

Отчёт по лабораторной работе №5

НПИбд-02-22

Чесноков Артемий Павлович

Содержание

1 Цель работы	5
2 Выполнение лабораторной работы	6
3 Выводы	16
Список литературы	17

Список иллюстраций

2.1	по ssh	6
2.2	пингуем и смотрим айпишники	6
2.3	loss 10%	7
2.4	7%	7
2.5	видим потери по sequence number	7
2.6	19%	8
2.7	больше нет потерь!	8
2.8	второй аргумент после процента потерь	9
2.9	68%	9
2.10	смотрим на количество ретраев	9
2.11	смотрим на количество ретраев	10
2.12	нарушаем sequence number	10
2.13	сразу в пинге видим DUP!	10
2.14	для простых потерь	11
2.15	Пропущены некоторые запросы (вторая колонка)	11
2.16	скопируем первый эксперимент	12
2.17	50%	12
2.18	смотрим так же вторую колонку	13
2.19	результаты кидаем в то же место	13
2.20	сложновато	14
2.21	ура	15

Список таблиц

1 Цель работы

Основной целью работы является получение навыков проведения интерактивных экспериментов в среде Mininet по исследованию параметров сети, связанных с потерей, дублированием, изменением порядка и повреждением пакетов при передаче данных. Эти параметры влияют на производительность протоколов и сетей.

2 Выполнение лабораторной работы

Подключаемся к виртуалке, предварительно её включив (рис. 2.1).

```
→ User ssh -Y mininet@192.168.56.101
mininet@192.168.56.101's password:
Warning: No xauth data; using fake authentication data for X11 forwarding.
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-42-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

Last login: Wed Oct 15 05:32:28 2025 from 192.168.56.1
mininet@mininet-vm:~$ ls
Desktop  Downloads  mininet.orig  oflops  openflow  pox  Templates  work
Documents  mininet  Music  oftest  Pictures  Public  Videos
mininet@mininet-vm:~$ cd work/
mininet@mininet-vm:~/work$ la
demo.mn  lab_iperf3  lab_netem_i
mininet@mininet-vm:~/work$ cd lab_netem_i/
mininet@mininet-vm:~/work/lab_netem_i$ ls
change-delay  correlation-delay  distribution-delay  jitter-delay  simple-delay
mininet@mininet-vm:~/work/lab_netem_i$ cd ..
mininet@mininet-vm:~/work$ echo $DISPLAY
localhost:10.0
mininet@mininet-vm:~/work$ sudo mn --topo=single,2 -x|
```

Рис. 2.1: по ssh

Проверяем соединение (рис. 2.2).

```
root@mininet-vm:/home/mininet# ifconfig
h2-eth0: flags=4163UP,BROADCAST,RUNNING,MULTICAST mtu 1500
    inet 10.0.0.2 netmask 255.0.0.0 broadcast 10.255.255.255
        ether d2:9b:86:12:72:71 txqueuelen 1000 (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73UP,LOOPBACK,RUNNING mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        loop txqueuelen 1000 (Local Loopback)
        RX packets 1793 bytes 734572 (734.5 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 1793 bytes 734572 (734.5 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
root@mininet-vm:/home/mininet# ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=3.93 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.604 ms
```

Рис. 2.2: пингуем и смотрим айпишники

Пингуем с потерями (рис. 2.3).

```

root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem loss 10%
root@mininet-vm:/home/mininet# ping -c 100 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.31 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=2.02 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.794 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.086 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.063 ms

```

Рис. 2.3: loss 10%

Наблюдаем те самые потери (рис. 2.4).

```

64 bytes from 10.0.0.2: icmp_seq=79 ttl=64 time=0.107 ms
64 bytes from 10.0.0.2: icmp_seq=80 ttl=64 time=0.119 ms
64 bytes from 10.0.0.2: icmp_seq=81 ttl=64 time=0.102 ms
64 bytes from 10.0.0.2: icmp_seq=82 ttl=64 time=0.064 ms
64 bytes from 10.0.0.2: icmp_seq=83 ttl=64 time=0.060 ms
64 bytes from 10.0.0.2: icmp_seq=84 ttl=64 time=0.164 ms
64 bytes from 10.0.0.2: icmp_seq=86 ttl=64 time=0.125 ms
64 bytes from 10.0.0.2: icmp_seq=87 ttl=64 time=0.069 ms
64 bytes from 10.0.0.2: icmp_seq=88 ttl=64 time=0.066 ms
64 bytes from 10.0.0.2: icmp_seq=89 ttl=64 time=0.060 ms
64 bytes from 10.0.0.2: icmp_seq=90 ttl=64 time=0.068 ms
64 bytes from 10.0.0.2: icmp_seq=92 ttl=64 time=0.086 ms
64 bytes from 10.0.0.2: icmp_seq=93 ttl=64 time=0.130 ms
64 bytes from 10.0.0.2: icmp_seq=94 ttl=64 time=0.128 ms
64 bytes from 10.0.0.2: icmp_seq=95 ttl=64 time=0.187 ms
64 bytes from 10.0.0.2: icmp_seq=97 ttl=64 time=0.072 ms
64 bytes from 10.0.0.2: icmp_seq=98 ttl=64 time=0.066 ms
64 bytes from 10.0.0.2: icmp_seq=99 ttl=64 time=0.096 ms
64 bytes from 10.0.0.2: icmp_seq=100 ttl=64 time=0.087 ms

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 93 received, 7% packet loss, time 101332ms
rtt min/avg/max/mdev = 0.053/0.120/2.018/0.247 ms
root@mininet-vm:/home/mininet# []

```

Рис. 2.4: 7%

Добавим так же потери на стороне получателя пинга (рис. 2.5).

```

root@mininet-vm:/home/mininet# sudo tc qdisc add dev h2-eth0 root netem loss 10%
root@mininet-vm:/home/mininet# ping -c 100 10.0.0.2
64 bytes from 10.0.0.2: icmp_seq=93 ttl=64 time=0.130 ms
64 bytes from 10.0.0.2: icmp_seq=94 ttl=64 time=0.130 ms
64 bytes from 10.0.0.2: icmp_seq=95 ttl=64 time=0.187 ms
64 bytes from 10.0.0.2: icmp_seq=97 ttl=64 time=0.072 ms
64 bytes from 10.0.0.2: icmp_seq=98 ttl=64 time=0.066 ms
64 bytes from 10.0.0.2: icmp_seq=99 ttl=64 time=0.096 ms
64 bytes from 10.0.0.2: icmp_seq=100 ttl=64 time=0.087 ms

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 93 received, 7% packet loss, time 101332ms
rtt min/avg/max/mdev = 0.053/0.120/2.018/0.247 ms
root@mininet-vm:/home/mininet# ping -c 100 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=1.88 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.74 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.772 ms
3 64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.056 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.059 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.072 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.057 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.059 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.131 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.071 ms

```

Рис. 2.5: видим потери по sequence number

Наблюдаем потери (рис. 2.6).

```
64 bytes from 10.0.0.2: icmp_seq=75 ttl=64 time=0.074 ms
64 bytes from 10.0.0.2: icmp_seq=76 ttl=64 time=0.072 ms
64 bytes from 10.0.0.2: icmp_seq=77 ttl=64 time=0.075 ms
64 bytes from 10.0.0.2: icmp_seq=78 ttl=64 time=0.058 ms
64 bytes from 10.0.0.2: icmp_seq=80 ttl=64 time=0.074 ms
64 bytes from 10.0.0.2: icmp_seq=81 ttl=64 time=0.069 ms
64 bytes from 10.0.0.2: icmp_seq=83 ttl=64 time=0.117 ms
64 bytes from 10.0.0.2: icmp_seq=85 ttl=64 time=0.073 ms
64 bytes from 10.0.0.2: icmp_seq=86 ttl=64 time=0.062 ms
64 bytes from 10.0.0.2: icmp_seq=87 ttl=64 time=0.102 ms
64 bytes from 10.0.0.2: icmp_seq=88 ttl=64 time=0.068 ms
64 bytes from 10.0.0.2: icmp_seq=89 ttl=64 time=0.063 ms
64 bytes from 10.0.0.2: icmp_seq=90 ttl=64 time=0.068 ms
64 bytes from 10.0.0.2: icmp_seq=92 ttl=64 time=0.057 ms
64 bytes from 10.0.0.2: icmp_seq=93 ttl=64 time=0.054 ms
64 bytes from 10.0.0.2: icmp_seq=94 ttl=64 time=0.088 ms
64 bytes from 10.0.0.2: icmp_seq=96 ttl=64 time=0.075 ms
64 bytes from 10.0.0.2: icmp_seq=97 ttl=64 time=0.066 ms
64 bytes from 10.0.0.2: icmp_seq=100 ttl=64 time=0.061 ms

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 81 received, 19% packet loss, time 101616ms
rtt min/avg/max/mdev = 0.054/0.123/1.884/0.280 ms
root@mininet-vm:/home/mininet#
```

Рис. 2.6: 19%

Убираем задержки (рис. 2.7).

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h2-eth0 root netem loss 10%
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
Cannot find device "h1-eth0"
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h2-eth0 root netem
root@mininet-vm:/home/mininet# [root@mininet-vm ~]#
```

```
64 bytes from 10.0.0.2: icmp_seq=80 ttl=64 time=0.062 ms
64 bytes from 10.0.0.2: icmp_seq=81 ttl=64 time=0.062 ms
64 bytes from 10.0.0.2: icmp_seq=82 ttl=64 time=0.068 ms
64 bytes from 10.0.0.2: icmp_seq=83 ttl=64 time=0.063 ms
64 bytes from 10.0.0.2: icmp_seq=85 ttl=64 time=0.068 ms
64 bytes from 10.0.0.2: icmp_seq=86 ttl=64 time=0.066 ms
64 bytes from 10.0.0.2: icmp_seq=87 ttl=64 time=0.057 ms
64 bytes from 10.0.0.2: icmp_seq=88 ttl=64 time=0.054 ms
64 bytes from 10.0.0.2: icmp_seq=89 ttl=64 time=0.054 ms
64 bytes from 10.0.0.2: icmp_seq=90 ttl=64 time=0.054 ms
64 bytes from 10.0.0.2: icmp_seq=91 ttl=64 time=0.054 ms
64 bytes from 10.0.0.2: icmp_seq=92 ttl=64 time=0.054 ms
64 bytes from 10.0.0.2: icmp_seq=93 ttl=64 time=0.054 ms
64 bytes from 10.0.0.2: icmp_seq=94 ttl=64 time=0.054 ms
64 bytes from 10.0.0.2: icmp_seq=95 ttl=64 time=0.054 ms
64 bytes from 10.0.0.2: icmp_seq=96 ttl=64 time=0.054 ms
64 bytes from 10.0.0.2: icmp_seq=97 ttl=64 time=0.054 ms
64 bytes from 10.0.0.2: icmp_seq=98 ttl=64 time=0.054 ms
64 bytes from 10.0.0.2: icmp_seq=99 ttl=64 time=0.054 ms
64 bytes from 10.0.0.2: icmp_seq=100 ttl=64 time=0.061 ms

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 81 received, 19% packet loss, time 101616ms
rtt min/avg/max/mdev = 0.054/0.123/1.884/0.280 ms
root@mininet-vm:/home/mininet# tc qdisc add dev h1-eth0 root netem
root@mininet-vm:/home/mininet# ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.21 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.935 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.407 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.054 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.077 ms
```

Рис. 2.7: больше нет потерь!

Добавляем зависимость пакета на 50% от предыдущего (рис. 2.8).

```
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.067 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.094 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.118 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.062 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.063 ms
^C
-- 10.0.0.2 ping statistics --
15 packets transmitted, 15 received, 0% packet loss, time 14276ms
rtt min/avg/max/mdev = 0.054/0.236/1.212/0.343 ms
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem loss 50
% 50%
root@mininet-vm:/home/mininet# ping -c 50 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.508 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.064 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.055 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=0.157 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=0.057 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=0.120 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=0.063 ms
64 bytes from 10.0.0.2: icmp_seq=21 ttl=64 time=0.065 ms
64 bytes from 10.0.0.2: icmp_seq=22 ttl=64 time=0.051 ms
64 bytes from 10.0.0.2: icmp_seq=23 ttl=64 time=0.246 ms
```

Рис. 2.8: второй аргумент после процента потерь

Итоги - 68% потерь! Ужас! (рис. 2.9).

```
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.508 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.064 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.055 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=0.157 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=0.057 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=0.120 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=0.063 ms
64 bytes from 10.0.0.2: icmp_seq=21 ttl=64 time=0.065 ms
64 bytes from 10.0.0.2: icmp_seq=22 ttl=64 time=0.051 ms
64 bytes from 10.0.0.2: icmp_seq=23 ttl=64 time=0.246 ms
64 bytes from 10.0.0.2: icmp_seq=26 ttl=64 time=0.062 ms
64 bytes from 10.0.0.2: icmp_seq=27 ttl=64 time=0.064 ms
64 bytes from 10.0.0.2: icmp_seq=28 ttl=64 time=0.077 ms
64 bytes from 10.0.0.2: icmp_seq=30 ttl=64 time=0.071 ms
64 bytes from 10.0.0.2: icmp_seq=31 ttl=64 time=0.106 ms
64 bytes from 10.0.0.2: icmp_seq=50 ttl=64 time=0.442 ms

--- 10.0.0.2 ping statistics ---
50 packets transmitted, 16 received, 68% packet loss, time 50222ms
rtt min/avg/max/mdev = 0.051/0.138/0.508/0.136 ms
root@minitvm:~# sudo ip link del dev h1 eth0 root netns
```

Рис. 2.9: 68%

Для проверки повреждения - поднимаем iPerf (рис. 2.10).

```
warning: this system does not seem to support IPv6 - trying IPv4
[...]
Server listening on 5201
[...]
Accepted connection from 10.0.0.1, port 34899
[ 7] local 10.0.0.2 port 5201 connected to 10.0.0.1 port 34892
[ 7] Interval Transfer Bitrate
[ 7] 0.00-0.00 sec 2.18 GBytes 16.7 Gbits/sec
[ 7] 1.00-2.00 sec 2.18 GBytes 16.7 Gbits/sec
[ 7] 2.00-3.00 sec 2.18 GBytes 16.7 Gbits/sec
[ 7] 3.00-4.00 sec 2.19 GBytes 23.2 Gbits/sec
[ 7] 4.00-5.00 sec 2.19 GBytes 23.2 Gbits/sec
[ 7] 5.00-6.00 sec 2.79 GBytes 24.0 Gbits/sec
[ 7] 6.00-7.00 sec 2.59 GBytes 22.2 Gbits/sec
[ 7] 7.00-8.00 sec 2.64 GBytes 22.8 Gbits/sec
[ 7] 8.00-9.00 sec 2.64 GBytes 22.8 Gbits/sec
[ 7] 9.00-10.00 sec 2.48 GBytes 21.3 Gbits/sec
[ 10] Interval Transfer Bitrate
[ 7] 0.00-10.00 sec 25.5 GBytes 21.9 Gbits/sec
[...]
Server listening on 5201
[...]
iperf Done.
root@mininet-vm:/home/mininet#
```

Рис. 2.10: смотрим на количество ретраев

Убираем повреждения и наблюдаем 0 ретраев (рис. 2.11).

```

[7] 5.00-6.00 sec 2.79 GBytes 20.4 Gbits/sec
[7] 6.00-7.00 sec 2.59 GBytes 22.2 Gbits/sec
[7] 7.00-8.00 sec 2.54 GBytes 21.8 Gbits/sec
[7] 8.00-9.00 sec 2.54 GBytes 21.3 Gbits/sec
[7] 9.00-10.00 sec 2.48 GBytes 21.3 Gbits/sec

[10] Interval: 1.000000 sec Transfer: 25.5 GBytes Bitrate: 21.9 Gbits/sec
[10] 0.00-10.00 sec 25.5 GBytes 21.9 Gbits/sec receiver
[10] 0.00-10.00 sec 25.5 GBytes 21.9 Gbits/sec receiver

Server listening on 5201
Accepted connection from 10.0.0.1, port 34894
[7] local 10.0.0.2 port 5201 connected to 10.0.0.1 port 34896
[10] Interval: 1.000000 sec Transfer: 25.5 GBytes Bitrate: 21.9 Gbits/sec
[7] 0.00-1.00 sec 3.13 GBytes 26.0 Gbits/sec
[7] 1.00-2.00 sec 3.22 GBytes 27.7 Gbits/sec
[7] 2.00-3.00 sec 2.88 GBytes 24.0 Gbits/sec
[7] 3.00-4.00 sec 2.88 GBytes 23.4 Gbits/sec
[7] 4.00-5.00 sec 2.97 GBytes 25.5 Gbits/sec
[7] 5.00-6.00 sec 2.72 GBytes 23.4 Gbits/sec
[7] 6.00-7.00 sec 2.81 GBytes 24.2 Gbits/sec
[7] 7.00-8.00 sec 2.81 GBytes 23.4 Gbits/sec
[7] 8.00-9.00 sec 2.88 GBytes 24.7 Gbits/sec

[7] 7.00-8.00 sec 2.75 GBytes 21.9 Gbits/sec 6 1.10 MBbytes
[7] 8.00-9.00 sec 2.64 GBytes 22.7 Gbits/sec 8 9.96 KBbytes
[7] 9.00-10.00 sec 2.48 GBytes 21.3 Gbits/sec 2 1.14 MBbytes

[10] Interval: 1.000000 sec Transfer: 25.5 GBytes Bitrate: 21.9 Gbits/sec
[7] 0.00-10.00 sec 25.5 GBytes 21.9 Gbits/sec 333 sender
[7] 0.00-10.00 sec 25.5 GBytes 21.9 Gbits/sec receiver

iperf Done
root@mininet-wm:~/home/mininetsub$ tc qdisc add dev eth1 htb root netem root@mininet-wm:~/home/mininetsub$ iperf -c 10.0.0.2
Connecting to host 10.0.0.2, port 5201
[7] local 10.0.0.1 port 34896 connected to 10.0.0.2 port 5201
[10] Interval: 1.000000 sec Transfer: 25.5 GBytes Bitrate: 21.9 Gbits/sec
[7] 0.00-1.00 sec 3.13 GBytes 26.9 Gbits/sec 9 8.84 MBbytes
[7] 1.00-2.00 sec 3.22 GBytes 27.7 Gbits/sec 0 8.84 MBbytes
[7] 2.00-3.00 sec 2.88 GBytes 24.0 Gbits/sec 0 12.2 MBbytes
[7] 3.00-4.00 sec 2.88 GBytes 23.4 Gbits/sec 0 12.2 MBbytes
[7] 4.00-5.00 sec 2.97 GBytes 25.5 Gbits/sec 0 12.2 MBbytes
[7] 5.00-6.00 sec 2.72 GBytes 23.4 Gbits/sec 0 12.2 MBbytes
[7] 6.00-7.00 sec 2.81 GBytes 24.2 Gbits/sec 0 12.2 MBbytes
[7] 7.00-8.00 sec 2.81 GBytes 23.4 Gbits/sec 0 12.2 MBbytes
[7] 8.00-9.00 sec 2.88 GBytes 24.7 Gbits/sec 0 12.2 MBbytes

```

Рис. 2.11: смотрим на количество ретраев

Добавляем переупорядочивание пакетов (рис. 2.12).

```
[ 7] 8.00-9.00 sec 2.88 GBytes 24.7 Gbits/sec 0 12.2 MBytes
[ 7] 9.00-10.00 sec 2.92 GBytes 25.1 Gbits/sec 0 12.2 MBytes
[ ID] Interval Transfer Bitrate Retr
[ 7] 0.00-10.00 sec 29.3 GBytes 25.2 Gbits/sec 9 sender
[ 7] 0.00-10.00 sec 29.3 GBytes 25.2 Gbits/sec receiver

iperf Done.
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 1
0ms reorder 25% 50%
root@mininet-vm:/home/mininet# ping -c 20 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=12.1 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=12.5 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=11.3 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=11.3 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=10.4 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=10.7 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.065 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=10.6 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=10.8 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=10.8 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=13.6 ms
```

Рис. 2.12: нарушаем sequence number

Обновим правила, добавив дублирование (рис. 2.13).

```
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=0.055 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=0.057 ms

--- 10.0.0.2 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19174ms
rtt min/avg/max/mdev = 0.052/45.679/101.728/49.908 ms
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem duplicate 50%
root@mininet-vm:/home/mininet# ping -c 20 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.96 ms
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=2.07 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=1.56 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=1.58 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.202 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.307 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.072 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.072 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.095 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.110 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.061 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.062 ms (DUP!)
```

Рис. 2.13: сразу в пинге видим DUP!

Создаем директорию для воспроизводимого эксперимента (рис. 2.14).

```
*** Done
completed in 1874.939 seconds
mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_ii/simple_drop
mininet@mininet-vm:~$ cd ~/work/lab_netem_ii/simple_drop
mininet@mininet-vm:~/work/lab_netem_ii/simple_drop$ nano lab_netem_ii.py
mininet@mininet-vm:~/work/lab_netem_ii/simple_drop$ nano lab_netem_ii.py
mininet@mininet-vm:~/work/lab_netem_ii/simple_drop$ nano lab_netem_ii.py
mininet@mininet-vm:~/work/lab_netem_ii/simple_drop$ nano lab_netem_ii.py
mininet@mininet-vm:~/work/lab_netem_ii/simple_drop$ nano Makefile
mininet@mininet-vm:~/work/lab_netem_ii/simple_drop$ make
sudo python lab_netem_ii.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
*** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
```

Рис. 2.14: для простых потерь

Результаты запуска (рис. 2.15).

```
lab_netem_ii.py  lab_netem_ii_test.py  Makefile  ping.dat
mininet@mininet-vm:~/work/lab_netem_ii/simple_drop$ cat ping.dat
1.54 2
0.612 4
0.090 6
0.085 7
0.111 8
0.060 9
0.076 12
0.118 13
0.152 14
0.067 15
0.077 16
0.130 18
0.085 19
0.170 20
0.069 21
0.109 23
0.054 24
0.083 26
0.105 27
0.055 28
0.087 30
0.122 31
0.064 32
0.066 33
0.138 34
0.062 36
0.062 37
```

Рис. 2.15: Пропущены некоторые запросы (вторая колонка)

Подготовим инфраструктуру для остальных экспериментов (рис. 2.16).

```

mininet@mininet-vm:~/work/lab_netem_ii/simple_drop$ make clean
rm -f *.dat
mininet@mininet-vm:~/work/lab_netem_ii/simple_drop$ ls
lab_netem_ii.py lab_netem_ii_test.py Makefile
mininet@mininet-vm:~/work/lab_netem_ii/simple_drop$ cd ..
mininet@mininet-vm:~/work/lab_netem_ii$ cp -r simple_drop/* correlation-drop/
cp: target `correlation-drop/' is not a directory
mininet@mininet-vm:~/work/lab_netem_ii$ mkdir correlation-drop
mininet@mininet-vm:~/work/lab_netem_ii$ cp -r simple_drop/* correlation-drop/
mininet@mininet-vm:~/work/lab_netem_ii$ ls correlation-drop/
lab_netem_ii.py lab_netem_ii_test.py Makefile
mininet@mininet-vm:~/work/lab_netem_ii$ mkdir simple-corrupt
mininet@mininet-vm:~/work/lab_netem_ii$ cp -r simple_drop/* simple-corrupt/
mininet@mininet-vm:~/work/lab_netem_ii$ mkdir simple-reorder
mininet@mininet-vm:~/work/lab_netem_ii$ cp -r simple_drop/* simple-reorder/
mininet@mininet-vm:~/work/lab_netem_ii$ cd correlation-drop/
mininet@mininet-vm:~/work/lab_netem_ii/correlation-drop$ nano
lab_netem_ii.py lab_netem_ii_test.py Makefile
mininet@mininet-vm:~/work/lab_netem_ii/correlation-drop$ nano
lab_netem_ii.py lab_netem_ii_test.py Makefile
mininet@mininet-vm:~/work/lab_netem_ii/correlation-drop$ nano |

```

Рис. 2.16: скопируем первый эксперимент

Правим код, добавляя корреляцию (рис. 2.17).

```

GNU nano 4.8
lab_netem_ii.py
Modified.

net.start()

info( '*** Set delay\n')
h1.cmdPrint( 'tc qdisc add dev h1-eth0 root netem loss 20% 50%' )
h2.cmdPrint( 'tc qdisc add dev h2-eth0 root netem loss 20% 50%' )

time.sleep(10) # Wait 10 seconds

info( '*** Ping\n')
ping_output = h1.cmd( 'ping -c 100', h2.IP() )
info( '*** Ping output:\n' )
print(ping_output)
match = re.search(r'(\d+)% packet loss', ping_output)
if match:
    loss_percentage = match.group(1)
    info( '*** Packet loss: %s\n' % loss_percentage )

lines = ping_output.strip().split('\n')
ping_data = []

for line in lines:
    if "time=" in line:
        time_match = re.search(r'time=(\d+)', line)
        seq_match = re.search(r'icmp_seq=(\d+)', line)
        if time_match and seq_match:

```

Рис. 2.17: 50%

Наблюдаем результаты (рис. 2.18).

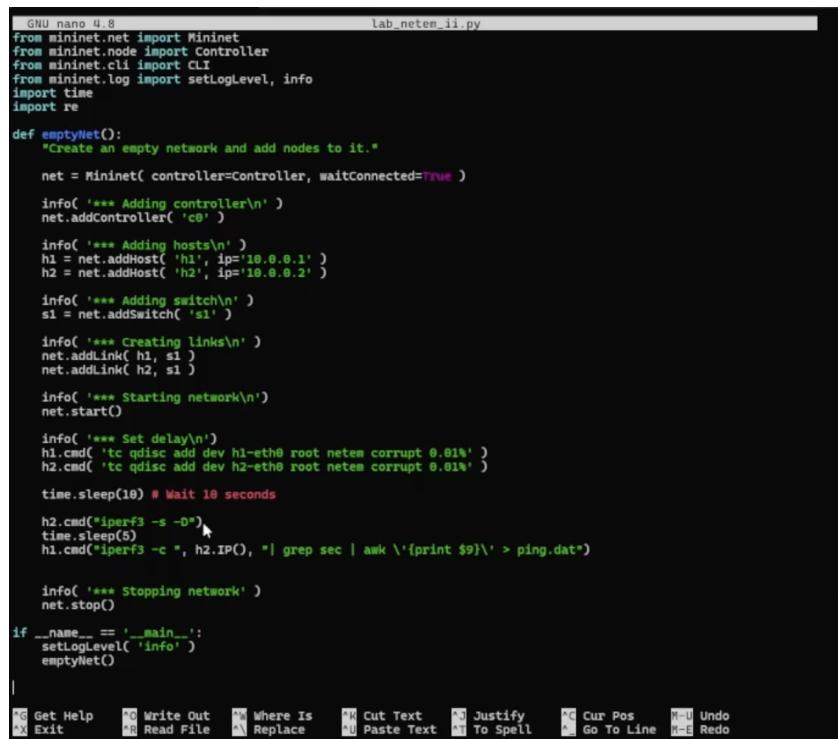
```

0.065 58
0.066 59
0.082 60
0.059 61
0.067 62
0.100 63
0.092 64
0.076 65
0.060 66
0.072 68
0.097 69
0.063 70
0.090 71
0.078 72
0.079 73
0.072 75
0.064 76
0.062 77
0.224 78
0.062 79
0.083 80
0.134 81
0.066 82
0.065 83
0.074 84
0.172 85
0.085 86
0.054 88
0.097 89
0.077 90
0.063 91
0.064 92
0.069 93
0.085 94
0.060 95
0.160 96

```

Рис. 2.18: смотрим так же вторую колонку

Чтобы проверить повреждения пакетов добавляем в логику кода - iperf (рис. 2.19).



```

GNU nano 4.8
lab.netem_ii.py

from mininet.net import Mininet
from mininet.node import Controller
from mininet.cli import CLI
from mininet.log import setLogLevel, info
import time
import re

def emptyNet():
    "Create an empty network and add nodes to it."
    net = Mininet( controller=Controller, waitConnected=True )

    info( '*** Adding controller\n' )
    net.addController( 'c0' )

    info( '*** Adding hosts\n' )
    h1 = net.addHost( 'h1', ip='10.0.0.1' )
    h2 = net.addHost( 'h2', ip='10.0.0.2' )

    info( '*** Adding switch\n' )
    s1 = net.addSwitch( 's1' )

    info( '*** Creating links\n' )
    net.addLink( h1, s1 )
    net.addLink( h2, s1 )

    info( '*** Starting network\n' )
    net.start()

    info( '*** Set delay\n' )
    h1.cmd( 'tc qdisc add dev h1-eth0 root netem corrupt 0.01%' )
    h2.cmd( 'tc qdisc add dev h2-eth0 root netem corrupt 0.01%' )

    time.sleep(10) # Wait 10 seconds

    h2.cmd("iperf3 -s -D")
    time.sleep(5)
    h1.cmd("iperf3 -c ", h2.IP(), "| grep sec | awk '{print \$9}' > ping.dat")

    info( '*** Stopping network' )
    net.stop()

if __name__ == '__main__':
    setLogLevel('info')
    emptyNet()

```

Рис. 2.19: результаты кидаем в то же место

Делаем код для переупорядочивания (рис. 2.20).

```
GNU nano 4.8                               lab_neten_ii.py                                Modified
info( '*** Adding hosts\n' )
h1 = net.addHost( 'h1', ip='10.0.0.1' )
h2 = net.addHost( 'h2', ip='10.0.0.2' )

info( '*** Adding switch\n' )
s1 = net.addSwitch( 's1' )

info( '*** Creating links\n' )
net.addLink( h1, s1 )
net.addLink( h2, s1 )

info( '*** Starting network\n' )
net.start()

info( '*** Set delay\n' )
h1.cmdPrint( 'tc qdisc add dev h1-eth0 root netem delay 100ms reorder 50% 60%' )
h2.cmdPrint( 'tc qdisc add dev h2-eth0 root netem delay 100ms reorder 50% 60%' )

time.sleep(10) # Wait 10 seconds

info( '*** Ping\n' )
ping_output = h1.cmd( 'ping -c 1000', h2.IP() )
info( '*** Ping output:\n' )
print(ping_output)
match = re.search(r'(\d+)% packet loss', ping_output)
if match:
    loss_percentage = match.group(1)
    info( '*** Packet loss: %s%%' % loss_percentage )

lines = ping_output.strip().split('\n')
ping_data = []

for line in lines:
    if "time=" in line:
        time_match = re.search(r'time=(\d+\.\d+)', line)
        seq_match = re.search(r'icmp_seq=(\d+)', line)
        if time_match and seq_match:
            time_r = time_match.group(1)
            seq_r = seq_match.group(1)
            ping_data.append("%s %s" % (time_r, seq_r))

with open('ping.dat', 'w') as f:
    for item in ping_data:
        f.write(item + '\n')

info( '*** Stopping network' )
```

Рис. 2.20: сложновато

Наблюдаем результат (рис. 2.21).

```
202 950
101 951
202 952
201 953
101 954
201 955
202 956
0.091 957
0.066 958
0.097 959
204 960
102 961
0.062 962
0.065 963
102 964
0.088 965
101 966
101 967
0.092 968
0.100 969
101 970
202 971
0.065 972
0.064 973
101 974
201 975
201 976
0.066 977
0.128 978
101 979
201 980
101 981
0.055 982
202 983
0.081 984
121 985
202 986
202 987
202 988
202 989
0.118 990
0.065 991
202 992
202 993
102 994
202 995
101 996
100 997
101 998
101 999
101 1000
mininet@mininet-vm:~/work/lab_netsim_ii/simple-reorder$ |
```

Рис. 2.21: ура

3 Выводы

В рамках выполнения лабораторной работы получили навыки проведения интерактивных экспериментов в среде Mininet по исследованию параметров сети, связанных с потерей, дублированием, изменением порядка и повреждением пакетов при передаче данных. Эти параметры влияют на производительность протоколов и сетей.

Список литературы