

# Отчёт по лабораторной работе №5

дисциплина: Компьютерный практикум по статистическому анализу  
данных

Студент: Кузнецова София Вадимовна

# Содержание

Цель работы	5
Теоретическое введение	6
Выполнение лабораторной работы	7
Выводы	16

# Список иллюстраций

0.1	Информацию о сетевых интерфейсах и IP-адресах хостов . . . . .	7
0.2	Проверка соединения между хостами . . . . .	7
0.3	Добавление потери пакетов . . . . .	8
0.4	Добавление потери пакетов . . . . .	9
0.5	Добавление значения корреляции для потери пакетов . . . . .	10
0.6	Добавление повреждения пакетов . . . . .	11
0.7	Добавление переупорядочивания пакетов . . . . .	12
0.8	Добавление дублирования пакетов . . . . .	13
0.9	Создание скрипта для эксперимента lab_netem_ii.py . . . . .	14
0.10	Редактирование скрипта . . . . .	14
0.11	Makefile . . . . .	15
0.12	Проведение эксперимента . . . . .	15

## Список таблиц

## Цель работы

Основной целью работы является изучение возможностей специализированных пакетов Julia для выполнения и оценки эффективности операций над объектами линейной алгебры.

# Теоретическое введение

Mininet[@mininet] – это эмулятор компьютерной сети. Под компьютерной сетью подразумеваются простые компьютеры — хосты, коммутаторы, а так же OpenFlow-контроллеры. С помощью простейшего синтаксиса в примитивном интерпретаторе команд можно разворачивать сети из произвольного количества хостов, коммутаторов в различных топологиях и все это в рамках одной виртуальной машины(ВМ). На всех хостах можно изменять сетевую конфигурацию, пользоваться стандартными утилитами(ifconfig, ping) и даже получать доступ к терминалу. На коммутаторы можно добавлять различные правила и маршрутизировать трафик.

# Выполнение лабораторной работы

Зададим простейшую топологию, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8. На хостах h1 и h2 введем команду ifconfig, чтобы отобразить информацию, относящуюся к их сетевым интерфейсам и назначенным им IP-адресам. В дальнейшем при работе с NETEM и командой tc будут использоваться интерфейсы h1-eth0 и h2-eth0.

```
root@mininet-vm:/home/mininet# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.1 brd 255.255.255.255 broadcast 10.255.255.255
        ether 72:8d:38:84:b8:99 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 brd 255.0.0.0
        loop txqueuelen 1000 (Local Loopback)
            RX packets 679 bytes 97244 (97.2 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 679 bytes 97244 (97.2 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
root@mininet-vm:/home/mininet# 

root@mininet-vm:/home/mininet# ifconfig
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.2 brd 255.255.255.255 broadcast 10.255.255.255
        ether 76:d9:1a:c3:ae:79 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 brd 255.0.0.0
        loop txqueuelen 1000 (Local Loopback)
            RX packets 787 bytes 103715 (103.7 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 787 bytes 103715 (103.7 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
root@mininet-vm:/home/mininet#
```

Рис. 0.1: Информацию о сетевых интерфейсах и IP-адресах хостов

Проверим подключение между хостами h1 и h2 с помощью команды ping с параметром -c 6.

```
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=18.2 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=2.52 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.537 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.093 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.102 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.097 ms
--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5103ms
rtt min/avg/max/mdev = 0.092/3.189/18.214/6.721 ms
root@mininet-vm:/home/mininet# 

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=2.52 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.094 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.099 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.093 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.099 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.094 ms
--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5083ms
rtt min/avg/max/mdev = 0.089/0.498/2.521/0.304 ms
root@mininet-vm:/home/mininet#
```

Рис. 0.2: Проверка соединения между хостами

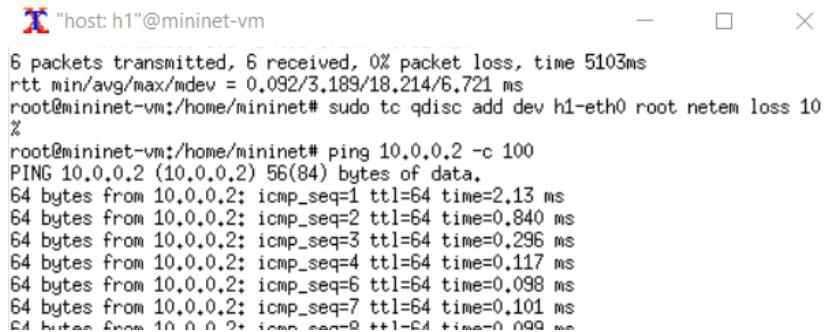
Пакеты могут быть потеряны в процессе передачи из-за таких факторов, как битовые ошибки и перегрузка сети. Скорость потери данных часто измеряется как процентная доля потерянных пакетов по отношению к количеству отправленных пакетов. На хосте h1 добавим 10% потерь пакетов к интерфейсу h1-eth0:

```
sudo tc qdisc add dev h1-eth0 root netem loss 10%
```

Здесь:

- sudo: выполнить команду с более высокими привилегиями;
- tc: вызвать управление трафиком Linux;
- qdisc: изменить дисциплину очередей сетевого планировщика;
- add: создать новое правило;
- dev h1-eth0: указать интерфейс, на котором будет применяться правило;
- netem: использовать эмулятор сети;
- loss 10%: 10% потерь пакетов.

Проверим, что на соединении от хоста h1 к хосту h2 имеются потери пакетов, используя команду ping с параметром -c 100 с хоста h1. Параметр -c указывает общее количество пакетов для отправки..

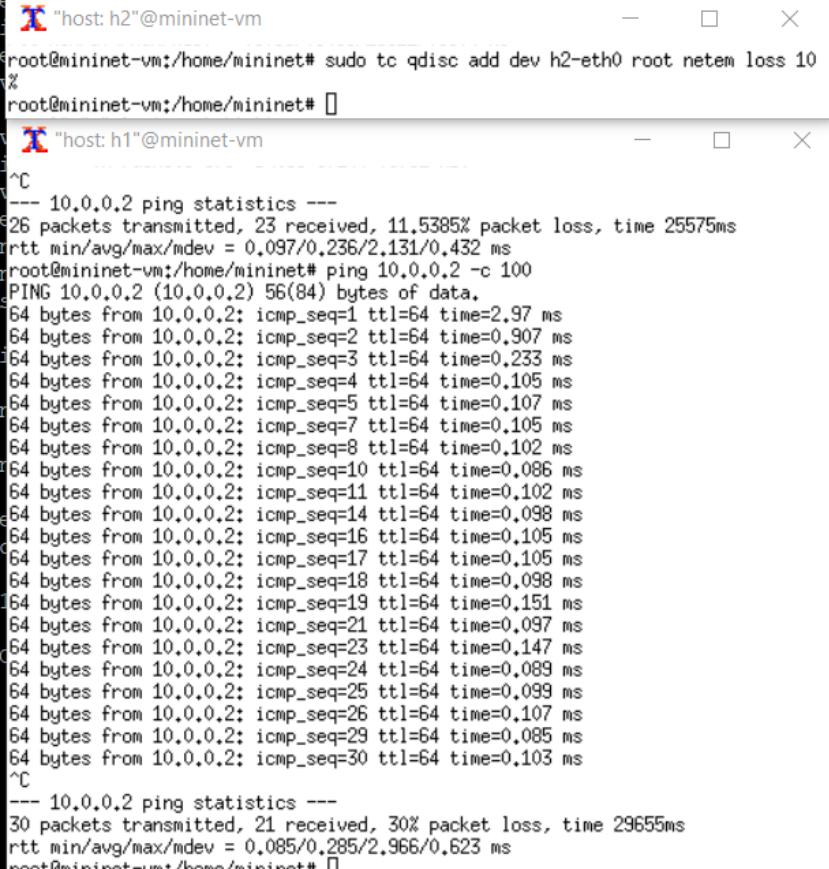


```
"host: h1"@mininet-vm
6 packets transmitted, 6 received, 0% packet loss, time 5103ms
rtt min/avg/max/mdev = 0.092/3.189/18.214/6.721 ms
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem loss 10%
root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 100
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=2.13 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.840 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.296 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.117 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.098 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=21 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=22 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=23 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=24 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=25 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=26 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=27 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=28 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=29 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=30 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=31 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=32 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=33 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=34 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=35 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=36 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=37 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=38 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=39 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=40 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=41 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=42 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=43 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=44 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=45 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=46 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=47 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=48 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=49 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=50 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=51 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=52 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=53 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=54 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=55 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=56 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=57 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=58 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=59 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=60 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=61 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=62 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=63 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=64 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=65 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=66 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=67 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=68 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=69 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=70 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=71 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=72 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=73 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=74 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=75 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=76 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=77 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=78 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=79 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=80 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=81 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=82 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=83 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=84 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=85 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=86 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=87 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=88 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=89 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=90 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=91 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=92 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=93 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=94 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=95 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=96 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=97 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=98 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=99 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=100 ttl=64 time=0.101 ms
root@mininet-vm:/home/mininet#
```

Рис. 0.3: Добавление потери пакетов

Для эмуляции глобальной сети с потерей пакетов в обоих направлениях необходимо к соответствующему интерфейсу на хосте h2 также добавить 10% потерь пакетов. Проверим, что соединение между хостом h1 и хостом h2 имеет больший процент

потерянных данных (10% от хоста h1 к хосту h2 и 10% от хоста h2 к хосту h1), повторив команду ping с параметром -c 100 на терминале хоста h1.



The screenshot shows two terminal windows side-by-side. The left window, titled "host: h2@mininet-vm", has a root prompt and runs the command "sudo tc qdisc add dev h2-eth0 root netem loss 10%". The right window, titled "host: h1@mininet-vm", also has a root prompt and runs "ping 10.0.0.2 -c 100". The output of the ping command shows 26 packets transmitted, 23 received, and 11.5385% packet loss. The detailed statistics for the first ping show 30 packets transmitted, 21 received, and 30% packet loss. The time for the second ping is 29655ms.

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h2-eth0 root netem loss 10%
root@mininet-vm:/home/mininet# 

root@mininet-vm:~# 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=2.97 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.907 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.233 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.105 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.107 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.105 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.102 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.086 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.102 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.098 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=0.105 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=0.105 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=0.098 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=0.151 ms
64 bytes from 10.0.0.2: icmp_seq=21 ttl=64 time=0.097 ms
64 bytes from 10.0.0.2: icmp_seq=23 ttl=64 time=0.147 ms
64 bytes from 10.0.0.2: icmp_seq=24 ttl=64 time=0.089 ms
64 bytes from 10.0.0.2: icmp_seq=25 ttl=64 time=0.099 ms
64 bytes from 10.0.0.2: icmp_seq=26 ttl=64 time=0.107 ms
64 bytes from 10.0.0.2: icmp_seq=29 ttl=64 time=0.085 ms
64 bytes from 10.0.0.2: icmp_seq=30 ttl=64 time=0.103 ms
^C
--- 10.0.0.2 ping statistics ---
30 packets transmitted, 21 received, 30% packet loss, time 29655ms
rtt min/avg/max/mdev = 0.085/0.285/2.966/0.623 ms
root@mininet-vm:/home/mininet#
```

Рис. 0.4: Добавление потери пакетов

Добавим на интерфейсе узла h1 коэффициент потери пакетов 50% (такой высокий уровень потери пакетов маловероятен), и каждая последующая вероятность зависит на 50% от предыдущей: Проверим, что на соединении от хоста h1 к хосту h2 имеются потери пакетов, используя команду ping с параметром -c 50 с хоста h1.

```

X "host: h1"@mininet-vm
22 packets transmitted, 22 received, 0% packet loss, time 21486ms
^tt min/avg/max/mdev = 0.093/0.263/2.705/0.553 ms
^oot@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem loss 50%
50%
^oot@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=3 ttl=64 time=0.539 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.127 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.167 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.116 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.099 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.098 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.116 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.104 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.102 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=0.100 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=0.100 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=0.103 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=0.098 ms
64 bytes from 10.0.0.2: icmp_seq=21 ttl=64 time=0.099 ms
64 bytes from 10.0.0.2: icmp_seq=25 ttl=64 time=0.335 ms
64 bytes from 10.0.0.2: icmp_seq=29 ttl=64 time=0.130 ms
64 bytes from 10.0.0.2: icmp_seq=32 ttl=64 time=0.099 ms
64 bytes from 10.0.0.2: icmp_seq=40 ttl=64 time=0.099 ms
64 bytes from 10.0.0.2: icmp_seq=42 ttl=64 time=0.104 ms
64 bytes from 10.0.0.2: icmp_seq=43 ttl=64 time=0.090 ms

--- 10.0.0.2 ping statistics ---
50 packets transmitted, 20 received, 60% packet loss, time 50169ms
^tt min/avg/max/mdev = 0.090/0.141/0.539/0.105 ms
^oot@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
^oot@mininet-vm:/home/mininet#

```

Рис. 0.5: Добавление значения корреляции для потери пакетов

Добавим на интерфейсе узла h1 0,01% повреждения пакетов. Проверим конфигурацию с помощью инструмента iPerf3 для проверки повторных передач.

The screenshot shows two terminal windows side-by-side. Both windows have a title bar with the host name and session ID: "host: h2" and "host: h1".

**Terminal h2 (Left):**

```

root@mininet-vm:/home/mininet# iperf3 -s
warning: this system does not seem to support IPv6 - trying IPv4
-----
Server listening on 5201
-----
Accepted connection from 10.0.0.1, port 58764
[ 7] local 10.0.0.2 port 5201 connected to 10.0.0.1 port 58766
[ ID] Interval Transfer Bitrate
[ 7] 0.00-1.00 sec 1.33 GBytes 11.5 Gbits/sec
[ 7] 1.00-2.00 sec 1.53 GBytes 13.1 Gbits/sec
[ 7] 2.00-3.00 sec 1.60 GBytes 13.8 Gbits/sec
[ 7] 3.00-4.00 sec 1.35 GBytes 11.6 Gbits/sec
[ 7] 4.00-5.00 sec 1.44 GBytes 12.3 Gbits/sec
[ 7] 5.00-6.00 sec 1.55 GBytes 13.3 Gbits/sec
[ 7] 6.00-7.00 sec 1.39 GBytes 11.9 Gbits/sec
[ 7] 7.00-8.00 sec 1.57 GBytes 13.4 Gbits/sec
[ 7] 8.00-9.00 sec 1.43 GBytes 12.3 Gbits/sec
[ 7] 9.00-10.00 sec 1.63 GBytes 14.0 Gbits/sec
[ 7] 10.00-10.00 sec 320 KBytes 3.65 Gbits/sec
-----
[ ID] Interval Transfer Bitrate
[ 7] 0.00-10.00 sec 14.8 GBytes 12.7 Gbits/sec
                                             receiver
-----
Server listening on 5201

```

**Terminal h1 (Right):**

```

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 58766 connected to 10.0.0.2 port 5201
[ ID] Interval Transfer Bitrate Retr Cwnd
[ 7] 0.00-1.00 sec 1.34 GBytes 11.5 Gbits/sec 3 506 KBytes
[ 7] 1.00-2.00 sec 1.53 GBytes 13.1 Gbits/sec 2 395 KBytes
[ 7] 2.00-3.00 sec 1.60 GBytes 13.8 Gbits/sec 2 402 KBytes
[ 7] 3.00-4.00 sec 1.35 GBytes 11.6 Gbits/sec 2 549 KBytes
[ 7] 4.00-5.00 sec 1.44 GBytes 12.4 Gbits/sec 4 1.43 MBytes
[ 7] 5.00-6.00 sec 1.55 GBytes 13.3 Gbits/sec 3 591 KBytes
[ 7] 6.00-7.00 sec 1.39 GBytes 11.9 Gbits/sec 2 871 KBytes
[ 7] 7.00-8.00 sec 1.56 GBytes 13.4 Gbits/sec 1 618 KBytes
[ 7] 8.00-9.00 sec 1.43 GBytes 12.3 Gbits/sec 4 535 KBytes
[ 7] 9.00-10.00 sec 1.63 GBytes 14.0 Gbits/sec 4 132 KBytes
-----
[ ID] Interval Transfer Bitrate Retr
[ 7] 0.00-10.00 sec 14.8 GBytes 12.7 Gbits/sec 27
                                             sender
[ 7] 0.00-10.00 sec 14.8 GBytes 12.7 Gbits/sec
                                             receiver
iperf Done.
root@mininet-vm:/home/mininet# 
```

Рис. 0.6: Добавление повреждения пакетов

Добавим на интерфейсе узла h1 следующее правило: 25% пакетов (со значением корреляции 50%) будут отправлены немедленно, а остальные 75% будут задержаны на 10 мс. Проверим, что на соединении от хоста h1 к хосту h2 имеются потери пакетов, используя команду ping с параметром -c 20 с хоста h1. Убедимся, что часть пакетов не будут иметь задержки (один из четырех, или 25%), а последующие несколько пакетов будут иметь задержку около 10 миллисекунд (три из четырех,

или 75%).

```
X "host: h1" @mininet-vm
oot@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 10
is reorder 252 502
rror: Exclusivity flag on, cannot modify.
oot@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.
4 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=10.5 ms
4 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=10.6 ms
4 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=10.6 ms
4 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=10.3 ms
4 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=10.7 ms
4 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=10.6 ms
4 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=10.6 ms
4 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=10.3 ms
4 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=10.5 ms
4 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=10.7 ms
4 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.094 ms
4 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=10.6 ms
4 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.083 ms
4 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=10.4 ms
4 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=10.7 ms
4 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=10.5 ms
4 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=10.6 ms
4 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=10.7 ms
4 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=10.4 ms
4 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=10.7 ms

-- 10.0.0.2 ping statistics --
0 packets transmitted, 20 received, 0% packet loss, time 19059ms
rtt min/avg/max/mdev = 0.083/9.492/10.730/3.137 ms
oot@mininet-vm:/home/mininet# ]
```

Рис. 0.7: Добавление переупорядочивания пакетов

Для интерфейса узла h1 зададим правило с дублированием 50% пакетов (т.е. 50% пакетов должны быть получены дважды): Проверим, что на соединении от хоста h1 к хосту h2 имеются дублированные пакеты, используя команду ping с параметром -c 20 с хоста h1. Дубликаты пакетов помечаются как DUP!.

```

X "host: h1"@mininet-vm - □ ×
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=0.416 ms
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.437 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.076 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.098 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.099 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.121 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.114 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.115 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.109 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.110 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.096 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.105 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.106 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.104 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.105 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.106 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.106 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.107 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.106 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.107 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.106 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.108 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.097 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.105 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.106 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=0.107 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=0.109 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=0.099 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=0.099 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=0.098 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=0.097 ms

--- 10.0.0.2 ping statistics ---
20 packets transmitted, 20 received, +11 duplicates, 0% packet loss, time 19412ms
rtt min/avg/max/mdev = 0.076/0.124/0.437/0.079 ms
root@mininet-vm:/home/mininet# █

```

Рис. 0.8: Добавление дублирования пакетов

В виртуальной среде mininet в своём рабочем каталоге с проектами создадим каталог simple-drop и перейдем в него. Создадим скрипт для эксперимента lab\_netem\_ii.py.

```

GNU nano 4.8
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 >' )

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

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

```

Рис. 0.9: Создание скрипта для эксперимента lab\_netem\_ii.py

Скорректируем скрипт так, чтобы на экран или в отдельный файл выводилась информация о потерях пакетов.

```

info( '*** Ping\n' )
h1.cmdPrint( 'ping -c 100', h2.IP(), '| grep "packet loss:" | awk \'{print $6, $7, $8}\' >'

```

Рис. 0.10: Редактирование скрипта

Создадим Makefile для управления процессом проведения эксперимента.

```
GNU nano 4.8                                     Makefile
all: ping.dat

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

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

Рис. 0.11: Makefile

Выполним эксперимент.

```
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
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 "packet loss:" | awk \'{print $6, $7, $8}\' > 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
```

Рис. 0.12: Проведение эксперимента

## Выводы

В результате выполнения данной лабораторной работы я получила навыки проведения интерактивных экспериментов в среде Mininet по исследованию параметров сети, связанных с потерей, дублированием, изменением порядка и повреждением пакетов при передаче данных.