

Отчёт по лабораторной работе №5

Эмуляция и измерение потерь пакетов в глобальных сетях

Ким Реачна

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
2.0.1	Задание для самостоятельной работы	16
3	Листинги программы	24
3.0.1	Скрипт lab_netem_ii.py	24
3.0.2	Скрипт lab_netem_ii.py для добавление значения корреляции correlation-drop	26
3.0.3	Скрипт lab_netem_ii.py для добавление повреждения пакетов corruption-drop:	28
3.0.4	Скрипт lab_netem_ii.py для добавление переупорядочивания reorder-drop:	30
3.0.5	Скрипт lab_netem_ii.py для добавление дублирования пакетов:	32
3.0.6	Makefile	33
3.0.7	Скрипт ping.py для вывода информации о потерянных пакетах	34
4	Вывод	37

Список иллюстраций

2.1	Права запуска X-соединения	5
2.2	Информация о сетевом интерфейсе и IP-адресе h1	6
2.3	Информация о сетевом интерфейсе и IP-адресе h2	6
2.4	Проверка подключение от h1 к h2	6
2.5	Проверка подключение от h2 к h1	7
2.6	Добавление 10% потери пакетов на хосте h1	7
2.7	Проверка подключение	7
2.8	Добавление 10% потери пакетов на хосте h2	8
2.9	Восстановила конфигурацию	9
2.10	Добавление значения корреляции	10
2.11	Добавление повреждения пакетов	11
2.12	Добавление переупорядочивания пакетов	12
2.13	Добавление дублирования пакетов	12
2.14	Создание каталогов	13
2.15	Скрипт lab_netem_ii.py	13
2.16	ping.py	14
2.17	Makefile	14
2.18	Выполните эксперимент	15
2.19	График	15
2.20	Скрипт lab_netem_ii.py для добавление значения корреляции . . .	16
2.21	Выполнение эксперимент	17
2.22	График	17
2.23	Скрипт lab_netem_ii.py для добавление повреждения пакетов . . .	18
2.24	Makefile	18
2.25	Выполнение эксперимент	19
2.26	Повторная передача	19
2.27	Скрипт lab_netem_ii.py для добавление переупорядочивания пакетов	20
2.28	Выполнение эксперимент	20
2.29	График	21
2.30	Скрипт lab_netem_ii.py для добавление дублирования пакетов . .	22
2.31	Выполнение эксперимент	22
2.32	График	23

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

1. Запустила виртуальную среду с mininet и исправила права запуска X-соединения.
2. Настройка простейшей топологии и проверка подключение между хостами h1 и h2

```
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 9b973b364964296ca26a889d70808266
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth add mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 9b973b364964296ca26a889d70808266
root@mininet-vm:~# logout
mininet@mininet-vm:~$ sudo mn --topo=single,2 -x
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Running terms on localhost:10.0
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
```

Рис. 2.1: Права запуска X-соединения

```

root@mininet-vm:/home/mininet# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.1 netmask 255.0.0.0 broadcast 10.255.255.255
    ether d6:3e:c1:ab:1e:61 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=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 1292 bytes 288836 (288.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1292 bytes 288836 (288.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Рис. 2.2: Информация о сетевом интерфейсе и IP-адресе h1

```

root@mininet-vm:/home/mininet# ifconfig
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.2 netmask 255.0.0.0 broadcast 10.255.255.255
    ether 4e:38:21:4a:26:81 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=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 979 bytes 267968 (267.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 979 bytes 267968 (267.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Рис. 2.3: Информация о сетевом интерфейсе и IP-адресе h2

```

root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 6
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=1.63 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.170 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.047 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.068 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5091ms
rtt min/avg/max/mdev = 0.038/0.332/1.631/0.582 ms
root@mininet-vm:/home/mininet#

```

Рис. 2.4: Проверка подключения от h1 к h2

Минимальное, среднее, максимальное и стандартное отклонение $min = 0.038$, $avg = 0.332$, $max = 1.631$, $mdev = 0.582$.

```
root@mininet-vm:/home/mininet# ping 10.0.0.1 -c 6
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=0.732 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.070 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.045 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.039 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.046 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.043 ms

--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5103ms
rtt min/avg/max/mdev = 0.039/0.162/0.732/0.254 ms
root@mininet-vm:/home/mininet#
```

Рис. 2.5: Проверка подключение от h2 к h1

Минимальное, среднее, максимальное и стандартное отклонение $min = 0.039$, $avg = 0.162$, $max = 0.732$, $mdev = 0.254$.

3. Добавление потери пакетов на интерфейс, подключённый к эмулируемой глобальной сети

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem loss 10%
root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 100
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=1.02 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.347 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.160 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.049 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.045 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.074 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.058 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.041 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.047 ms
```

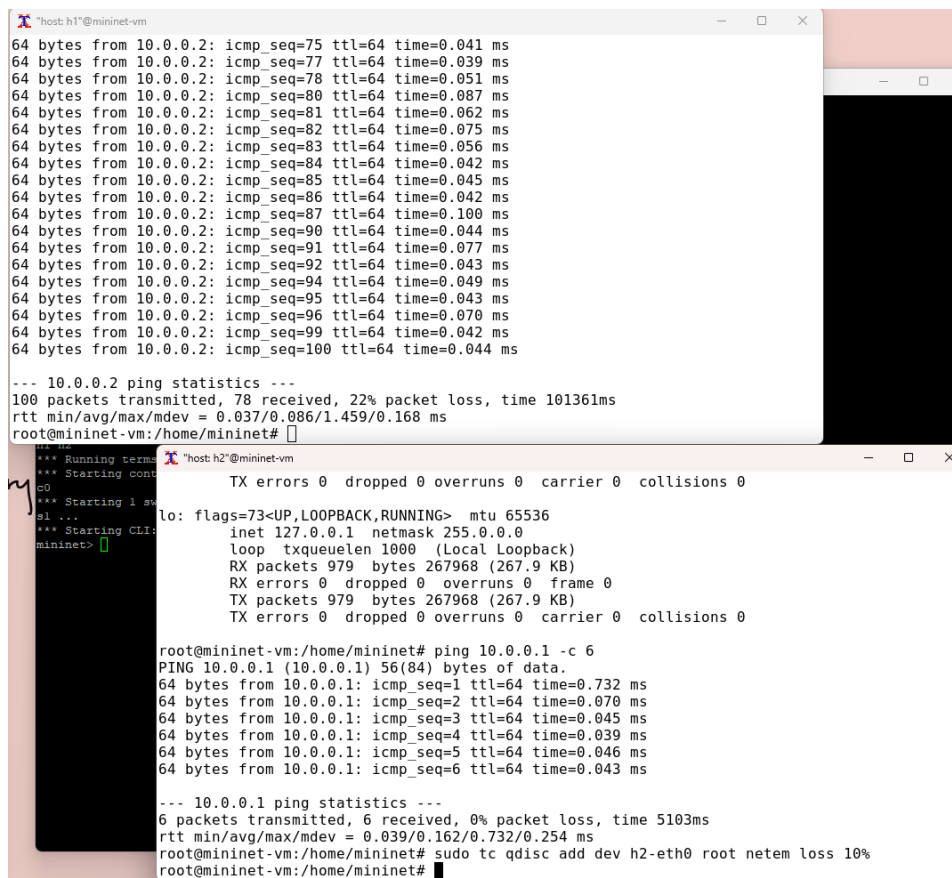
Рис. 2.6: Добавление 10% потери пакетов на хосте h1

```
64 bytes from 10.0.0.2: icmp_seq=94 ttl=64 time=0.104 ms
64 bytes from 10.0.0.2: icmp_seq=96 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=97 ttl=64 time=0.053 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=0.073 ms
64 bytes from 10.0.0.2: icmp_seq=100 ttl=64 time=0.037 ms

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 94 received, 6% packet loss, time 101353ms
rtt min/avg/max/mdev = 0.033/0.071/1.016/0.106 ms
```

Рис. 2.7: Проверка подключение

4. Добавление 10% потери пакетов на хосте h2 и проверка подключение между хостами h1 и h2 используя команду ping с параметром -c 100 с хоста h1:



```
host h1@mininet-vm
64 bytes from 10.0.0.2: icmp_seq=75 ttl=64 time=0.041 ms
64 bytes from 10.0.0.2: icmp_seq=77 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=78 ttl=64 time=0.051 ms
64 bytes from 10.0.0.2: icmp_seq=80 ttl=64 time=0.087 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=0.075 ms
64 bytes from 10.0.0.2: icmp_seq=83 ttl=64 time=0.056 ms
64 bytes from 10.0.0.2: icmp_seq=84 ttl=64 time=0.042 ms
64 bytes from 10.0.0.2: icmp_seq=85 ttl=64 time=0.045 ms
64 bytes from 10.0.0.2: icmp_seq=86 ttl=64 time=0.042 ms
64 bytes from 10.0.0.2: icmp_seq=87 ttl=64 time=0.100 ms
64 bytes from 10.0.0.2: icmp_seq=90 ttl=64 time=0.044 ms
64 bytes from 10.0.0.2: icmp_seq=91 ttl=64 time=0.077 ms
64 bytes from 10.0.0.2: icmp_seq=92 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=94 ttl=64 time=0.049 ms
64 bytes from 10.0.0.2: icmp_seq=95 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=96 ttl=64 time=0.070 ms
64 bytes from 10.0.0.2: icmp_seq=99 ttl=64 time=0.042 ms
64 bytes from 10.0.0.2: icmp_seq=100 ttl=64 time=0.044 ms

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 78 received, 22% packet loss, time 101361ms
rtt min/avg/max/mdev = 0.037/0.086/1.459/0.168 ms
root@mininet-vm:/home/mininet#

*** Running terms
*** Starting con
so
*** Starting l s
sl ...
*** Starting CLI
mininet>

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
loop txqueuelen 1000 (Local Loopback)
RX packets 979 bytes 267968 (267.9 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 979 bytes 267968 (267.9 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet# ping 10.0.0.1 -c 6
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=0.732 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.070 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.045 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.039 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.046 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.043 ms

--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5103ms
rtt min/avg/max/mdev = 0.039/0.162/0.732/0.254 ms
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h2-eth0 root netem loss 10%
root@mininet-vm:/home/mininet#
```

Рис. 2.8: Добавление 10% потери пакетов на хосте h2

5. Восстановила конфигурацию по умолчанию, удалив все правила и проверила соединения:


```
host: h1@mininet-vm
rtt min/avg/max/mdev = 0.037/0.086/1.459/0.168 ms
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 100
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=0.919 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.441 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.207 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.044 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.047 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.081 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.059 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.091 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.039 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.044 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.044 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.039 ms
^C
--- 10.0.0.2 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 14318ms
rtt min/avg/max/mdev = 0.039/0.145/0.919/0.231 ms
root@mininet-vm:/home/mininet#

*** Running terms
*** Starting cont
c0
*** Starting l s:lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
sl ...
*** Starting CLI:
mininet>
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 979 bytes 267968 (267.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 979 bytes 267968 (267.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet# ping 10.0.0.1 -c 6
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=0.732 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.070 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.045 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.039 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.046 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.043 ms
--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5103ms
rtt min/avg/max/mdev = 0.039/0.162/0.732/0.254 ms
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 h2-eth0 root netem
root@mininet-vm:/home/mininet#
```

Рис. 2.9: Восстановила конфигурацию

- Добавила на интерфейсе узла h1 коэффициент потери пакетов 50% , и каждая последующая вероятность зависит на 50% от последней и проверка подключение между хостами h1 и h2 используя команду ping с параметром -с 50:

```

root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem loss 50% 50%
root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 50
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=0.788 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.656 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.156 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.071 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.049 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.055 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.065 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.055 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=0.117 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=0.054 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=0.093 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=0.041 ms
64 bytes from 10.0.0.2: icmp_seq=21 ttl=64 time=0.062 ms
64 bytes from 10.0.0.2: icmp_seq=24 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=25 ttl=64 time=0.071 ms
64 bytes from 10.0.0.2: icmp_seq=27 ttl=64 time=0.052 ms
64 bytes from 10.0.0.2: icmp_seq=30 ttl=64 time=0.047 ms
64 bytes from 10.0.0.2: icmp_seq=31 ttl=64 time=0.042 ms
64 bytes from 10.0.0.2: icmp_seq=32 ttl=64 time=0.050 ms
64 bytes from 10.0.0.2: icmp_seq=33 ttl=64 time=0.364 ms
64 bytes from 10.0.0.2: icmp_seq=35 ttl=64 time=0.045 ms
64 bytes from 10.0.0.2: icmp_seq=36 ttl=64 time=0.042 ms
64 bytes from 10.0.0.2: icmp_seq=39 ttl=64 time=0.055 ms
64 bytes from 10.0.0.2: icmp_seq=40 ttl=64 time=0.042 ms
64 bytes from 10.0.0.2: icmp_seq=41 ttl=64 time=0.134 ms
64 bytes from 10.0.0.2: icmp_seq=43 ttl=64 time=0.151 ms
64 bytes from 10.0.0.2: icmp_seq=44 ttl=64 time=0.042 ms
64 bytes from 10.0.0.2: icmp_seq=45 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=46 ttl=64 time=0.056 ms
64 bytes from 10.0.0.2: icmp_seq=48 ttl=64 time=0.085 ms
64 bytes from 10.0.0.2: icmp_seq=49 ttl=64 time=0.037 ms

--- 10.0.0.2 ping statistics ---
50 packets transmitted, 31 received, 38% packet loss, time 50138ms
rtt min/avg/max/mdev = 0.037/0.118/0.788/0.170 ms
root@mininet-vm:/home/mininet# █

```

Рис. 2.10: Добавление значения корреляции

7. Добавила повреждения пакетов в эмулируемой глобальной сети и проверка конфигурацию с помощью инструмента iPerf3:

```

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 corrupt 0.01%
root@mininet-vm:/home/mininet# iperf3 -c 10.0.0.2
Connecting to host 10.0.0.2, port 5201
[ 7] local 10.0.0.1 port 52860 connected to 10.0.0.2 port 5201
[ ID] Interval      Transfer    Bitrate    Retr  Cwnd
[ 7] 0.00-1.00    sec 2.18 GBytes 18.7 Gbits/sec 17 3.07 MBytes
[ 7] 1.00-2.00    sec 2.18 GBytes 18.7 Gbits/sec 20 2.15 MBytes
[ 7] 2.00-3.00    sec 2.10 GBytes 18.0 Gbits/sec 17 2.15 MBytes
[ 7] 3.00-4.00    sec 2.15 GBytes 18.5 Gbits/sec 11 1.50 MBytes
[ 7] 4.00-5.00    sec 2.20 GBytes 18.9 Gbits/sec 12 1.50 MBytes
[ 7] 5.00-6.00    sec 2.15 GBytes 18.5 Gbits/sec 19 1.05 MBytes
[ 7] 6.00-7.00    sec 2.05 GBytes 17.6 Gbits/sec 16 1.05 MBytes
[ 7] 7.00-8.00    sec 1.74 GBytes 14.9 Gbits/sec 18 754 KBytes
[ 7] 8.00-9.00    sec 1.73 GBytes 14.9 Gbits/sec 19 617 KBytes
[ 7] 9.00-10.00   sec 1.94 GBytes 16.7 Gbits/sec 10 754 KBytes
-----
[ ID] Interval      Transfer    Bitrate    Retr
[ 7] 0.00-10.00   sec 20.4 GBytes 17.6 Gbits/sec 159      sender
[ 7] 0.00-10.00   sec 20.4 GBytes 17.5 Gbits/sec      receiver

iperf Done.
root@mininet-vm:/home/mininet#
*** Starting connection from host h2 to mininet-vm
*** Starting server on mininet-vm
warning: this system does not seem to support IPv6 - trying IPv4
*** Starting CLI: mininet>
mininet> Server listening on 5201
-----
Accepted connection from 10.0.0.1, port 52858
[ 7] local 10.0.0.2 port 5201 connected to 10.0.0.1 port 52860
[ ID] Interval      Transfer    Bitrate
[ 7] 0.00-1.00    sec 2.16 GBytes 18.5 Gbits/sec
[ 7] 1.00-2.00    sec 2.18 GBytes 18.7 Gbits/sec
[ 7] 2.00-3.00    sec 2.10 GBytes 18.0 Gbits/sec
[ 7] 3.00-4.00    sec 2.15 GBytes 18.5 Gbits/sec
[ 7] 4.00-5.00    sec 2.20 GBytes 18.9 Gbits/sec
[ 7] 5.00-6.00    sec 2.16 GBytes 18.5 Gbits/sec
[ 7] 6.00-7.00    sec 2.05 GBytes 17.6 Gbits/sec
[ 7] 7.00-8.00    sec 1.74 GBytes 15.0 Gbits/sec
[ 7] 8.00-9.00    sec 1.73 GBytes 14.9 Gbits/sec
[ 7] 9.00-10.00   sec 1.94 GBytes 16.7 Gbits/sec
-----
[ ID] Interval      Transfer    Bitrate
[ 7] 0.00-10.00   sec 20.4 GBytes 17.5 Gbits/sec      receiver
Server listening on 5201

```

Рис. 2.11: Добавление повреждения пакетов

8. Добавила переупорядочивания пакетов в интерфейс подключения к эмулируемой глобальной сети проверка подключение между хостами h1 и h2 используя команду `ring` с параметром `-s 20`:

```

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 delay 10ms reorder 25% 50%
root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 20
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=11.8 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=11.1 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=10.2 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.059 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=10.7 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=10.1 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=10.8 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=10.7 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=10.7 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=10.8 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=10.1 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=10.8 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=10.7 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=10.5 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=10.5 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=10.1 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=10.5 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=10.7 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=10.7 ms

--- 10.0.0.2 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19041ms
rtt min/avg/max/mdev = 0.059/10.108/11.778/2.335 ms
root@mininet-vm:/home/mininet#

```

Рис. 2.12: Добавление переупорядочивания пакетов

9. Добавила дублирования пакетов в интерфейс подключения к эмулируемой глобальной сети

```

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 10.0.0.2 -c 20
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=0.839 ms
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.869 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.537 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.143 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.442 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.050 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.055 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.073 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.049 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.045 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.045 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.098 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.098 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.076 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.049 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.057 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.057 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.083 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.083 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=0.042 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=0.046 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=0.060 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=0.061 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=0.042 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=0.066 ms

--- 10.0.0.2 ping statistics ---
20 packets transmitted, 20 received, +7 duplicates, 0% packet loss, time 19441ms
rtt min/avg/max/mdev = 0.042/0.153/0.869/0.228 ms
root@mininet-vm:/home/mininet#

```

Рис. 2.13: Добавление дублирования пакетов

10. Создала каталог, в который будут размещаться файлы эксперимента:

```

mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_ii/simple-drop
mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_ii/correlation-drop
mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_ii/corruption-drop
mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_ii/reorder-drop
mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_ii/duplicate-drop
mininet@mininet-vm:~$ cd ~/work/lab_netem_ii
mininet@mininet-vm:~/work/lab_netem_ii$ ls
correlation-drop  corruption-drop  duplicate-drop  reorder-drop  simple-drop

```

Рис. 2.14: Создание каталогов

11. Добавление потери пакетов на интерфейс, подключённый к эмулируемой глобальной сети:

- Создала скрипт для эксперимента lab_netem_ii.py, Makefile, ping.py:

```

GNU nano 4.8 /home/mininet/work/lab_netem_ii/simple-drop/lab_netem_ii.py Modified
#!/usr/bin/env python

'''
Simple experiment.
Output: ping.dat
'''

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

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.cmdPrint( 'tc qdisc add dev h1-eth0 root netem loss 10%' )
    h2.cmdPrint( 'tc qdisc add dev h2-eth0 root netem loss 10%' )

    time.sleep(10) # Wait 10 seconds

    info( '*** Ping\n' )
    h1.cmdPrint( 'ping -c 100', h2.IP(), '| grep "time=" | awk \'{print $5, $7}\'} | sed -e \'/s/time=//g\' -e \'/icmp_seq=//g\' > ping.dat' )

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

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

```

Рис. 2.15: Скрипт lab_netem_ii.py

```
mc [mininet@mininet-vm]:~/work/lab_netem_i/simple-drop
GNU nano 4.8 /home/mininet/work/lab_netem_i/simple-drop/ping.py
def analyze_results(file_path='ping.dat', total_packets=100):
    with open(file_path, 'r') as f:
        lines = f.readlines()

        received_packets = set(int(line.split()[0]) for line in lines)
        lost_packets = set(range(1, total_packets + 1)) - received_packets

        lost_packets_count = len(lost_packets)
        loss_percentages = (lost_packets_count / total_packets) * 100

        print(f'Total packets: {total_packets}')
        print(f'Lost packets: {lost_packets_count}')
        print(f'Lost packet numbers: {sorted(list(lost_packets))}')
        print(f'Loss percentage: {loss_percentages:.2f}%')

if __name__ == '__main__':
    analyze_results()
```

Рис. 2.16: ping.py

```
GNU nano 4.8 /home/mininet/work/lab_netem_i/simple-drop/Makefile
all: ping.dat ping.png ping.check

ping.dat:
    sudo python lab_netem_i.py
    sudo chown mininet:mininet ping.dat

ping.png: ping.dat
    ./ping_plot

ping.check:
    sudo python ping.py > ping.txt
    cat ping.txt

clean:
    -rm -f *.dat *.txt
```

Рис. 2.17: Makefile

- Выполнила эксперимент:

```

mininet@mininet-vms:~/work/lab_netem_i1/simple-drop$ make
sudo python lab_netem_i1.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
s1
*** Set delay
*** h1 : ('tc qdisc add dev h1-eth0 root netem loss 10%,)
*** h2 : ('tc qdisc add dev h2-eth0 root netem loss 10%,)
*** Ping
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "time=" | awk '{print $5, $7}' | sed -e 's/time=//g' -e 's/icmp_seq=//g' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
sudo python ping.py > ping.txt
cat ping.txt
Total packets: 100
Lost packets: 23
Lost packet numbers: [3, 7, 13, 18, 23, 26, 31, 37, 48, 49, 52, 53, 59, 62, 63, 66, 68, 70, 85, 86, 88, 93, 95]
Loss percentage: 23.00%

```

Рис. 2.18: Выполните эксперимент

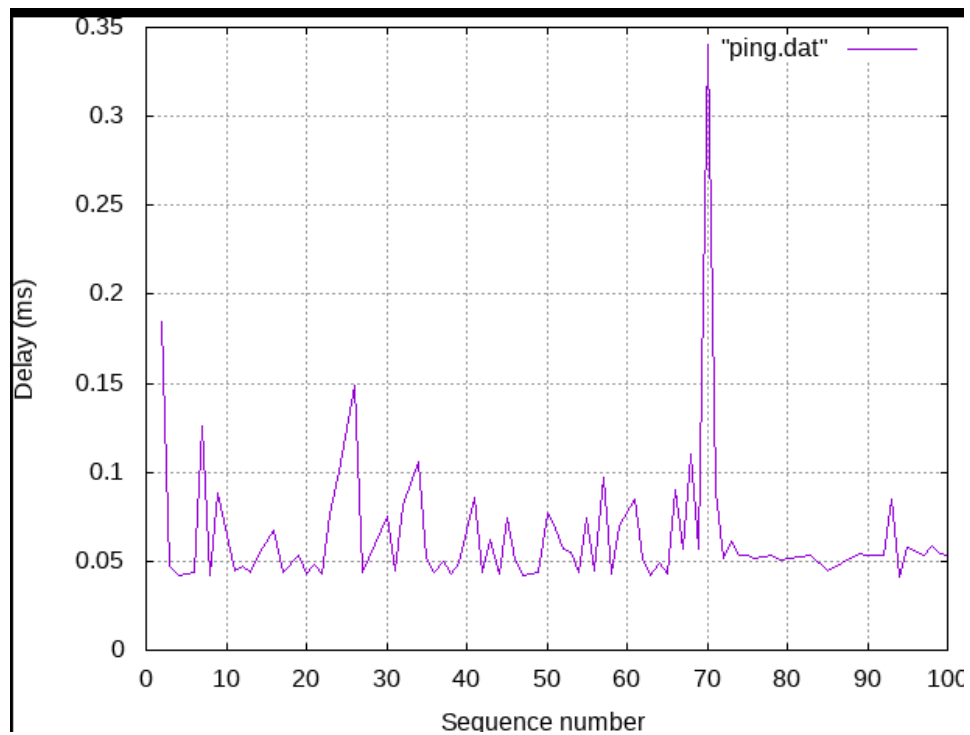


Рис. 2.19: График

2.0.1 Задание для самостоятельной работы

1. Реализовала воспроизводимые эксперименты по добавление значения корреляции:

```
GNU nano 4.8 /home/mininet/work/lab_netem_ii/correlation-drop/lab_netem_ii.py
#!/usr/bin/env python

"""
Simple experiment.
Output: ping.dat
"""

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

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.cmdPrint( 'tc qdisc add dev h1-eth0 root netem loss 50% 50%' )

    time.sleep(10) # Wait 10 seconds

    info( '*** Ping\n' )
    h1.cmdPrint( 'ping -c 100', h2.IP(), '| grep "time=" | awk \'{print $5, $7}\'} | sed -e \'/s/time=//g\' -e \'/icmp_seq=//g\' > ping.dat' )

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

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

Рис. 2.20: Скрипт lab_netem_ii.py для добавление значения корреляции


```

mininet@mininet-vm:~/work/lab_netem_ii/correlation-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
s1
*** Set delay
*** h1 : ('tc qdisc add dev h1-eth0 root netem loss 50% 50%,)
*** Ping
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "time=" | awk '{print $5, $7}' | sed -e 's/time=//g' -e 's/icmp_seq=//g' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
sudo python ping.py > ping.txt
cat ping.txt
Total packets: 100
Lost packets: 50
Lost packet numbers: [2, 3, 4, 5, 6, 7, 9, 10, 11, 14, 15, 18, 20, 22, 23, 24, 26, 27, 28, 30, 33, 35, 36, 37, 38, 39, 41, 43, 44, 46, 48, 52, 53,
60, 61, 66, 67, 68, 73, 75, 77, 78, 80, 81, 88, 89, 90, 91, 96, 100]
Loss percentage: 50.00%
Total packets: 100
Lost packets: 50
Lost packet numbers: [2, 3, 4, 5, 6, 7, 9, 10, 11, 14, 15, 18, 20, 22, 23, 24, 26, 27, 28, 30, 33, 35, 36, 37, 38, 39, 41, 43, 44, 46, 48, 52, 53,
60, 61, 66, 67, 68, 73, 75, 77, 78, 80, 81, 88, 89, 90, 91, 96, 100]
Duplicated packets: 0
Duplicated packet numbers: []
Loss percentage: 50.00%
mininet@mininet-vm:~/work/lab_netem_ii/correlation-drop$

```

Рис. 2.21: Выполнение эксперимент

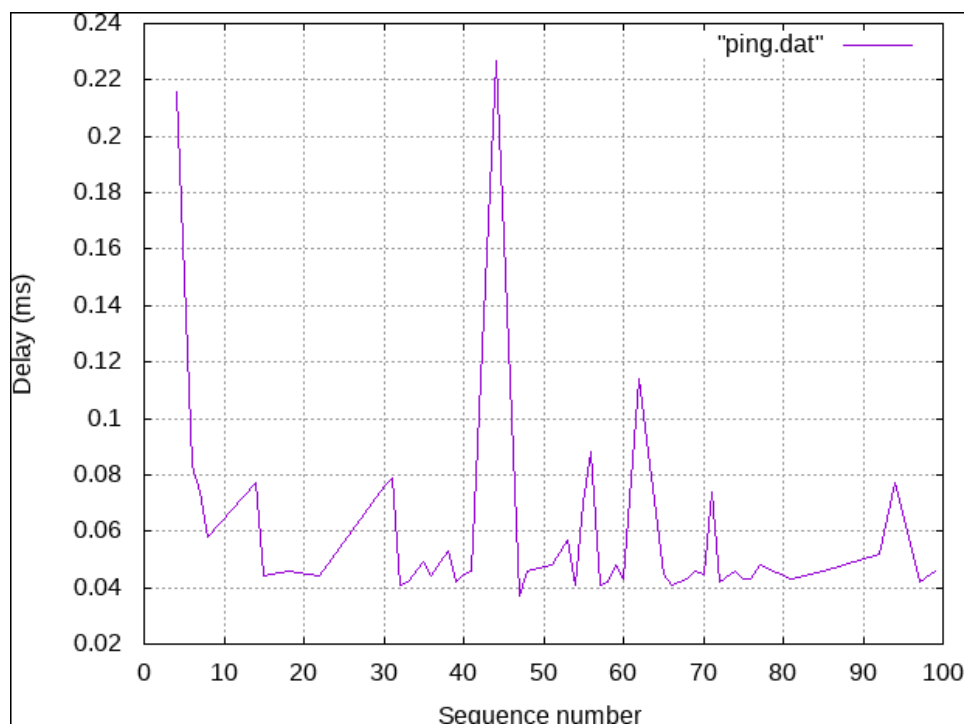


Рис. 2.22: График

2. Реализовала воспроизводимые эксперименты по добавление повреждения

пакетов:

```
GNU nano 4.8 /home/mininet/work/lab_netem_ii/corruption-drop/lab_netem_ii.py
#!/usr/bin/env python

'''
Simple experiment.
Output: ping.dat
'''

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

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.cmdPrint( 'tc qdisc add dev h1-eth0 root netem corrupt 0.01%' )

    info( '*** Traffic generation\n' )
    h2.cmdPrint( 'iperf3 -s -D -l' )
    time.sleep(10) # Wait 10 seconds for servers to start
    h1.cmdPrint( 'iperf3 -c', h2.IP(), '-J > iperf_result.json' )

    info( '*** Ping\n' )
    h1.cmdPrint( 'ping -c 100', h2.IP(), '| grep "time=" | awk '{print $5, $7}{' | sed -e 's/time=//q' -e 's/icmp_seq=//q' > ping.dat' )

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

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

Рис. 2.23: Скрипт lab_netem_ii.py для добавление повреждения пакетов

```
GNU nano 4.8 /home/mininet/work/lab_netem_ii/corruption-drop/Makefile
all: ping.dat ping.plot

ping.dat:
    sudo python lab_netem_ii.py

ping.png: ping.dat
    ./ping_plot

ping.plot: iperf_results.json
    plot_iperf.sh iperf_results.json

clean:
    -rm -f *.dat *.json *.csv
    -rm -rf results
```

Рис. 2.24: Makefile

```

mininet@mininet-vms:~/work/lab_netem_i1/corruption-drop$ make
sudo python lab_netem_i1.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
s1
*** Set delay
*** h1 : ('tc qdisc add dev h1-eth0 root netem corrupt 0.01%',)
*** Traffic generation
*** h2 : ('iperf3 -s -D -l',)
*** h1 : ('iperf3 -c', '10.0.0.2', '-J > iperf_result.json')
*** Ping
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "time=" | awk '{print $5, $7}' | sed -e 's/time=//g' -e 's/icmp_seq=//g' > ping.dat')

```

Рис. 2.25: Выполнение эксперимент

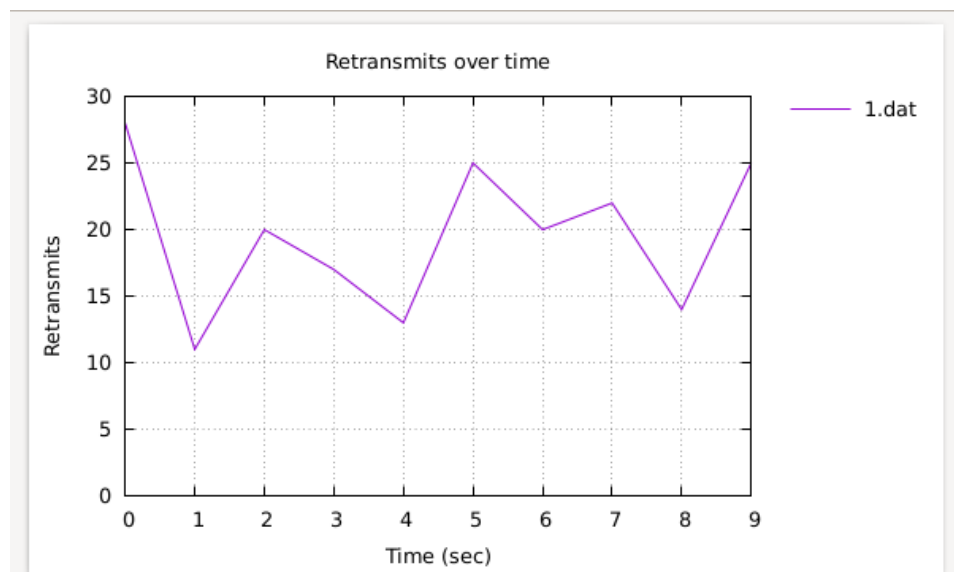


Рис. 2.26: Повторная передача

3. Реализовала воспроизводимые эксперименты по добавление переупорядочивания пакетов:

```

GNU nano 4.8 /home/mininet/work/lab_netem_ii/duplicate-drop/lab_netem_ii.py
#!/usr/bin/env python

"""
Simple experiment.
Output: ping.dat
"""

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

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.cmdPrint( 'tc qdisc add dev h1-eth0 root netem duplicate 50%' )

    time.sleep(10) # Wait 10 seconds

    info( '*** Ping\n' )
    h1.cmdPrint( 'ping -c 100', h2.IP(), '| grep "time=" | awk \'{print $5, $7}\'} | sed -e \'s/time=//g\' -e \'s/icmp_seq=//g\' > ping.dat' )

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

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

```

Рис. 2.27: Скрипт lab_netem_ii.py для добавление переупорядочивания пакетов

```

mininet@mininet-vm:~/work/lab_netem_ii/reorder-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
s1
*** Set delay
*** h1 : ('tc qdisc add dev h1-eth0 root netem delay 10ms reorder 25% 25%',)
*** Ping
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "time=" | awk \'{print $5, $7}\'} | sed -e \'s/time=//g\' -e \'s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
sudo python ping.py > ping.txt
cat ping.txt
Total packets: 100
Lost packets: 0
Lost packet numbers: []
Loss percentage: 0.00%
Total packets: 100
Lost packets: 0
Lost packet numbers: []
Duplicated packets: 0
Duplicated packet numbers: []
Loss percentage: 0.00%

```

Рис. 2.28: Выполнение эксперимент

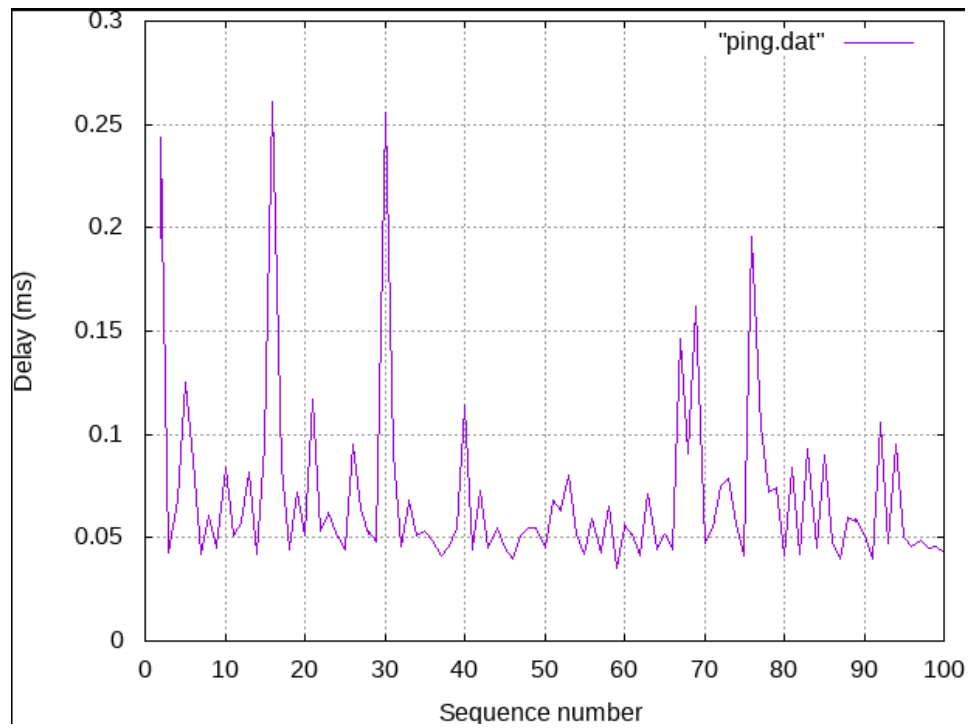


Рис. 2.29: График

4. Реализовала воспроизводимые эксперименты по добавление дублирования пакетов:

```

GNU nano 4.8 /home/mininet/work/lab_netem_i1/reorder-drop/lab_netem_i1.py
/usr/bin/env python

'''
Simple experiment.
Output: ping.dat
'''

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

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.cmdPrint( 'tc qdisc add dev h1-eth0 root netem delay 10ms reorder 25% 25%' )

    time.sleep(10) # Wait 10 seconds

    info( '*** Ping\n' )
    h1.cmdPrint( 'ping -c 100', h2.IP(), '| grep "time=" | awk \'{print $5, $7}\'} | sed -e \'s/time=//g\' -e \'s/icmp_seq=//g\' > ping.dat' )

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

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

```

Рис. 2.30: Скрипт lab_netem_i1.py для добавление дублирования пакетов

```

mininet@mininet-vm:~/work/lab_netem_i1/duplicate-drop$ make
sudo python lab_netem_i1.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
s1
*** Set delay
*** h1 : ('tc qdisc add dev h1-eth0 root netem duplicate 50%,)
*** Ping
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "time=" | awk \'{print $5, $7}\'} | sed -e \'s/time=//g\' -e \'s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
sudo python ping.py > ping.txt
cat ping.txt
Total packets: 100
Lost packets: 0
Lost packet numbers: []
Loss percentage: 0.00%
Total packets: 100
Lost packets: 0
Lost packet numbers: []
Duplicated packets: 50
Duplicated packet numbers: [2, 5, 9, 12, 14, 17, 18, 20, 22, 23, 24, 25, 27, 28, 29, 31, 33, 36, 40, 41, 43, 44, 50, 51, 52, 54, 56, 57, 60, 61, 63, 64, 65, 66, 69, 70, 72, 77, 78, 81, 84, 88, 89, 92, 93, 95, 96, 97, 98, 99]
Loss percentage: 0.00%

```

Рис. 2.31: Выполнение эксперимент

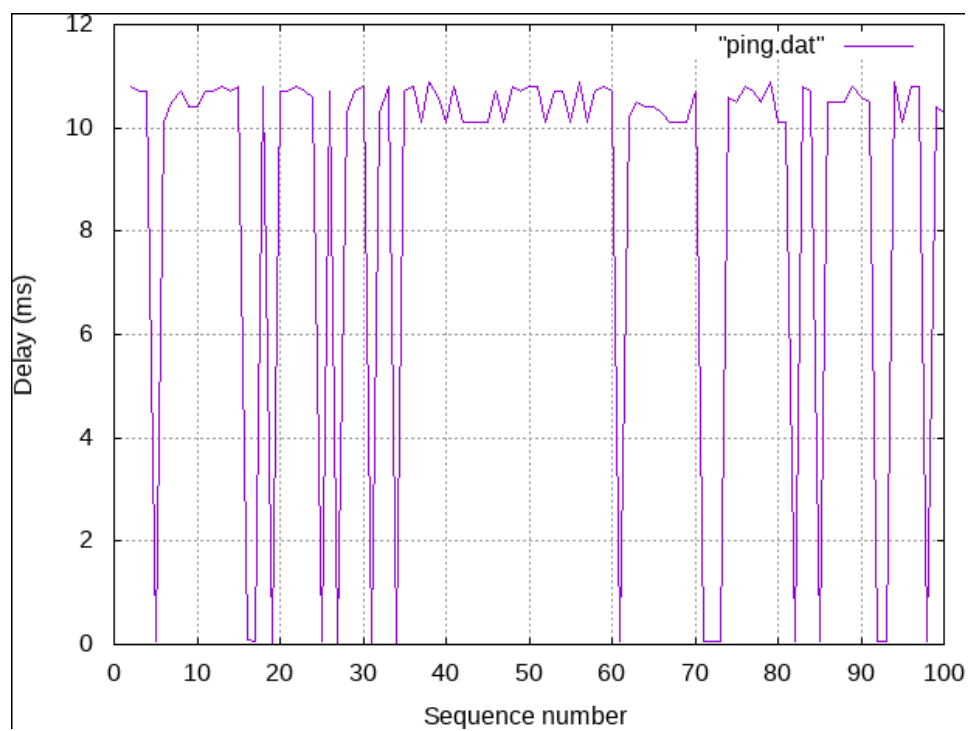


Рис. 2.32: График

3 Листинги программы

3.0.1 Скрипт lab_netem_ii.py

```
#!/usr/bin/env python

"""
Simple experiment.
Output: ping.dat
"""

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

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.cmdPrint( 'tc qdisc add dev h1-eth0 root netem loss 10%' )
h2.cmdPrint( 'tc qdisc add dev h2-eth0 root netem loss 10%' )

time.sleep(10) # Wait 10 seconds

info( '*** Ping\n' )
h1.cmdPrint( 'ping -c 100', h2.IP(), '| grep "time=" |
awk \'{print $5, $7}\'' | sed -e \'s/time=//g\'
-e \'s/icmp_seq=//g\' > ping.dat' )

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

```

```

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

```

3.0.2 Скрипт lab_netem_ii.py для добавление значения корреляции correlation-drop

```

#!/usr/bin/env python

"""
Simple experiment.
Output: ping.dat
"""

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

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.cmdPrint( 'tc qdisc add dev h1-eth0 root netem loss 50% 50%' )

time.sleep(10) # Wait 10 seconds

info( '*** Ping\n' )
h1.cmdPrint( 'ping -c 100', h2.IP(), '| grep "time=" |
awk \'{print $5, $7}\'' | sed -e \'s/time=//g\'
-e \'s/icmp_seq=//g\' > ping.dat' )

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

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

```

```
emptyNet()
```

3.0.3 Скрипт lab_netem_ii.py для добавление повреждения пакетов corruption-drop:

```
#!/usr/bin/env python

"""
Simple experiment.
Output: ping.dat
"""

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

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.cmdPrint( 'tc qdisc add dev h1-eth0 root netem corrupt 0.01%' )

info( '*** Traffic generation\n' )
h2.cmdPrint( 'iperf3 -s -D -1' )
time.sleep(10) # Wait 10 seconds for servers to start
h1.cmdPrint( 'iperf3 -c', h2.IP(), '-J > iperf_result.json' )

info( '*** Ping\n' )
h1.cmdPrint( 'ping -c 100', h2.IP(), '| grep "time=" |
awk \'{print $5, $7}\'' | sed -e \'s/time=//g\'
-e \'s/icmp_seq=//g\' > ping.dat' )

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

if __name__ == '__main__':

```

```
setLogLevel( 'info' )  
emptyNet()
```

3.0.4 Скрипт lab_netem_ii.py для добавление переупорядочивания reorder-drop:

```
#!/usr/bin/env python
```

```
"""
```

```
Simple experiment.
```

```
Output: ping.dat
```

```
"""
```

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

```
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.cmdPrint( 'tc qdisc add dev h1-eth0 root netem delay 10ms reorder 25%' )

time.sleep(10) # Wait 10 seconds

info( '*** Ping\n' )
h1.cmdPrint( 'ping -c 100', h2.IP(), '| grep "time=" |
awk \'{print $5, $7}\'' | sed -e \'s/time=//g\'
-e \'s/icmp_seq=//g\' > ping.dat' )

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

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

```

3.0.5 Скрипт lab_netem_ii.py для добавление дублирования пакетов:

```
#!/usr/bin/env python

"""
Simple experiment.
Output: ping.dat
"""

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

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.cmdPrint( 'tc qdisc add dev h1-eth0 root netem duplicate 50%' )

time.sleep(10) # Wait 10 seconds

info( '*** Ping\n')
h1.cmdPrint( 'ping -c 100', h2.IP(), '| grep "time=" |
awk \'{print $5, $7}\'' | sed -e \'s/time=//g\'
-e \'s/icmp_seq=//g\' > ping.dat'>

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

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

```

3.0.6 Makefile

```
all: ping.dat ping.png ping.check
```

```
ping.dat:
```

```
sudo python lab_netem_ii.py
sudo chown mininet:mininet ping.dat
```

```
ping.png: ping.dat
        ./ping_plot
```

```
ping.check:
        sudo python ping.py > ping.txt
        cat ping.txt
```

```
clean:
        -rm -f *.dat *.txt
```

3.0.7 Скрипт ping.py для вывода информации о потерянных пакетах

```
def analyze_results(file_path='ping.dat', total_packets=100):
    with open(file_path, 'r') as f:
        lines = f.readlines()

    received_packets = set(int(line.split()[0]) for line in lines)
    lost_packets = set(range(1, total_packets + 1)) - received_packets

    lost_packets_count = len(lost_packets)
    loss_percentages = (lost_packets_count / total_packets) * 100

    print(f'Total packets: {total_packets}')
    print(f'Lost packets: {lost_packets_count}')
    print(f'Lost packet numbers: {sorted(list(lost_packets))}')
    print(f'Loss percentage: {loss_percentages:.2f}%')
```

```

if __name__ == '__main__':
    analyze_results()

def analyze_results(file_path='ping.dat', total_packets=100):
    with open(file_path, 'r') as f:
        lines = f.readlines()

    received_packets = set()
    duplicated_packets = set()
    packet_numbers = []

    for line in lines:
        packet_number = int(line.split()[0])

        if packet_number in received_packets:
            duplicated_packets.add(packet_number)
        else:
            received_packets.add(packet_number)

        packet_numbers.append(packet_number)

    lost_packets = set(range(1, total_packets + 1)) - received_packets

    lost_packets_count = len(lost_packets)
    duplicated_packets_count = len(duplicated_packets)
    loss_percentages = (lost_packets_count / total_packets) * 100

    print(f'Total packets: {total_packets}')
    print(f'Lost packets: {lost_packets_count}')

```

```
print(f'Lost packet numbers: {sorted(list(lost_packets))}')
print(f'Duplicated packets: {duplicated_packets_count}')
print(f'Duplicated packet numbers: {sorted(list(duplicated_packets))}')
print(f'Loss percentage: {loss_percentages:.2f}%')

if __name__ == '__main__':
    analyze_results()
```

4 Вывод

Получила 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.