

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра теории вероятностей и кибербезопасности

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

дисциплина: Моделирование сетей передачи данных

Студент: Быстров Глеб

Группа: НПИбд-01-20

МОСКВА

2023 г.

Цель работы

В данной лабораторной работе мне будет необходимо получить 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.

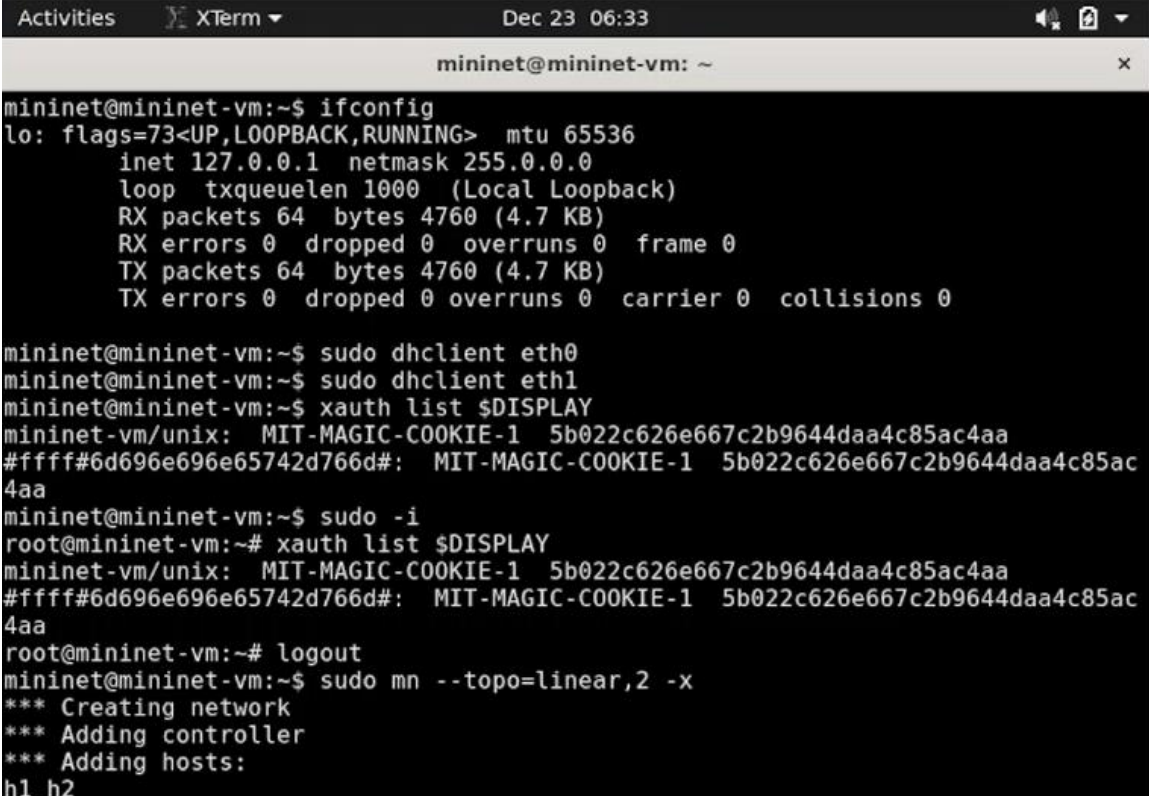
Описание процесса выполнения работы

Запуск лабораторной топологии

1. Запустите виртуальную среду с mininet. Задайте топологию сети, состоящую из двух хостов и двух коммутаторов с назначенной по умолчанию mininet сетью 10.0.0.0/8:

```
sudo mn --topo=linear,2 -x
```

После введения этой команды запустятся терминалы двух хостов, двух коммутаторов и контроллера (рис. 1).



```
Activities  XTerm  Dec 23 06:33
mininet@mininet-vm: ~

mininet@mininet-vm:~$ ifconfig
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 64  bytes 4760 (4.7 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 64  bytes 4760 (4.7 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

mininet@mininet-vm:~$ sudo dhclient eth0
mininet@mininet-vm:~$ sudo dhclient eth1
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix: MIT-MAGIC-COOKIE-1  5b022c626e667c2b9644daa4c85ac4aa
#ffff#6d696e696e65742d766d#: MIT-MAGIC-COOKIE-1  5b022c626e667c2b9644daa4c85ac4aa
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth list $DISPLAY
mininet-vm/unix: MIT-MAGIC-COOKIE-1  5b022c626e667c2b9644daa4c85ac4aa
#ffff#6d696e696e65742d766d#: MIT-MAGIC-COOKIE-1  5b022c626e667c2b9644daa4c85ac4aa
root@mininet-vm:~# logout
mininet@mininet-vm:~$ sudo mn --topo=linear,2 -x
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
```

Рис. 1. Запуск топологии сети

2. На хостах h1, h2 и на коммутаторах s1, s2 введите 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, h2-eth0, s1-eth2 (рис. 2-5).

```

"host: h1"
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

root@mininet-virtual-machine:~# 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 7e:c8:48:eb:47:bf 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 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

```

Рис. 2. h1

```

"host: h2"
root@mininet-virtual-machine:~# 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 66:8a:7b:38:7f:19 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 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

```

Рис. 3. h2

3. Проверьте подключение между хостами h1 и h2 с помощью команды `ping` с параметром `-c 6` (рис. 4).

Укажите в отчёте минимальное, среднее, максимальное и стандартное отклонение времени приёма-передачи (RTT), информацию о наличии или отсутствии потерь данных.

- Минимальное: 0.045

- Среднее: 0.502
- Максимальное: 2.538
- Стандартное отклонение: 0.913

```

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=2.54 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.260 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.045 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.063 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.060 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.048 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5096ms
rtt min/avg/max/mdev = 0.045/0.502/2.538/0.913 ms

```

Рис. 4. Проверка подключения

Добавление потери пакетов на интерфейс, подключённый к эмулируемой глобальной сети

4. 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 добавьте 10% потерь пакетов к интерфейсу h1-eth0: (рис. 5).

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.

```

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

```

Рис. 5. Конфигурация интерфейса

5. Проверьте, 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. Обратите внимание на значения `icmp_seq`. Некоторые номера последовательности отсутствуют из-за потери пакетов. В сводном отчёте `ping` сообщает о проценте потерянных пакетов после завершения передачи (рис. 6).

```
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=0.305 ms
 64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.055 ms
 64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.056 ms
 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=0.051 ms
 64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.053 ms
 64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.054 ms
 64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.122 ms
 64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.102 ms
 64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.092 ms
 64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.053 ms
 64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.119 ms
```

Рис. 6. Проверка соединения

6. Для эмуляции глобальной сети с потерей пакетов в обоих направлениях необходимо к соответствующему интерфейсу на хосте h2 также добавить 10% потерь пакетов (рис. 7).

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

Рис. 7. Конфигурация интерфейса

7. Проверьте, что соединение между хостом h1 и хостом h2 имеет большой 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. Укажите в отчёте отсутствующие из-за потери

пакетов номера последовательности (значения `icmp_seq`), процент потерянных пакетов после завершения передачи (рис. 8).

- Отсутствующие из-за потери пакетов номера: 1, 12, 16, 18, 28, 32, 47, 50, 51, 81, 87, 97
- Процент потерянных пакетов после завершения передачи: 13%.

```
64 bytes from 10.0.0.2: icmp_seq=95 ttl=64 time=0.082 ms
64 bytes from 10.0.0.2: icmp_seq=96 ttl=64 time=0.079 ms
64 bytes from 10.0.0.2: icmp_seq=98 ttl=64 time=0.050 ms
64 bytes from 10.0.0.2: icmp_seq=99 ttl=64 time=0.060 ms
64 bytes from 10.0.0.2: icmp_seq=100 ttl=64 time=0.056 ms

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 87 received, 13% packet loss, time 101298ms
rtt min/avg/max/mdev = 0.046/0.098/1.032/0.150 ms
root@mininet-vm:/home/mininet#
```

Рис. 8. Информация о пакетах

8. Восстановите конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса. Для отправителя `h1` и получателя `h2`: (рис. 9-10).

```
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
```

Рис. 9. Восстановление конфигурации для `h1`

```
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h2-eth0 root netem
root@mininet-vm:/home/mininet#
```

Рис. 10. Восстановление конфигурации для `h2`

Добавление значения корреляции для потери пакетов в эмулируемой глобальной сети

9. Добавьте на интерфейсе узла `h1` коэффициент потери пакетов 50% (такой высокий уровень потери пакетов маловероятен), и каждая последующая вероятность зависит на 50% от последней (рис. 11).

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem
m loss 50% 50%
```

Рис. 11. Конфигурация интерфейса

10. Проверьте, что на соединении от хоста h1 к хосту h2 имеются потери пакетов, используя команду `ping` с параметром `-c 50` с хоста h1. Укажите в отчёте отсутствующие из-за потери пакетов номера последовательности (значения `icmp_seq`), процент потерянных пакетов после завершения передачи (рис. 12).

- Отсутствующие из-за потери пакетов номера: 14, 15, 16, 17, 20, 22, 23, 24, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 39, 42.
- Процент потерянных пакетов после завершения передачи: 64%.

```
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.062 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.051 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.091 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.051 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=0.048 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=0.049 ms
64 bytes from 10.0.0.2: icmp_seq=21 ttl=64 time=0.063 ms
64 bytes from 10.0.0.2: icmp_seq=25 ttl=64 time=0.048 ms
64 bytes from 10.0.0.2: icmp_seq=38 ttl=64 time=1.88 ms
64 bytes from 10.0.0.2: icmp_seq=40 ttl=64 time=0.051 ms
64 bytes from 10.0.0.2: icmp_seq=41 ttl=64 time=0.049 ms
64 bytes from 10.0.0.2: icmp_seq=43 ttl=64 time=0.050 ms

--- 10.0.0.2 ping statistics ---
50 packets transmitted, 18 received, 64% packet loss, time 50123ms
rtt min/avg/max/mdev = 0.048/0.176/1.875/0.418 ms
```

Рис. 12. Информация о пакетах

11. Восстановите конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса. Для отправителя h1: (рис. 13).

```
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
```

Рис. 13. Восстановление конфигурации для h1

Добавление повреждения пакетов в эмулируемой глобальной сети

12. Добавьте на интерфейсе узла h1 0,01% повреждения пакетов: (рис. 14).

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem
m corrupt 0.01%
root@mininet-vm:/home/mininet#
```


Рис. 14. Конфигурация интерфейса

13. Проверьте конфигурацию с помощью инструмента iPerf3 для проверки повторных передач. Для этого:

– запустите iPerf3 в режиме сервера в терминале хоста h2: (рис. 15)

iPerf3 -s

запустите iPerf3 в клиентском режиме в терминале хоста h1: (рис. 16)

iPerf3 -c 10.0.0.2

- Значения повторной передачи на каждом временном интервале:
 - 4
 - 4
 - 5
 - 2
 - 2
 - 3
 - 4
 - 3
 - 3
 - 3
- Общее количество повторно переданных пакетов: 33

```

Server listening on 5201
-----
Accepted connection from 10.0.0.1, port 53420
[ 7] local 10.0.0.2 port 5201 connected to 10.0.0.1 port 53422
[ ID] Interval          Transfer      Bitrate
[ 7]  0.00-1.00      sec  1.32 GBytes  11.4 Gbits/sec
[ 7]  1.00-2.00      sec  1.50 GBytes  12.9 Gbits/sec
[ 7]  2.00-3.00      sec  1.40 GBytes  12.0 Gbits/sec
[ 7]  3.00-4.00      sec  1.47 GBytes  12.6 Gbits/sec
[ 7]  4.00-5.00      sec  1.45 GBytes  12.5 Gbits/sec
[ 7]  5.00-6.00      sec  1.43 GBytes  12.3 Gbits/sec
[ 7]  6.00-7.00      sec  1.50 GBytes  12.9 Gbits/sec
[ 7]  7.00-8.00      sec  1.43 GBytes  12.3 Gbits/sec
[ 7]  8.00-9.00      sec  1.49 GBytes  12.8 Gbits/sec
[ 7]  9.00-10.00     sec  1.40 GBytes  12.0 Gbits/sec
[ 7] 10.00-10.01     sec  15.9 MBytes  10.4 Gbits/sec
-----
[ ID] Interval          Transfer      Bitrate
[ 7]  0.00-10.01     sec  14.4 GBytes  12.4 Gbits/sec
ceiver

```

Рис. 15. Режим сервера в терминале хоста h2

```

"host: h1" x
[ 7]  3.00-4.02      sec  1.45 GBytes  12.2 Gbits/sec  2  1.16 MBytes
[ 7]  4.02-5.00      sec  1.46 GBytes  12.8 Gbits/sec  2  1.51 MBytes
[ 7]  5.00-6.00      sec  1.42 GBytes  12.2 Gbits/sec  3   826 KBytes
[ 7]  6.00-7.00      sec  1.51 GBytes  12.9 Gbits/sec  4  1.52 MBytes
[ 7]  7.00-8.00      sec  1.42 GBytes  12.3 Gbits/sec  3  1.21 MBytes
[ 7]  8.00-9.00      sec  1.49 GBytes  12.8 Gbits/sec  3   713 KBytes
[ 7]  9.00-10.00     sec  1.41 GBytes  12.1 Gbits/sec  3  1.10 MBytes
-----
[ ID] Interval          Transfer      Bitrate      Retr
[ 7]  0.00-10.00     sec  14.4 GBytes  12.4 Gbits/sec  33
ender
[ 7]  0.00-10.01     sec  14.4 GBytes  12.4 Gbits/sec
ceiver
iperf Done.

```

Рис. 16. Клиентский режим в терминале хоста h1

14. Восстановите конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса. Для отправителя h1: (рис. 17).

```
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
```

Рис. 17. Восстановление конфигурации для h1

Добавление переупорядочивания пакетов в интерфейс подключения к эмулируемой глобальной сети

15. Добавьте на интерфейсе узла h1 правило (рис. 18).

Здесь 25% пакетов (со значением корреляции 50%) будут отправлены немедленно, а остальные 75% будут задержаны на 10 мс.

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 10ms reorder 25% 50%
```

Рис. 18. Конфигурация интерфейса

16. Проверьте, что на соединении от хоста h1 к хосту h2 имеются потери пакетов, используя команду `ping` с параметром `-c 20` с хоста h1. Убедитесь, 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%). При необходимости повторите тест. Укажите в отчёте отсутствующие из-за потери пакетов номера последовательности (значения `icmp_seq`), процент потерянных пакетов после завершения передачи.

- Отсутствующие из-за потери пакетов номера последовательности (значения `icmp_seq`): -
- Процент потерянных пакетов после завершения передачи: 0%

```

[ ID] Ir 64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=10.4 ms
[ 7] 64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=10.8 ms
[ 7] 64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=10.9 ms
[ 7] 64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=10.8 ms
[ 7] 64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=10.2 ms
[ 7] 64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=10.2 ms
[ 7] 64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=10.8 ms
[ 7] 64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=10.3 ms
[ 7] 64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=10.4 ms
[ 7] 64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=10.8 ms
[ 7] 64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=10.8 ms
[ 7] 64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=10.7 ms
[ 7] 64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=10.4 ms
[ ID] Ir 64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=10.4 ms
[ 7] 64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=10.8 ms
ceiver --- 10.0.0.2 ping statistics ---
----- 20 packets transmitted, 20 received, 0% packet loss, time 19041ms
Server 10.0.0.1: icmp_seq=1 ttl=64 time=10.334 ms
Server 10.0.0.1: icmp_seq=2 ttl=64 time=10.746 ms
Server 10.0.0.1: icmp_seq=3 ttl=64 time=10.637 ms
Server 10.0.0.1: icmp_seq=4 ttl=64 time=10.716 ms

```

Рис. 19. Информация о пакетах

17. Восстановите конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса (рис. 20).

```

root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
m

```

Рис. 20. Восстановление конфигурации для h1

Добавление дублирования пакетов в интерфейс подключения к эмулируемой глобальной сети

18. Для интерфейса узла h1 задайте правило с дублированием 50% пакетов (т.е. 50% пакетов должны быть получены дважды) (рис. 21):

```

root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem
duplicate 50%

```

Рис. 21. Конфигурация интерфейса

19. Проверьте, 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!`. Измеренная скорость дублирования пакетов будет приближаться к настроенной скорости по мере выполнения большего количества попыток (рис. 22).

```

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=1.52 ms
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.55 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.910 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.167 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.056 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.052 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.052 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.207 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.211 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.054 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.054 ms (DUP!)
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=8 ttl=64 time=0.059 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.058 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.081 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.081 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.104 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.105 ms (DUP!)

```

Рис. 22. Информация о пакетах

20. Восстановите конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса (рис. 23).

```

root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root nete
m

```

Рис. 23. Восстановление конфигурации для h1

Предварительная подготовка

21. Для каждого воспроизводимого эксперимента expname создайте свой каталог, в котором будут размещаться файлы эксперимента (рис. 24):

```
mkdir -p ~/work/lab_netem_ii/expname
```

Здесь expname может принимать значения simple-drop, correlation-drop и т.п. Для каждого случая создайте скрипт для проведения эксперимента lab_netem_ii.py.

```

mininet@mininet-vm:~$ mkdir /home/mininet/work/lab_netem_ii/expname
mininet@mininet-vm:~$ cd /home/mininet/work/lab_netem_ii/expname
mininet@mininet-vm:~/work/lab_netem_ii/expname$ cd
mininet@mininet-vm:~$ mkdir /home/mininet/work/lab_netem_ii/simple_drop
mininet@mininet-vm:~$ cd /home/mininet/work/lab_netem_ii/simple_drop
mininet@mininet-vm:~/work/lab_netem_ii/simple_drop$ nano lab_netem_ii.py

```


Рис. 24. Работа с каталогами и файлами

Добавление потери пакетов на интерфейс, подключённый к эмулируемой глобальной сети

22. Создаёте скрипт для эксперимента lab_netem_ii.py (рис. 25-26)

В строках 38-39 задаётся информация о потерях.

```
#!/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' )
```

Рис. 25. Скрипт lab_netem_ii.py

```
    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}' )

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

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

Рис. 26. Скрипт lab_netem_ii.py

23. Скорректируйте скрипт так, чтобы на экран или в отдельный файл выводилась информация о потерях пакетов (рис. 27).

```
def analyze(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_packet_count = len(lost_packets)

    print(f'Total packets: {total_packets}')
    print(f'Lost packets: {lost_packet_count}')
    print(f'Lost packet numbers: {sorted(list(lost_packets))}')
```

Рис. 27. Скрипт lab_netem_ii.py

24. Выполните эксперимент (рис. 28) и результаты (рис. 29):

```
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
```

Рис. 28. Работа скрипта

```
Total packets: 100
Lost packets: 4
Lost packet numbers: [19, 26, 44, 81]
```

Рис. 29. Результаты

25. Очистите каталог от результатов проведения экспериментов (рис. 30):

make clean


```
mininet@mininet-vm:~/work/lab_netem_ii/simple_drop$ make clean
rm -f *.dat
```

Рис. 30. Очистка каталога

Задание для самостоятельной работы

26. Самостоятельно реализуйте воспроизводимые эксперименты по исследованию параметров сети, связанных с потерей, изменением порядка и повреждением пакетов при передаче данных (рис. 31-34).

```
#!/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' )
```

Рис. 31. Скрипт lab_netem_ii.py

```
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%' )
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}' )

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

Рис. 32. Скрипт lab_netem_ii.py

```
def analyze(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 = float(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)

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

    lost_packet_count = len(lost_packets)
```

Рис. 33. Скрипт lab_netem_ii.py

```
print(f'Total packets: {total_packets}')
print(f'Lost packets: {lost_packet_count}')
print(f'Lost packet numbers: {sorted(list(lost_packets))}')
print(f'Duplicated packets: {duplicated_packet_count}')
print(f'Duplicated packet numbers: {sorted(list(duplicated_packets))}')

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

Рис. 34. Скрипт lab_netem_ii.py

27. Выполнил эксперимент и получил результаты (рис. 35):

```
Duplicated packets: 51
Duplicated packet numbers: [2.0, 3.0, 4.0, 6.0, 9.0, 12.0, 13.0, 14.0, 17.0, 21.0, 22.0, 23.0, 25.0, 30.0, 31.0, 32.0, 34.0, 35.0, 36.0, 39.0, 40.0, 42.0, 43.0, 47.0, 53.0, 55.0, 57.0, 58.0, 59.0, 60.0, 61.0, 62.0, 63.0, 64.0, 66.0, 67.0, 70.0, 72.0, 74.0, 77.0, 82.0, 84.0, 86.0, 87.0, 88.0, 90.0, 91.0, 95.0, 96.0, 99.0, 100.0]
```

Рис. 35. Результаты эксперимента

28. Очистил каталог от результатов проведения экспериментов (рис. 36):

make clean

```
mininet@mininet-vm:~/work/lab_netem_ii/simple_drop$ make clean
rm -f *.dat
```

Рис. 36. Очистка каталога

Вывод

В данной лабораторной работе мне успешно удалось получить 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.