RFP Report for FIT9137 A3

TASK A: ROUTING

This task involves configuring all routers for static routing to ensure valid traffic flow and prevent routing loops.

Configurations:

In this task, we use the command:

```
/sbin/ip route add 94.78.71.0/24 via 94.78.71.1
```

In this task, we have two approaches to routing:

1. Route only for the directly connected node in each router
2. Route for all existing LANs, regardless of distance

I chose the second strategy as it focuses more on subnet communication and is easier to verify and comprehend. (Consequently, each router will have 7 routing rules, corresponding to the 7 subnets in total.)

A.1 R1 Configuration

- For routing to nodes behind the gateway router R3, we have two options that differ in delay and speed. As per the requirement to ignore delay, we consistently choose the path with the highest speed of 1 Gbps.
- For the remaining connections to other LANs, we configure the static routing table with the most direct path.

A.2 R2 Configuration

- Since R2 is directly connected to R3 and offers the highest speed, we choose the path to R3 for all external connections.

- For internal connections, we select the shortest path as it provides direct access with sufficient bandwidth.

A.3 R3 Configuration

- For internal routing, we select the path with higher speed.
- For external routing, we choose the only available path, as R3 serves as the gateway to the internal network.

A.4 R4 Configuration

- We prioritize the path with the larger bandwidth. As a result, we route most external connections through R2.

A.5 Other Configuration

- We follow the same strategy to give configurations for Minerva and internet routers.

A.6 Default Gate Configuration

```
/sbin/ip route add default via 47.36.115.1
```

- For R1 and R2, we only keep R3 as the default routing.
- For R4 we pick R2 because it's the fastest path.
- For R3 and Minerva routers, we pick the default routing on the Internet

A.7 Loop Routing Concern

We have two methods to treat the path with insufficient bandwidth：

1. keep the path and use **metric** keywords in the command to decrease the priority
2. directly abandon all paths with lower bandwidth

- In conclusion, I picked the second method so this directly breaks the loop pattern in the paths between the routers in the Talos network. Thus a higher

bandwidth is always obtained, and the possibility of a loop routing is eliminated at the same time.

Verification

After all static routing is completed, we could test any traffic between DMZ, Delos to Talos when we run the simulation session.

> 💡 We have two methods to ping from one node to another node:
>
> 1. open the terminal of the source node and ping the destination IP address
> 2. use the “two node tool” of the core emulator to check the connections

TASK B: DHCP

This task is about configuring the DHCP services to give dynamic IP assigned to the clients in the Delos network.

Configurations:

B.1 DHCP Configuration

We configure the DHCP service on Minerva based on R1 configuration.

```
log-facility local6;

default-lease-time 36000;
max-lease-time 72000;

ddns-update-style none;


subnet 43.131.110.0 netmask 255.255.255.0 {
```

```
    pool {
      range 43.131.110.100 43.131.110.200 ;
      default-lease-time 36000;
      option routers 43.131.110.1;
      option domain-name-servers 60.98.165.10;
      option domain-name "delos.edu";
    }
}
```

1. pick subnet 43.131.110.0 to match the client subnet
2. For the netmask, using 255.255.255.0(/24) as it's general in the whole network
3. Allocate the pool of IP addresses from .100 to .200
4. Modify the option router to the original default gateway which is eth1 of Minerva
5. Set the IP address of global DNS for the domain-name-servers configuration.

B.2 DHCP Client Configuration

We simply drop the default IP of the two clients in Delos and turn on the DHCP Client options in the services configuration.

Leto is not involved in this task.

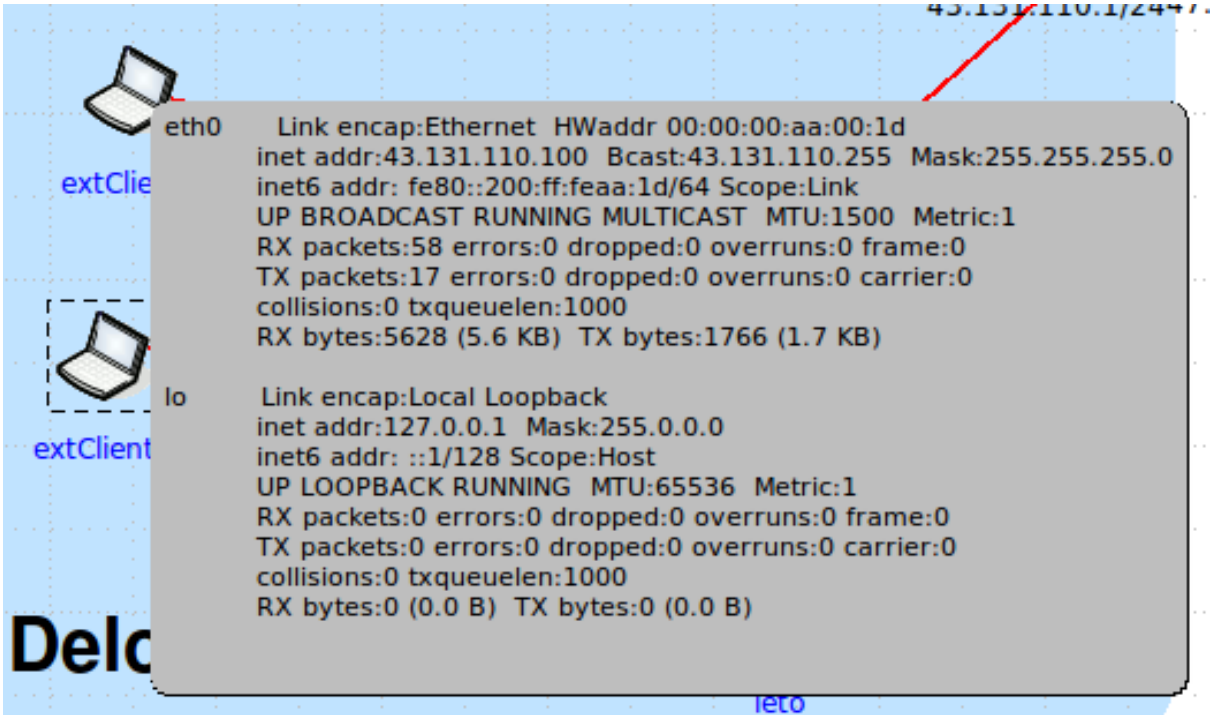
Verifications

We run the simulation sessions and observe the IP addresses of the clients in the Delos network.

We have two methods to check IP address based on ifconfig:

1. directly open the terminal to run the command ifconfig
2. use this tool to check any nodes on the diagram

As a result, we can find our clients have been allocated a new IP address starting from .100:



These two clients own the IP addresses 43.131.110.100 and 43.131.110.101, if another client is added in this subnet, it will obtain an IP address of 43.131.110.102.

TASK C: FIREWALL

This task involves configuring the firewall service to allow only specific types of traffic. Throughout the configuration process, tests will be conducted incrementally.

Prepare

Firstly, we disable all traffic:

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

Explanation

- INPUT DROP for dropping all input traffic to the firewall node
- OUTPUT DROP for dropping all output traffic from the firewall node
- FORWARD DROP for dropping all passing traffic through the firewall node

C.1 Open public services for all nodes

We modify the rules for these public service servers both as a source and as a destination.

```
iptables -A FORWARD -p udp --dport 53 -d 94.78.215.10 -j ACCE
iptables -A FORWARD -p udp --sport 53 -s 94.78.215.10 -j ACCE

iptables -A FORWARD -p tcp --dport 80 -d 94.78.215.11 -j ACCE
iptables -A FORWARD -p tcp --sport 80 -s 94.78.215.11 -j ACCE

iptables -A FORWARD -p tcp --dport 25 -d 94.78.215.12 -j ACCE
iptables -A FORWARD -p tcp --sport 25 -s 94.78.215.12 -j ACCE
```

Explanation

based on the principle that

- DNS services use UDP 53
- web services use TCP 80
- mail use TCP 25

Otherwise,

- A is for adding

- p is for protocal
- dport is for destination port
- sport is for source port
- d is for destination IP
- s is for source IP

Test

After two directions are implemented,

- test web by lynx command
- test DNS by the lynx command but not with IP
- test mail by Telnet

```
lynx <web ip address>         ->WEB
lynx www.talos.edu            ->DNS
telnet <mail ip address> 25   ->MAIL
```

Explanation

- lynx is a browser function to check the interactions of web services
- After DNS services is on, we can access web services only using domain name as DNS will match the domain name and IP
- telnet is used to test SMTP, and it's using port 25

C.2 Allow the DMZ server to initiate communications

We use state **NEW** to permit all new communications which is sourcing from the DMZ servers

```
iptables -A FORWARD -s 94.78.215.10 -m state --state NEW -j A
iptables -A FORWARD -s 94.78.215.11 -m state --state NEW -j A
iptables -A FORWARD -s 94.78.215.12 -m state --state NEW -j A
```

Test

In this case, we only allow the DMZ to initiate communications but not allow DMZ to receive, so we could test using **tcpdump** command to see if there is traffic after the firewall.

- For example, we configure the rule for the DNS server, when we try pinging the intranet server, the firewall router R3 will forward it to R2 and then the subnet, we use **tcpdump** on R2 to check if the request and reply are passing.

This is the successful attempt:

```
root@R2:/tmp/pycore.57970/R2.conf# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C22:01:33.109426 ARP, Request who-has 94.78.5.10 tell 94.78.5.1, length 28
22:01:33.109463 ARP, Reply 94.78.5.10 is-at 00:00:00:aa:00:08 (oui Ethernet), le
ngth 28
22:01:33.109465 IP 94.78.215.11 > 94.78.5.10: ICMP echo request, id 185, seq 1,
length 64
22:01:33.109473 IP 94.78.5.10 > 94.78.215.11: ICMP echo reply, id 185, seq 1, le
ngth 64
22:01:34.673780 IP6 fe80::200:ff:feaa:8 > ip6-allrouters: ICMP6, router solicita
tion, length 16

5 packets captured
5 packets received by filter
0 packets dropped by kernel
root@R2:/tmp/pycore.57970/R2.conf#
```

If not configured properly, there would be no traffic captured in this R2's tcpdump result:

```
root@R2:/tmp/pycore.57970/R2.conf# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
```

C.3 Allow internal hosts to have full access to DMZ servers

We allow all Talos subnets to access DMZ servers

```
iptables -A FORWARD -s 94.78.71.0/24 -d 94.78.215.0/24 -j ACC
iptables -A FORWARD -s 94.78.182.0/24 -d 94.78.215.0/24 -j AC
```

```
iptables -A FORWARD -s 94.78.5.0/24 -d 94.78.215.0/24 -j ACCE
iptables -A FORWARD -d 94.78.71.0/24 -s 94.78.215.0/24 -j ACC
iptables -A FORWARD -d 94.78.182.0/24 -s 94.78.215.0/24 -j AC
iptables -A FORWARD -d 94.78.5.0/24 -s 94.78.215.0/24 -j ACCE
```

Test

to check the full access, we use an SSH command from an internal node to DMZ to get the results like:

```
root@client1:/tmp/pycore.58008/client1.conf# ssh client1@94.78.215.11
client1@94.78.215.11's password:
```

If not configured properly, this command will give no response.

Explanation

SSHis a cryptographic network protocol used for securing communication.

After full access is gained, we can access any services in the command:

```
ssh <user_name>@<target_ipaddress>
```

C.4 Allow internal to internal communications in Talos

the inner network is already very clear, we just add rules for eth1 and eth2 communications if there are any.

```
iptables -A FORWARD -i eth1 -o eth2 -j ACCEPT
iptables -A FORWARD -i eth2 -o eth1 -j ACCEPT
```

C.5 Allow internal to external communications with special rules

we add rules for eth1/eth2(inner interfaces) and eth3(outer interface):

- We ensure the inner subnet can be **NEW/ESTABLISHED/RELATED** to the outer interface
- For traffic from outside to inside, we only allow **ESTABLISHED/RELATED**

```
iptables -A FORWARD -i eth1 -o eth3 -m state --state NEW,RELA
iptables -A FORWARD -i eth3 -o eth1 -m state --state RELATED,
iptables -A FORWARD -i eth2 -o eth3 -m state --state NEW,RELA
iptables -A FORWARD -i eth3 -o eth2 -m state --state RELATED,
```

Explanation

- **NEW** is for packets which initiate a new connection
- **ESTABLISHED** is for packets which is part of an existing communiction
- **RELATED** is for packets which is related to an existing communiction, but not part of it

Test

To Test this we can simply try initiating the communications from outside to inside and from inside to outside.

C.6 Allow Talos clients to SSH to the firewall node

we need to approve the 94.78.71.0/24 subnet to directly access R3 using SSH, we use INPUT and OUTPUT for this kind of direct communication.

```
iptables -A INPUT -s 94.78.71.0/24 -p tcp --dport 22 -j ACCEP
iptables -A OUTPUT -d 94.78.71.0/24 -p tcp --sport 22 -j ACCE
```

Explanation

SSH use tcp port 22 to communicate.

Test

The test method is the same as the previous SSH test case.

C.7 Allow ICMP communications between firewall node and Talos nodes

We focus on eth0,1,2 which represents DMZ and the internal subnet of Talos. We configure with the command **-p icmp**.

```
iptables -A INPUT -s 94.78.0.0/16 -p icmp -j ACCEPT
iptables -A OUTPUT -d 94.78.0.0/16 -p icmp -j ACCEPT
```

Explanation

Filtering by icmp protocal.

Test

We can ping from different subnets.

Due to the static routing strategy, I found the path may not be as expected, so I removed the interface part of the rules, only tracking the source and destination of Talos IP addresses, and then it worked better.