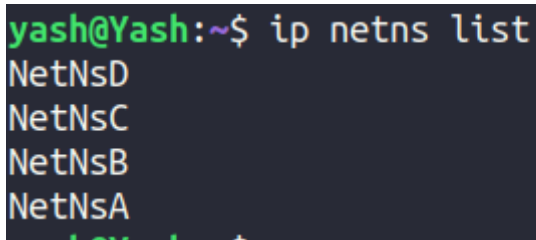# CS 549: Performance Analysis of Computer Networks

Lab Assignment 3

Submitted By : Yash Gupta , B21147

Creating four namespaces , NetNsA ...NetNsB

Different IP Addresses have been assigned to each interface on each namespace.
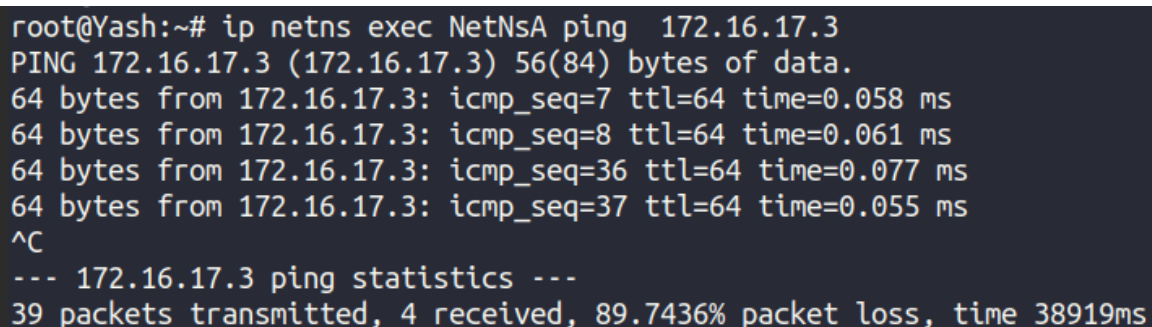
NetNsA – 172.16.17.1
NetNsB – 172.16.17.2
NetNsC – 172.16.17.3
NetNsD – 172.16.17.4

(a) Run ping between NetNsA and NetNsB. Observe the traffic using Wireshark.

Observations :- Initially I gave the IP addresses in the same range as my host machine i.e. 172.16.17.x / 24. The packets loss was around 90 % (see the attached scrresnhot below).
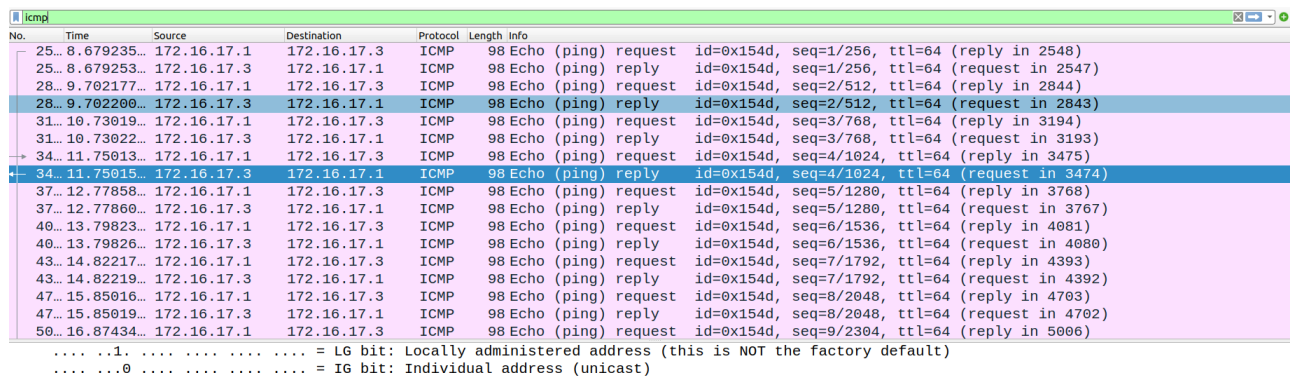

Later on when I changed the ip range to 192.0.2.x/24 for every namespace interface – the packets loss was 0 %.
(attached screenshot below)

```
root@Yash:~# ip netns exec NetNsA ping 192.0.2.2
PING 192.0.2.2 (192.0.2.2) 56(84) bytes of data.
64 bytes from 192.0.2.2: icmp_seq=1 ttl=64 time=0.081 ms
64 bytes from 192.0.2.2: icmp_seq=2 ttl=64 time=0.059 ms
64 bytes from 192.0.2.2: icmp_seq=3 ttl=64 time=0.056 ms
64 bytes from 192.0.2.2: icmp_seq=4 ttl=64 time=0.120 ms
64 bytes from 192.0.2.2: icmp_seq=5 ttl=64 time=0.049 ms
64 bytes from 192.0.2.2: icmp_seq=6 ttl=64 time=0.049 ms
64 bytes from 192.0.2.2: icmp_seq=7 ttl=64 time=0.051 ms
64 bytes from 192.0.2.2: icmp_seq=8 ttl=64 time=0.056 ms
64 bytes from 192.0.2.2: icmp_seq=9 ttl=64 time=0.041 ms
64 bytes from 192.0.2.2: icmp_seq=10 ttl=64 time=0.049 ms
64 bytes from 192.0.2.2: icmp_seq=11 ttl=64 time=0.053 ms
64 bytes from 192.0.2.2: icmp_seq=12 ttl=64 time=0.045 ms
64 bytes from 192.0.2.2: icmp_seq=13 ttl=64 time=0.073 ms
^C
--- 192.0.2.2 ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 12272ms
rtt min/avg/max/mdev = 0.041/0.060/0.120/0.020 ms
```

The network traffic visualized on wireshark is also attached :

Every request and response is captured in wireshark for applied *ICMP* filter.

(b) Adding queue discipline with 20 % loss

There are 3 possibilities based on where we are adding the policy of 20% loss
1. On the sender side
2. on the receiver side
3. On both sides

Conisdering the first option as most appropriate in accordance with the real world sceenario , where packets loss occcurs because of congestion in the network, Applying queuing discpline to sender side (interface macvlanA in namespace NetNsA).

```
root@Yash:~# sudo ip netns exec NetNsA tc qdisc add dev macvlanA root netem loss 20%
root@Yash:~#
root@Yash:~# sudo ip netns exec NetNsA tc qdisc show dev macvlanA
qdisc netem 8001: root refcnt 2 limit 1000 loss 20%
```

Now pinging 192.0.2.2 from this interface

```
root@Yash:~# grc ip netns exec NetNsA ping 192.0.2.2
PING 192.0.2.2 (192.0.2.2) 56(84) bytes of data.
64 bytes from 192.0.2.2: icmp_seq=1 ttl=64 time=0.037 ms
64 bytes from 192.0.2.2: icmp_seq=2 ttl=64 time=0.034 ms
64 bytes from 192.0.2.2: icmp_seq=3 ttl=64 time=0.039 ms
64 bytes from 192.0.2.2: icmp_seq=4 ttl=64 time=0.051 ms
64 bytes from 192.0.2.2: icmp_seq=5 ttl=64 time=0.037 ms
64 bytes from 192.0.2.2: icmp_seq=6 ttl=64 time=0.042 ms
64 bytes from 192.0.2.2: icmp_seq=7 ttl=64 time=0.066 ms
64 bytes from 192.0.2.2: icmp_seq=10 ttl=64 time=0.045 ms
64 bytes from 192.0.2.2: icmp_seq=12 ttl=64 time=0.062 ms
64 bytes from 192.0.2.2: icmp_seq=13 ttl=64 time=0.048 ms
64 bytes from 192.0.2.2: icmp_seq=14 ttl=64 time=0.044 ms
^C
--- 192.0.2.2 ping statistics ---
14 packets transmitted, 11 received, 21.4286% packet loss, time 13299ms
rtt min/avg/max/mdev = 0.034/0.045/0.066/0.009 ms
```

packets with seq no - 8 , 9 , 11 are lost. (loss fraction = 3 / 14 = 0.21)

```
23… 11.00720… 192.0.2.1    192.0.2.2    ICMP    98 Echo (ping) request  id=0x0f60, seq=6/1536, ttl=64 (reply in 2332)
23… 11.00722… 192.0.2.2    192.0.2.1    ICMP    98 Echo (ping) reply    id=0x0f60, seq=6/1536, ttl=64 (request in 2331)
25… 12.03126… 192.0.2.1    192.0.2.2    ICMP    98 Echo (ping) request  id=0x0f60, seq=7/1792, ttl=64 (reply in 2517)
25… 12.03128… 192.0.2.2    192.0.2.1    ICMP    98 Echo (ping) reply    id=0x0f60, seq=7/1792, ttl=64 (request in 2516)
30… 15.10320… 192.0.2.1    192.0.2.2    ICMP    98 Echo (ping) request  id=0x0f60, seq=10/2560, ttl=64 (reply in 3054)
30… 15.10322… 192.0.2.2    192.0.2.1    ICMP    98 Echo (ping) reply    id=0x0f60, seq=10/2560, ttl=64 (request in 3053)
34… 17.15125… 192.0.2.1    192.0.2.2    ICMP    98 Echo (ping) request  id=0x0f60, seq=12/3072, ttl=64 (reply in 3427)
```

This is also captured in wireshark.
As packets with sequence number 8 , 9 , 11
are not captured by wireshark.

Hence the applied fixed loss of 20% in
queuing discipline of sender is perfeclty
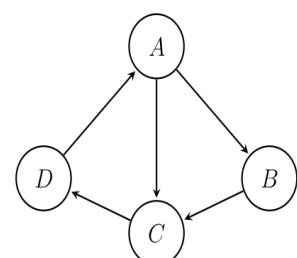reflected in the experiment results.

```
…0,  seq=6/1536,  ttl
…0,  seq=7/1792,  ttl
…0,  seq=7/1792,  ttl
…0,  seq=10/2560,  tt
…0,  seq=10/2560,  tt
…0,  seq=12/3072,  tt
```

Part -2

From the requirements of the downstream tasks, it is
very clear that :

1. For each outgoing link from every node , there has to be a interface corresponding to it.
2. There should be atleast 1 receiving interface for each node. (I am making only 1)

Based on this , The interfaces corresponsing to each namespace with their IP addresses are:

```
root@Yash:~# ip netns exec NetNsA ifconfig AB 192.0.2.1/24
root@Yash:~# ip netns exec NetNsA ifconfig AC 192.0.2.2/24
root@Yash:~# ip netns exec NetNsA ifconfig Ai 192.0.2.3/24
root@Yash:~#
root@Yash:~# ip netns exec NetNsB ifconfig BC 192.0.2.4/24
root@Yash:~# ip netns exec NetNsB ifconfig Bi 192.0.2.5/24
root@Yash:~#
root@Yash:~# ip netns exec NetNsC ifconfig CD 192.0.2.6/24
root@Yash:~# ip netns exec NetNsC ifconfig ci 192.0.2.7/24
root@Yash:~#
root@Yash:~# ip netns exec NetNsD ifconfig DA 192.0.2.8/24
root@Yash:~# ip netns exec NetNsD ifconfig Di 192.0.2.9/24
root@Yash:~#
```

Here AB is the interface corresponding to Namespace of node A and outgoing link to B.

Similary Ai , Bi , etc are the interfaces for all the incoming links to the node.

I am facing some logical errors hence can't proceed further. I thought that creating different interfaces for each link will help to deactivate a particular link by just making 100 % loss on corresponding interface. But doing so is blocking whole namespace/ node.

Hence submitting the partial assignment for now.