio successfully received the dns request.

ii) Mail

79.194.46.12 is demeter's ip address, which is the mail server of delos.





Test passed!

3. Internal to DMZ

Need to use dns, web, mail and ssh services.

A. Configuration

#dns

iptables -A FORWARD -o eth0 -s 120.219.0.0/16 -d 120.219.109.10 -p udp --dport 53 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A FORWARD -i eth0 -s 120.219.109.10 -d 120.219.0.0/16 -p udp --sport 53 -m state --state ESTABLISHED,RELATED -j ACCEPT

#web

iptables -A FORWARD -o eth0 -s 120.219.0.0/16 -d 120.219.109.11 -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A FORWARD -i eth0 -s 120.219.109.11 -d 120.219.0.0/16 -p tcp --sport 80 -m state --state ESTABLISHED,RELATED -j ACCEPT

#mail

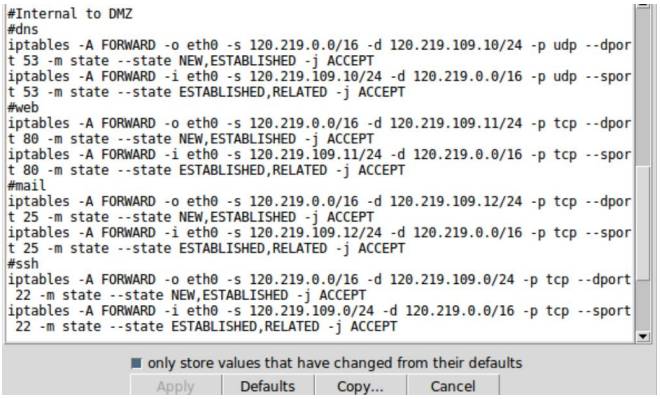
iptables -A FORWARD -o eth0 -s 120.219.0.0/16 -d 120.219.109.12 -p tcp --dport 25 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A FORWARD -i eth0 -s 120.219.109.12 -d 120.219.0.0/16 -p tcp --sport 25 -m state --state ESTABLISHED,RELATED -j ACCEPT

#ssh

iptables -A FORWARD -o eth0 -s 120.219.0.0/16 -d 120.219.109.0/24 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A FORWARD -i eth0 -s 120.219.109.0/24 -d 120.219.0.0/16 -p tcp --sport 22 -m state --state ESTABLISHED,RELATED -j ACCEPT



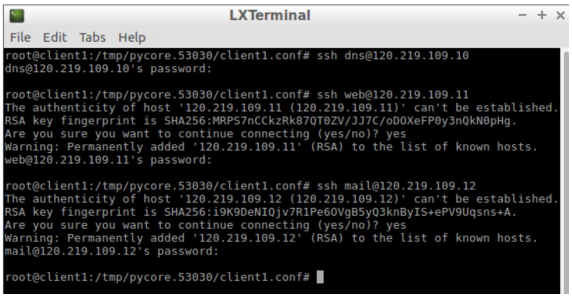
Based on question 1, here stricter stateful inspection is configured to limit traffic from DMZ to internal.

In addition, ssh servce use tcp protocol on port 22.

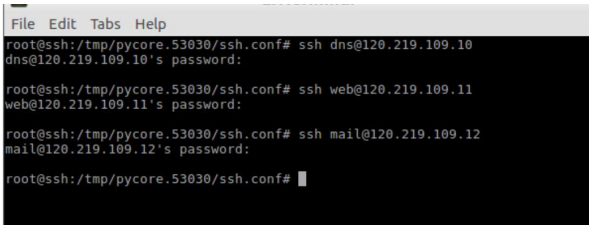
B. Testing

Ssh command can build a secured tunnel between two nodes using ssh service.

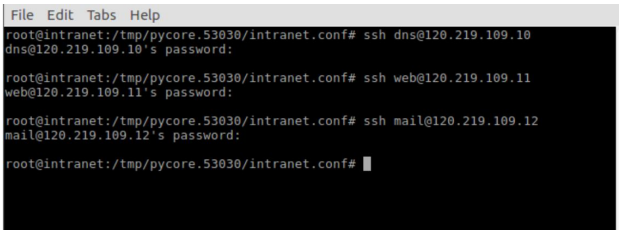
i) 120.219.91.0/24 to DMZ



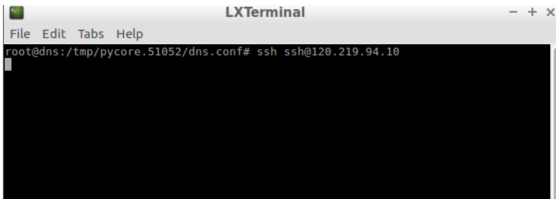
ii) 120.219.94.0/24 to DMZ



iii) 120.219.152.0/24 to DMZ



iv) From DMZ to internal ssh server



Cannot reach due to stateful inspection

Test passed!

4. Internal to internal

Since all internal traffic in my design do not go through the firewall, there is no need to modify the configuration.

5. Internal to external servers

Internal nodes only need access to external web, mail and ssh services. Dns request must go through dns server in DMZ to reach the global dns server clio, so it cannot be directly configured here. It is noticeable that there is another web server called leto in delos.

A. Configuration

Using subnet mask /16 to get the network address of delos.

#web

iptables -A FORWARD -i eth1 -o eth3 -d 79.194.0.0/16 -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A FORWARD -i eth3 -o eth1 -s 79.194.0.0/16 -p tcp --sport 80 -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -A FORWARD -i eth2 -o eth3 -d 79.194.0.0/16 -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A FORWARD -i eth3 -o eth2 -s 79.194.0.0/16 -p tcp --sport 80 -m state --state ESTABLISHED,RELATED -j ACCEPT

#mail

iptables -A FORWARD -i eth1 -o eth3 -d 79.194.46.12 -p tcp --dport 25 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A FORWARD -i eth3 -o eth1 -s 79.194.46.12 -p tcp --sport 25 -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -A FORWARD -i eth2 -o eth3 -d 79.194.46.12 -p tcp --dport 25 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A FORWARD -i eth3 -o eth2 -s 79.194.46.12 -p tcp --sport 25 -m state --state ESTABLISHED,RELATED -j ACCEPT

#ssh

iptables -A FORWARD -i eth1 -o eth3 -d 79.194.46.0/24 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A FORWARD -i eth3 -o eth1 -s 79.194.46.0/24 -p tcp --sport 22 -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -A FORWARD -i eth2 -o eth3 -d 79.194.46.0/24 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A FORWARD -i eth3 -o eth2 -s 79.194.46.0/24 -p tcp --sport 22 -m state --state ESTABLISHED,RELATED -j ACCEPT

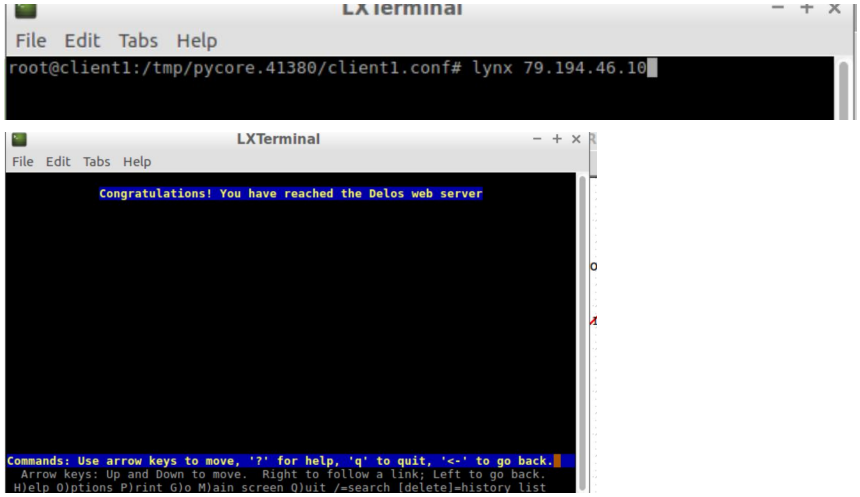
```
#internal to external servers
#web
iptables -A FORWARD -i eth1 -o eth3 -d 79.194.0.0/16 -p tcp --dport 80 -m state
--state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i eth3 -o eth1 -s 79.194.0.0/16 -p tcp --sport 80 -m state
--state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -i eth2 -o eth3 -d 79.194.0.0/16 -p tcp --dport 80 -m state
--state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i eth3 -o eth2 -s 79.194.0.0/16 -p tcp --sport 80 -m state
--state ESTABLISHED,RELATED -j ACCEPT
#mail
iptables -A FORWARD -i eth1 -o eth3 -d 79.194.46.12 -p tcp --dport 25 -m state -
-state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i eth3 -o eth1 -s 79.194.46.12 -p tcp --sport 25 -m state -
-state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -i eth2 -o eth3 -d 79.194.46.12 -p tcp --dport 25 -m state -
-state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i eth3 -o eth2 -s 79.194.46.12 -p tcp --sport 25 -m state -
-state ESTABLISHED.RELATED -i ACCEPT
```

```
#ssh
iptables -A FORWARD -i eth1 -o eth3 -d 79.194.46.0/24 -p tcp --dport 22 -m state
 --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i eth3 -o eth1 -s 79.194.46.0/24 -p tcp --sport 22 -m state
 --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -i eth2 -o eth3 -d 79.194.46.0/24 -p tcp --dport 22 -m state
 --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i eth3 -o eth2 -s 79.194.46.0/24 -p tcp --sport 22 -m state
 --state ESTABLISHED,RELATED -j ACCEPT
```

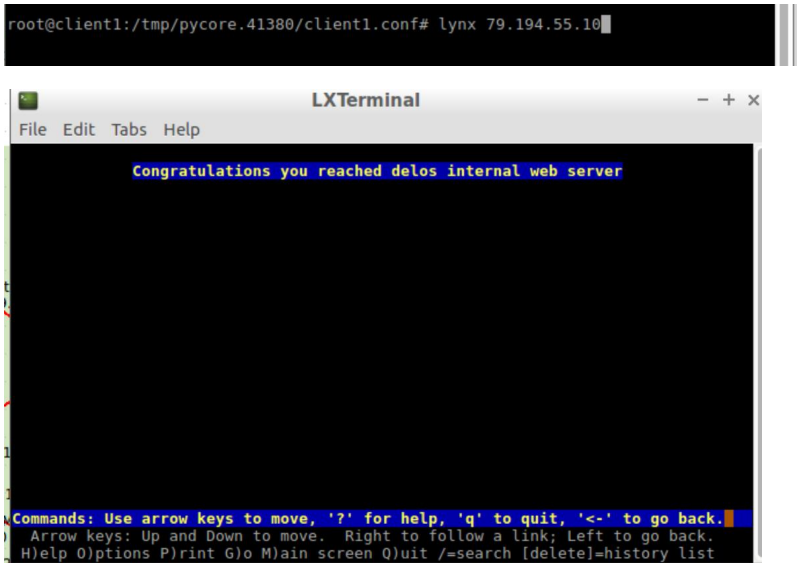
B. Testing

i) Web

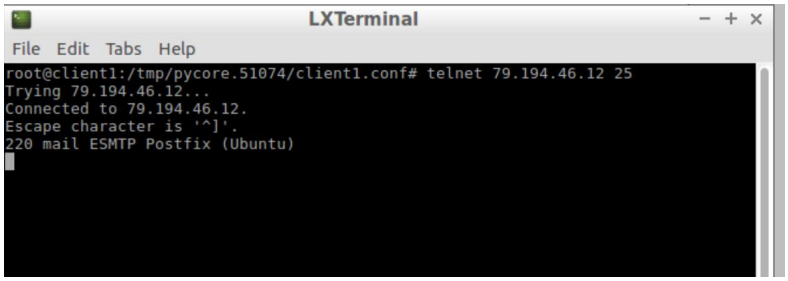
apollo



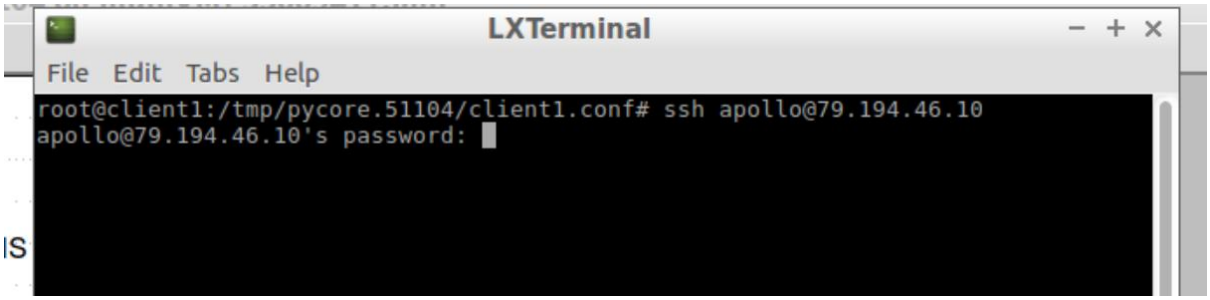
leto



ii) Mail



iii) Ssh



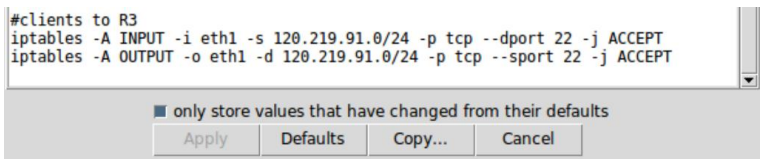
Test passed!

6. Clients to R3

A. Configuration

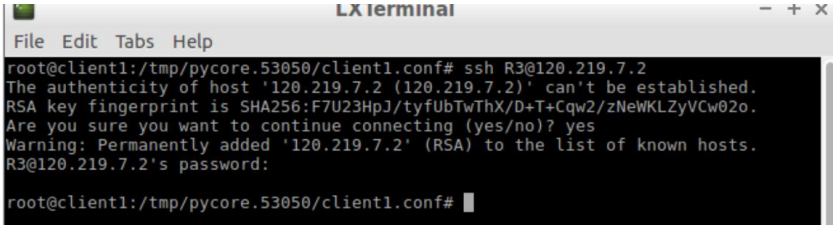
iptables -A INPUT -i eth1 -s 120.219.91.0/24 -p tcp --dport 22 -j ACCEPT

iptables -A OUTPUT -o eth1 -d 120.219.91.0/24 -p tcp --sport 22 -j ACCEPT



B. Testing

From client1, use ssh command to reach R3.



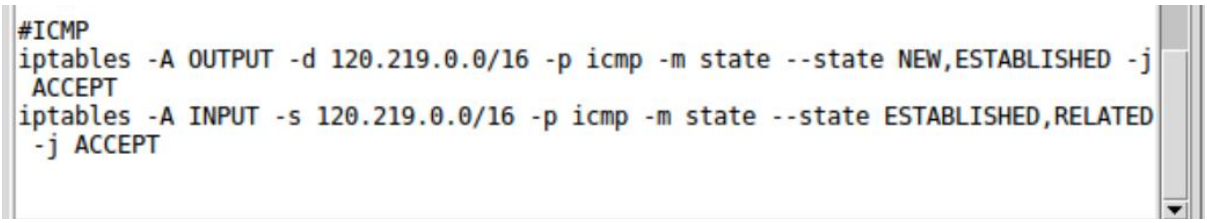
Test passed!

7. R3 ICMP

A. Configuration

iptables -A OUTPUT -d 120.219.0.0/16 -p icmp -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A INPUT -s 120.219.0.0/16 -p icmp -m state --state ESTABLISHED,RELATED -j ACCEPT

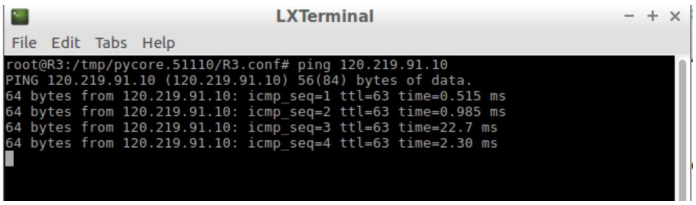


Here use icmp protocol.

Only icmp request from R3 is allowed, and request from outside is forbidden. Instead, only icmp reply from outside is allowed. That is why the stateful inspection is configured here.

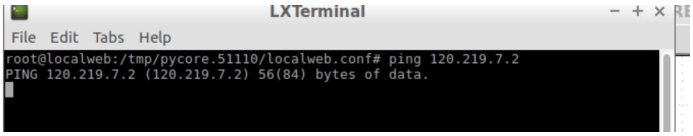
B. Testing

i) R3 to internal



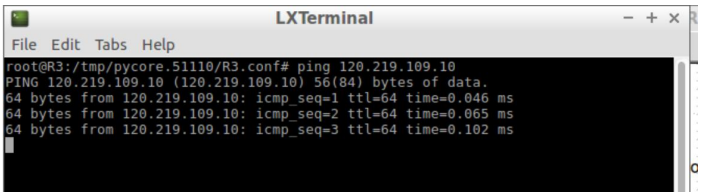
Succeeded

Vice versa



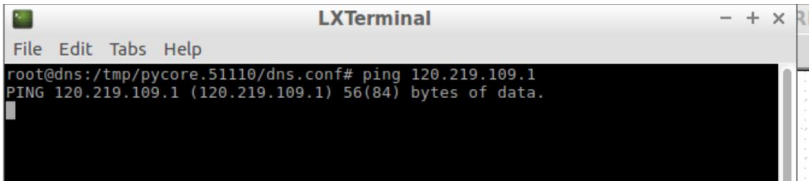
Failed

ii) R3 to DMZ



Succeeded

Vice versa



Failed

Test passed!

End of report.

Acknowledgment of ChatGPT Usage

In the preparation of this work, I utilized OpenAI's ChatGPT model to obtain preliminary information on the subject matter. The content provided by ChatGPT was subsequently reviewed, refined, and supplemented with additional research and personal analysis to ensure the accuracy and originality of the final submission.