

# **Отчёт по лабораторной работе №5**

## **Моделирование сетей передачи данных**

**Эмуляция и измерение потерь пакетов в глобальных сетях**

Выполнил: Махорин Иван Сергеевич,  
НПИбд-02-21, 1032211221

# Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Вывод	26
4	Список литературы. Библиография	27

# Список иллюстраций

2.1	Исправление прав запуска X-соединения в виртуальной машине mininet . . . . .	6
2.2	Создание простейшей топологии . . . . .	7
2.3	Отображение информации их сетевых интерфейсов и IP-адресов . . . . .	8
2.4	Проверка подключения между хостами h1 и h2 . . . . .	9
2.5	Добавление 10% потерь пакетов на хосте h1 . . . . .	9
2.6	Проверка . . . . .	10
2.7	Добавление 10% потерь пакетов на хосте h2 . . . . .	10
2.8	Проверка . . . . .	11
2.9	Восстановление конфигурации по умолчанию для хоста h1 и хоста h2 . . . . .	11
2.10	Проверка . . . . .	12
2.11	Добавление на узле h1 коэффициента потери пакетов 50% . . . . .	12
2.12	Проверка . . . . .	13
2.13	Восстановление конфигурации интерфейса по умолчанию . . . . .	13
2.14	Добавление на узле h1 0.01% повреждения пакетов . . . . .	13
2.15	Проверка конфигурации с помощью инструмента iPerf3 для проверки повторных передач . . . . .	14
2.16	Восстановление конфигурации интерфейса по умолчанию . . . . .	14
2.17	Добавление на узле h1 правила из лабораторной работы . . . . .	15
2.18	Проверка . . . . .	15
2.19	Восстановление конфигурации интерфейса по умолчанию . . . . .	15
2.20	Добавление на узле h1 правила с дублированием 50% пакетов . . . . .	16
2.21	Проверка . . . . .	17
2.22	Восстановление конфигурации интерфейса по умолчанию . . . . .	17
2.23	Создание каталога exrname . . . . .	18
2.24	Создание каталога simple-drop и дальнейшее его открытие . . . . .	18
2.25	Создание скрипта для эксперимента . . . . .	19
2.26	Создание нового скрипта для вывода информации о потере пакетов . . . . .	20
2.27	Создание Makefile и помещение в него скрипта . . . . .	20
2.28	Выполнение эксперимента и последующая очистка каталога . . . . .	21
2.29	Реализация воспроизводимого эксперимента по исследованию параметров сети . . . . .	22
2.30	Реализация воспроизводимого эксперимента по исследованию параметров сети . . . . .	23
2.31	Реализация воспроизводимого эксперимента по исследованию параметров сети . . . . .	24

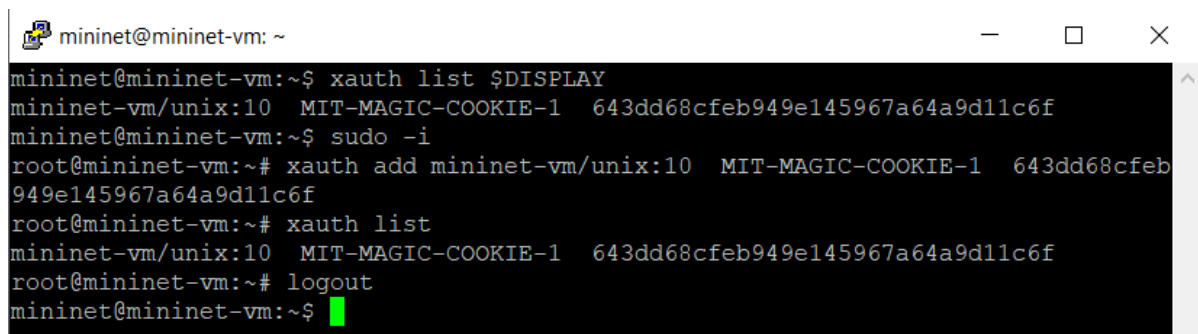
2.32 Реализация воспроизводимого эксперимента по исследованию параметров сети . . . . .	25
---	----

# 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

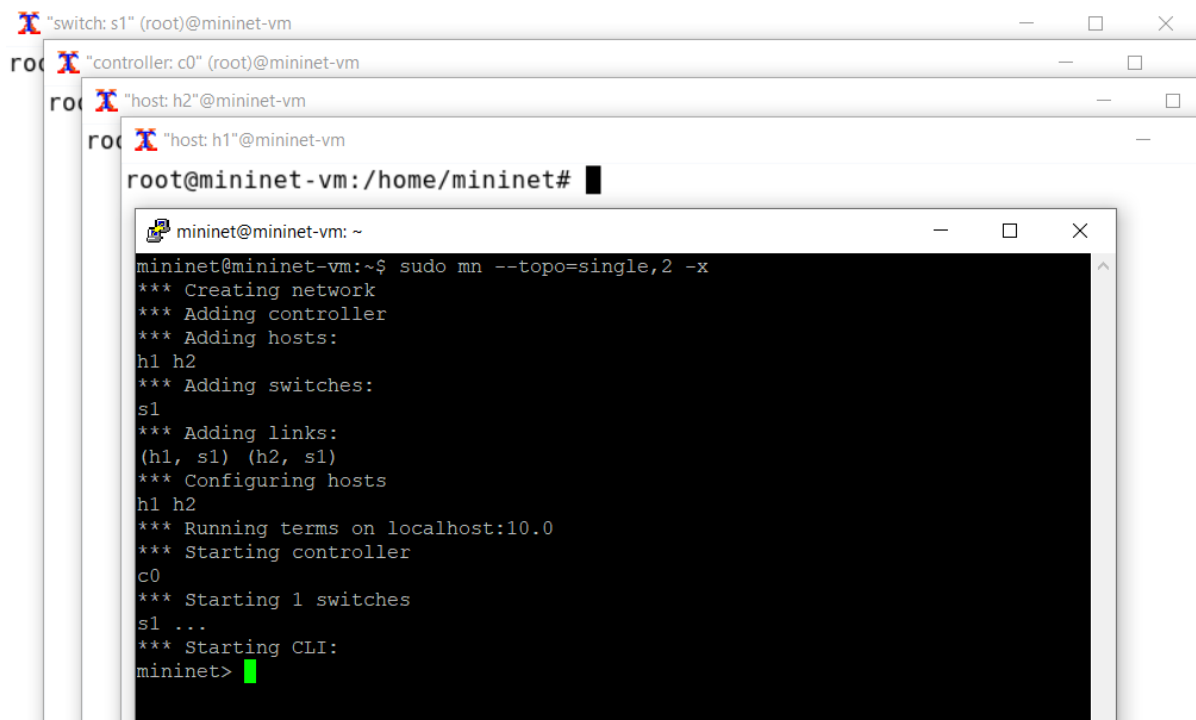
В виртуальной машине mininet исправим права запуска X-соединения (рис. 2.1):



```
mininet@mininet-vm: ~  
mininet@mininet-vm:~$ xauth list $DISPLAY  
mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  643dd68cfef949e145967a64a9d11c6f  
mininet@mininet-vm:~$ sudo -i  
root@mininet-vm:~# xauth add mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  643dd68cfef949e145967a64a9d11c6f  
root@mininet-vm:~# xauth list  
mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  643dd68cfef949e145967a64a9d11c6f  
root@mininet-vm:~# logout  
mininet@mininet-vm:~$
```

Рис. 2.1: Исправление прав запуска X-соединения в виртуальной машине mininet

Зададим простейшую топологию, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8 (рис. 2.2):

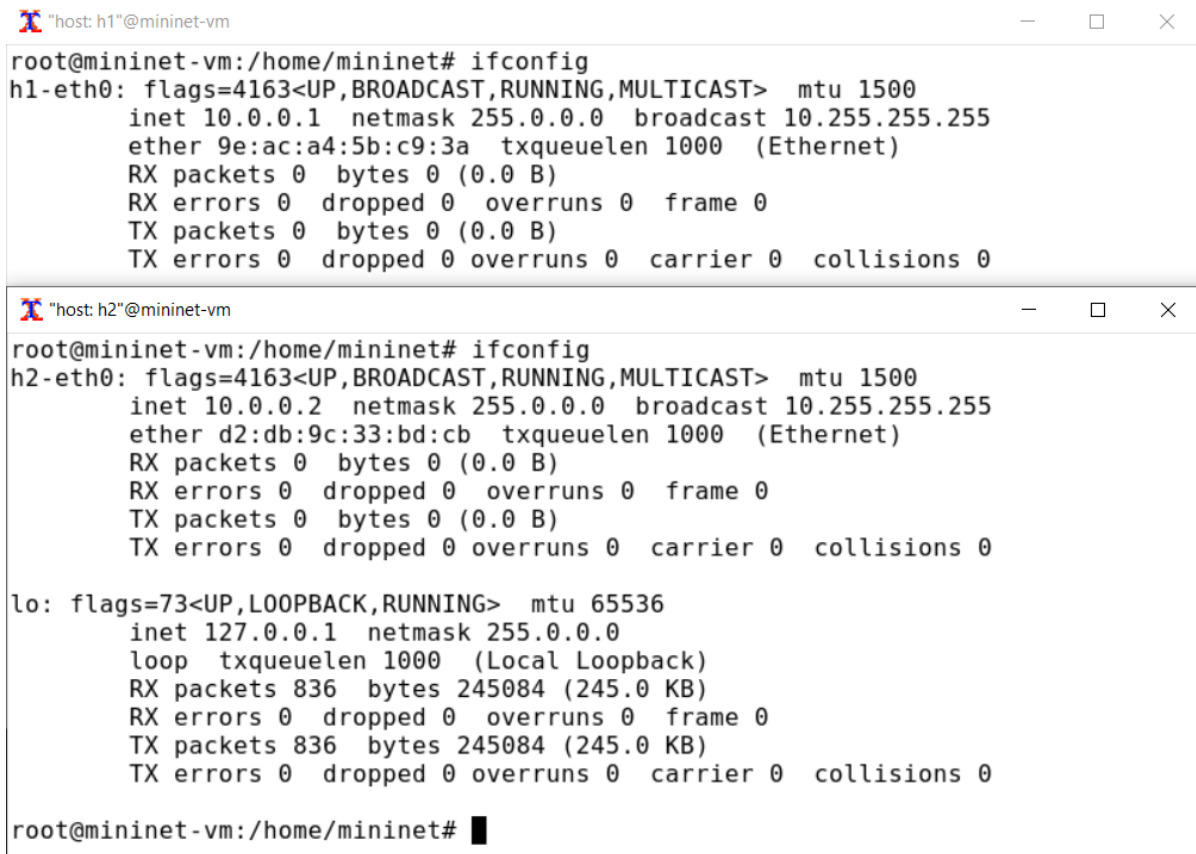


The image shows a stack of terminal windows from the Mininet environment. The top window is the root prompt at mininet-vm. Below it are prompts for switch s1, controller c0, and host h2. The bottom window is the mininet user prompt, where the command `sudo mn --topo=single,2 -x` has been executed. The output shows the step-by-step creation of the network topology.

```
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:
mininet>
```

Рис. 2.2: Создание простейшей топологии

На хостах h1 и h2 введём 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 (рис. 2.3):



The image shows two terminal windows from a Mininet VM. The top window is for host h1, showing the output of the 'ifconfig' command for the 'h1-eth0' interface. It displays an IP address of 10.0.0.1, a netmask of 255.0.0.0, a broadcast address of 10.255.255.255, and an Ethernet MAC address of 9e:ac:a4:5b:c9:3a. The bottom window is for host h2, showing the output of the 'ifconfig' command for the 'h2-eth0' interface. It displays an IP address of 10.0.0.2, a netmask of 255.0.0.0, a broadcast address of 10.255.255.255, and an Ethernet MAC address of d2:db:9c:33:bd:cb. Additionally, the 'lo' (loopback) interface is shown for h2 with IP 127.0.0.1 and a netmask of 255.0.0.0.

```
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 9e:ac:a4:5b:c9:3a 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

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 d2:db:9c:33:bd:cb 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

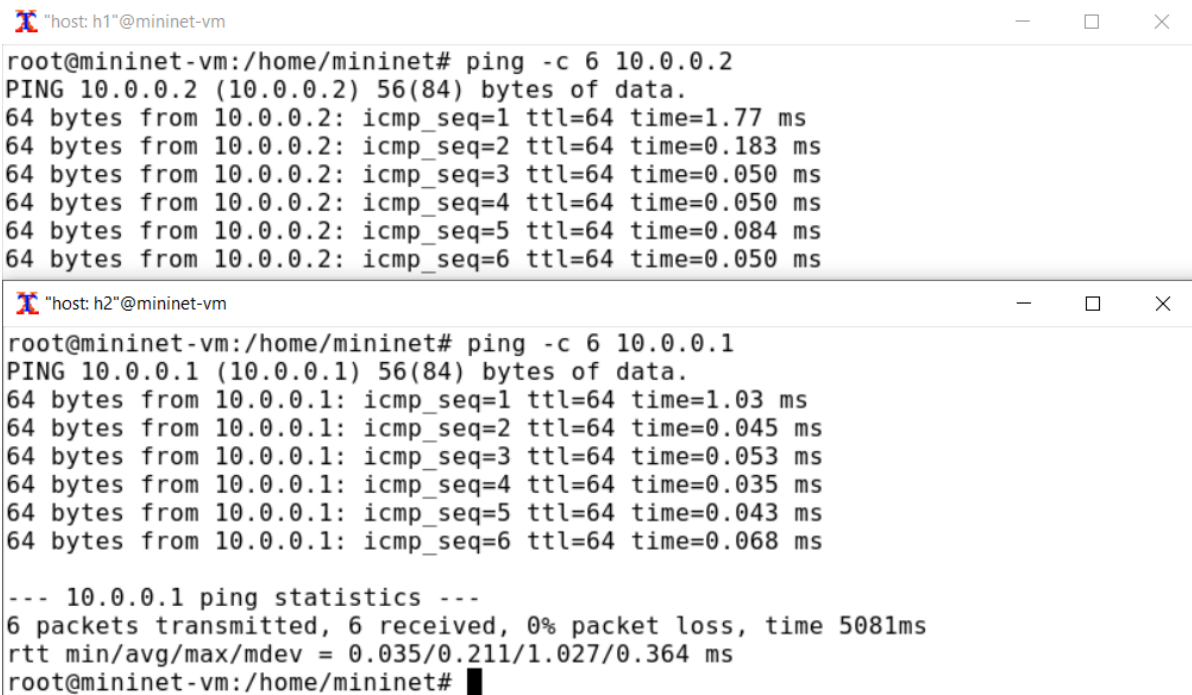
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 836 bytes 245084 (245.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 836 bytes 245084 (245.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet#
```

Рис. 2.3: Отображение информации их сетевых интерфейсов и IP-адресов

Проверим подключение между хостами h1 и h2 с помощью команды ping с параметром -c 6 (рис. 2.4):





The image shows two terminal windows from a Mininet VM. The top window, titled "host: h1", shows a ping command being executed from root@mininet-vm to 10.0.0.2. The output shows six successful ping requests with varying times. The bottom window, titled "host: h2", shows a ping command being executed from root@mininet-vm to 10.0.0.1. The output shows six successful ping requests, followed by a summary of the statistics: 6 packets transmitted, 6 received, 0% packet loss, and a total time of 5081ms.

```
root@mininet-vm:/home/mininet# ping -c 6 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.77 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.183 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.050 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=0.084 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.050 ms

root@mininet-vm:/home/mininet# ping -c 6 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=1.03 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.045 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.053 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.035 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.043 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.068 ms

--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5081ms
rtt min/avg/max/mdev = 0.035/0.211/1.027/0.364 ms
root@mininet-vm:/home/mininet#
```

Рис. 2.4: Проверка подключения между хостами h1 и h2

На хосте h1 добавим 10% потерь пакетов к интерфейсу h1-eth0 (рис. 2.5):



The image shows a terminal window titled "host: h1" from a Mininet VM. The root user at mininet-vm executes the command 'sudo tc qdisc add dev h1-eth0 root netem loss 10 %'. The output shows the command was executed successfully, followed by a prompt for the next command.

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem loss 10
%
root@mininet-vm:/home/mininet#
```

Рис. 2.5: Добавление 10% потерь пакетов на хосте h1

Проверим, что на соединении от хоста h1 к хосту h2 имеются потери пакетов, используя команду ping с параметром -c 100 с хоста h1 (рис. 2.6):

```
host: h1@mininet-vm
64 bytes from 10.0.0.2: icmp_seq=81 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=82 ttl=64 time=0.066 ms
64 bytes from 10.0.0.2: icmp_seq=84 ttl=64 time=0.059 ms
64 bytes from 10.0.0.2: icmp_seq=85 ttl=64 time=0.047 ms
64 bytes from 10.0.0.2: icmp_seq=86 ttl=64 time=0.057 ms
64 bytes from 10.0.0.2: icmp_seq=87 ttl=64 time=0.056 ms
64 bytes from 10.0.0.2: icmp_seq=88 ttl=64 time=0.044 ms
64 bytes from 10.0.0.2: icmp_seq=89 ttl=64 time=0.058 ms
64 bytes from 10.0.0.2: icmp_seq=90 ttl=64 time=0.041 ms
64 bytes from 10.0.0.2: icmp_seq=91 ttl=64 time=0.082 ms
64 bytes from 10.0.0.2: icmp_seq=92 ttl=64 time=0.041 ms
64 bytes from 10.0.0.2: icmp_seq=93 ttl=64 time=0.069 ms
64 bytes from 10.0.0.2: icmp_seq=94 ttl=64 time=0.181 ms
64 bytes from 10.0.0.2: icmp_seq=95 ttl=64 time=0.044 ms
64 bytes from 10.0.0.2: icmp_seq=96 ttl=64 time=0.044 ms
64 bytes from 10.0.0.2: icmp_seq=97 ttl=64 time=0.041 ms
64 bytes from 10.0.0.2: icmp_seq=98 ttl=64 time=0.057 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=0.116 ms

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 93 received, 7% packet loss, time 101324ms
rtt min/avg/max/mdev = 0.037/0.076/0.868/0.100 ms
root@mininet-vm:/home/mininet#
```

Рис. 2.6: Проверка

Для эмуляции глобальной сети с потерей пакетов в обоих направлениях необходимо к соответствующему интерфейсу на хосте h2 также добавить 10% потерь пакетов (рис. 2.7):

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

Рис. 2.7: Добавление 10% потерь пакетов на хосте h2

Проверим, 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 (рис. 2.8):

```
"host: h1"@mininet-vm
64 bytes from 10.0.0.2: icmp_seq=76 ttl=64 time=0.055 ms
64 bytes from 10.0.0.2: icmp_seq=77 ttl=64 time=0.041 ms
64 bytes from 10.0.0.2: icmp_seq=78 ttl=64 time=0.069 ms
64 bytes from 10.0.0.2: icmp_seq=79 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=80 ttl=64 time=0.053 ms
64 bytes from 10.0.0.2: icmp_seq=81 ttl=64 time=0.063 ms
64 bytes from 10.0.0.2: icmp_seq=82 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=84 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=85 ttl=64 time=0.169 ms
64 bytes from 10.0.0.2: icmp_seq=87 ttl=64 time=0.056 ms
64 bytes from 10.0.0.2: icmp_seq=88 ttl=64 time=0.053 ms
64 bytes from 10.0.0.2: icmp_seq=89 ttl=64 time=0.045 ms
64 bytes from 10.0.0.2: icmp_seq=90 ttl=64 time=0.046 ms
64 bytes from 10.0.0.2: icmp_seq=94 ttl=64 time=0.107 ms
64 bytes from 10.0.0.2: icmp_seq=95 ttl=64 time=0.046 ms
64 bytes from 10.0.0.2: icmp_seq=97 ttl=64 time=0.046 ms
64 bytes from 10.0.0.2: icmp_seq=98 ttl=64 time=0.077 ms
64 bytes from 10.0.0.2: icmp_seq=99 ttl=64 time=0.051 ms
64 bytes from 10.0.0.2: icmp_seq=100 ttl=64 time=0.042 ms

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 82 received, 18% packet loss, time 101302ms
rtt min/avg/max/mdev = 0.040/0.073/0.619/0.077 ms
root@mininet-vm:/home/mininet#
```

Рис. 2.8: Проверка

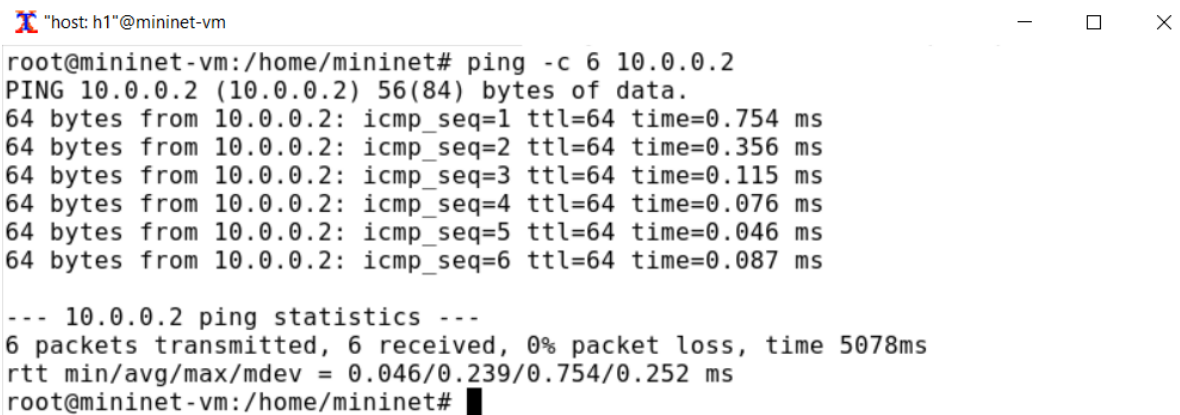
Восстановим конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса (рис. 2.9):

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

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

Рис. 2.9: Восстановление конфигурации по умолчанию для хоста h1 и хоста h2

Убедимся, что соединение от хоста h1 к хосту h2 не имеет явной потери пакетов, запустив команду `ping` с терминала хоста h1 и затем нажав `Ctrl + c`, чтобы остановить тест (рис. 2.10):

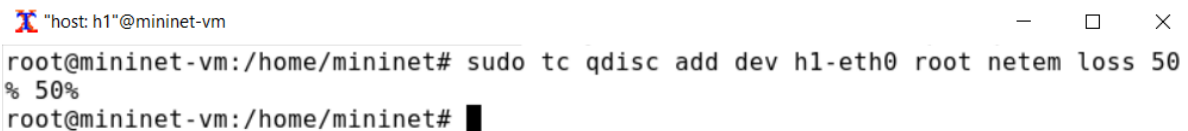


```
root@mininet-vm:/home/mininet# ping -c 6 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.754 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.356 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.115 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.076 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.046 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.087 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5078ms
rtt min/avg/max/mdev = 0.046/0.239/0.754/0.252 ms
root@mininet-vm:/home/mininet#
```

Рис. 2.10: Проверка

Добавим на интерфейсе узла h1 коэффициент потери пакетов 50% (такой высокий уровень потери пакетов маловероятен), и каждая последующая вероятность зависит на 50% от последней (рис. 2.11):



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

Рис. 2.11: Добавление на узле h1 коэффициента потери пакетов 50%

Проверим, что на соединении от хоста h1 к хосту h2 имеются потери пакетов, используя команду ping с параметром -c 50 с хоста h1 (рис. 2.12):

```
"host: h1"@mininet-vm
From 10.0.0.1 icmp_seq=9 Destination Host Unreachable
From 10.0.0.1 icmp_seq=10 Destination Host Unreachable
From 10.0.0.1 icmp_seq=11 Destination Host Unreachable
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=1025 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.490 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.046 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.062 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=0.049 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=0.067 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=0.048 ms
64 bytes from 10.0.0.2: icmp_seq=22 ttl=64 time=0.065 ms
64 bytes from 10.0.0.2: icmp_seq=24 ttl=64 time=0.047 ms
64 bytes from 10.0.0.2: icmp_seq=25 ttl=64 time=0.054 ms
64 bytes from 10.0.0.2: icmp_seq=30 ttl=64 time=0.078 ms
64 bytes from 10.0.0.2: icmp_seq=31 ttl=64 time=0.053 ms
64 bytes from 10.0.0.2: icmp_seq=42 ttl=64 time=0.404 ms
64 bytes from 10.0.0.2: icmp_seq=44 ttl=64 time=0.058 ms
64 bytes from 10.0.0.2: icmp_seq=45 ttl=64 time=0.075 ms
64 bytes from 10.0.0.2: icmp_seq=50 ttl=64 time=0.101 ms

--- 10.0.0.2 ping statistics ---
50 packets transmitted, 20 received, +3 errors, 60% packet loss, time 50168ms
rtt min/avg/max/mdev = 0.046/51.390/1024.658/223.283 ms, pipe 4
root@mininet-vm:/home/mininet#
```

Рис. 2.12: Проверка

Восстановим конфигурацию интерфейса по умолчанию на узле h1 (рис. 2.13):

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

Рис. 2.13: Восстановление конфигурации интерфейса по умолчанию

Добавим на интерфейсе узла h1 0,01% повреждения пакетов (рис. 2.14):

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

Рис. 2.14: Добавление на узле h1 0.01% повреждения пакетов

Проверим конфигурацию с помощью инструмента iPerf3 для проверки повторных передач. Для этого запустим iPerf3 в режиме сервера в терминале хоста h2, запустим iPerf3 в клиентском режиме в терминале хоста h1 (рис. 2.15):

```
"host: h1"@mininet-vm
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 56164 connected to 10.0.0.2 port 5201
[ ID] Interval          Transfer      Bitrate      Retr  Cwnd
[ 7]  0.00-1.00    sec   1.65 GBytes  14.2 Gbits/sec    1   3.43 MBytes
[ 7]  1.00-2.00    sec   1.61 GBytes  13.9 Gbits/sec    3   1.19 MBytes
[ 7]  2.00-3.00    sec   1.42 GBytes  12.2 Gbits/sec    0   1.19 MBytes
[ 7]  3.00-4.00    sec    788 MBytes   6.60 Gbits/sec    3    591 KBytes
[ 7]  4.00-5.00    sec   1.08 GBytes   9.31 Gbits/sec    2    609 KBytes
[ 7]  5.00-6.00    sec   1.65 GBytes  14.2 Gbits/sec    0    609 KBytes
[ 7]  6.00-7.00    sec   1.66 GBytes  14.3 Gbits/sec    2    665 KBytes
[ 7]  7.00-8.00    sec   1.67 GBytes  14.4 Gbits/sec    3    621 KBytes
[ 7]  8.00-9.00    sec   1.67 GBytes  14.4 Gbits/sec    4    584 KBytes
[ 7]  9.00-10.00   sec   1.47 GBytes  12.6 Gbits/sec    4    602 KBytes
-----
[ ID] Interval          Transfer      Bitrate      Retr
[ 7]  0.00-10.00   sec   14.7 GBytes  12.6 Gbits/sec   22
[ 7]  0.00-10.00   sec   14.6 GBytes  12.6 Gbits/sec
sender
receiver

"host: h2"@mininet-vm
-----
Accepted connection from 10.0.0.1, port 56162
[ 7] local 10.0.0.2 port 5201 connected to 10.0.0.1 port 56164
[ ID] Interval          Transfer      Bitrate
[ 7]  0.00-1.00    sec   1.63 GBytes  14.0 Gbits/sec
[ 7]  1.00-2.00    sec   1.61 GBytes  13.8 Gbits/sec
[ 7]  2.00-3.01    sec   1.42 GBytes  12.1 Gbits/sec
[ 7]  3.01-4.00    sec    788 MBytes   6.66 Gbits/sec
[ 7]  4.00-5.00    sec   1.08 GBytes   9.29 Gbits/sec
[ 7]  5.00-6.00    sec   1.65 GBytes  14.2 Gbits/sec
[ 7]  6.00-7.00    sec   1.66 GBytes  14.3 Gbits/sec
[ 7]  7.00-8.00    sec   1.67 GBytes  14.4 Gbits/sec
[ 7]  8.00-9.00    sec   1.67 GBytes  14.4 Gbits/sec
[ 7]  9.00-10.00   sec   1.47 GBytes  12.6 Gbits/sec
-----
[ ID] Interval          Transfer      Bitrate
[ 7]  0.00-10.00   sec   14.6 GBytes  12.6 Gbits/sec
receiver
-----
Server listening on 5201
-----
```

Рис. 2.15: Проверка конфигурации с помощью инструмента iPerf3 для проверки повторных передач

Восстановим для узла h1 конфигурацию по умолчанию (рис. 2.16):

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

Рис. 2.16: Восстановление конфигурации интерфейса по умолчанию

Добавим на интерфейсе узла h1 правило из лабораторной работы (рис. 2.17):



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

Рис. 2.17: Добавление на узле h1 правила из лабораторной работы

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%) (рис. 2.18):

```
"host: h1"@mininet-vm
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=10.3 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=11.2 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=10.3 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=10.6 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=10.6 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=10.3 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.061 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.046 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=10.9 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=10.4 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.063 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=10.8 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=10.8 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=10.7 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=10.7 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=10.1 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=0.384 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=10.8 ms

--- 10.0.0.2 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19090ms
rtt min/avg/max/mdev = 0.046/8.530/11.194/4.202 ms
root@mininet-vm:/home/mininet#
```

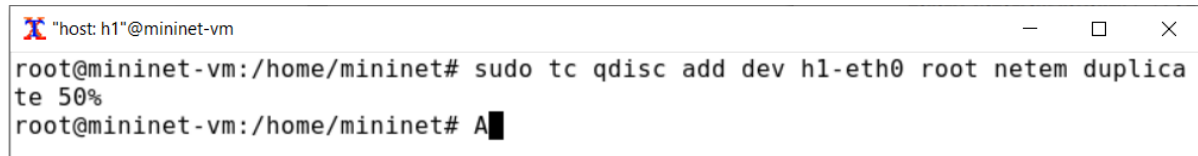
Рис. 2.18: Проверка

Восстановим конфигурацию интерфейса по умолчанию на узле h1 (рис. 2.19):

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

Рис. 2.19: Восстановление конфигурации интерфейса по умолчанию

Для интерфейса узла h1 зададим правило с дублированием 50% пакетов (т.е. 50% пакетов должны быть получены дважды) (рис. 2.20):

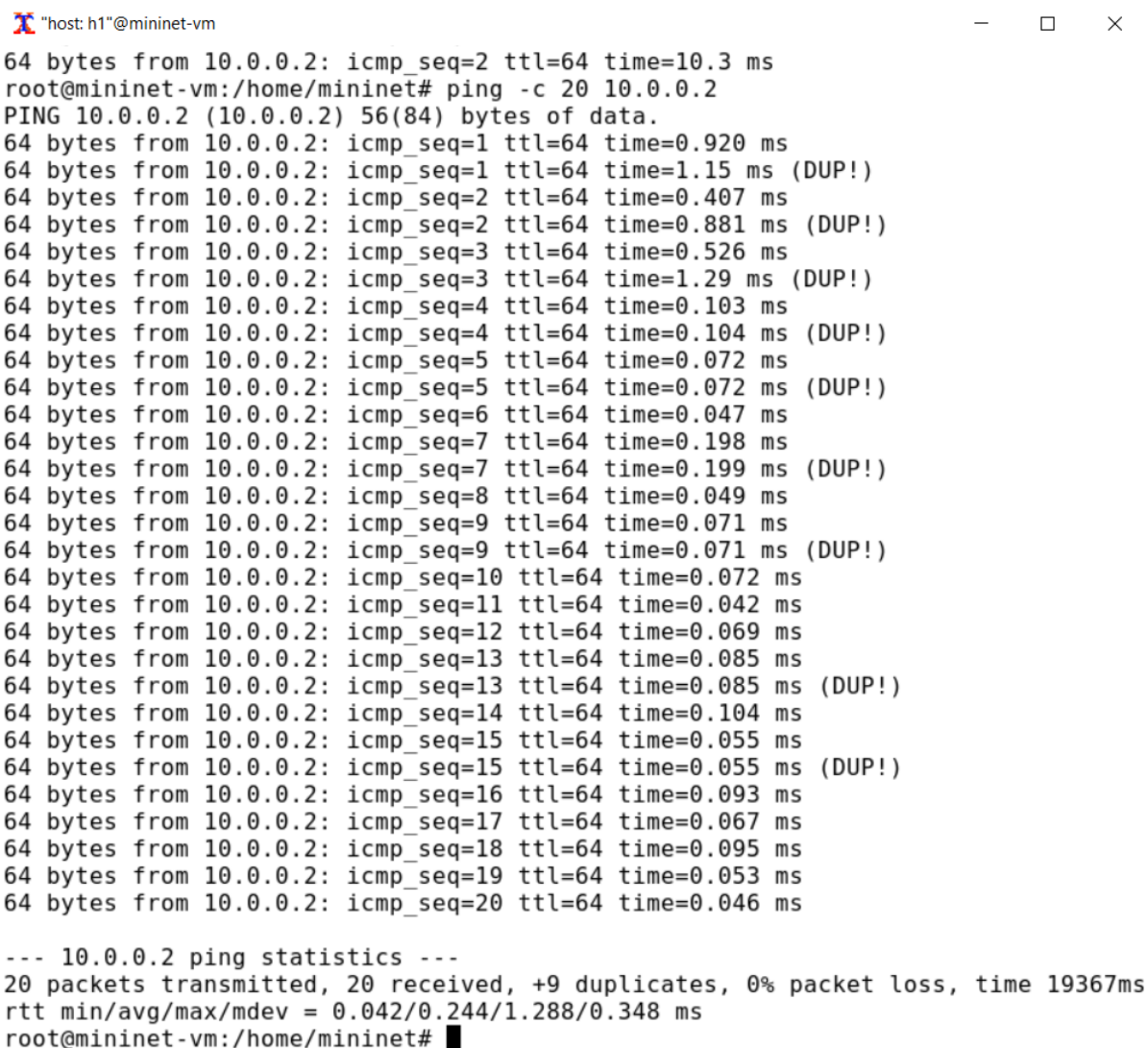


```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem duplicate 50%
root@mininet-vm:/home/mininet# A
```

Рис. 2.20: Добавление на узле h1 правила с дублированием 50% пакетов

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!. Измеренная скорость дублирования пакетов будет приближаться к настроенной скорости по мере выполнения большего количества попыток (рис. 2.21):





```

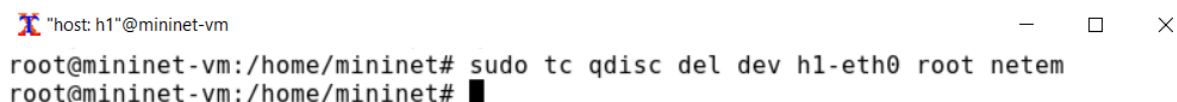
X "host: h1"@mininet-vm
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=10.3 ms
root@mininet-vm:/home/mininet# ping -c 20 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.920 ms
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.15 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.407 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.881 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.526 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=1.29 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.103 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.104 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.072 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.072 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.047 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.198 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.199 ms (DUP!)
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=0.071 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.071 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.072 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.042 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.069 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.085 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.085 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.104 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.055 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.055 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=0.093 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=0.067 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=0.095 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=0.053 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=0.046 ms

--- 10.0.0.2 ping statistics ---
20 packets transmitted, 20 received, +9 duplicates, 0% packet loss, time 19367ms
rtt min/avg/max/mdev = 0.042/0.244/1.288/0.348 ms
root@mininet-vm:/home/mininet#

```

Рис. 2.21: Проверка

Восстановим конфигурацию интерфейса по умолчанию на узле h1 (рис. 2.22):



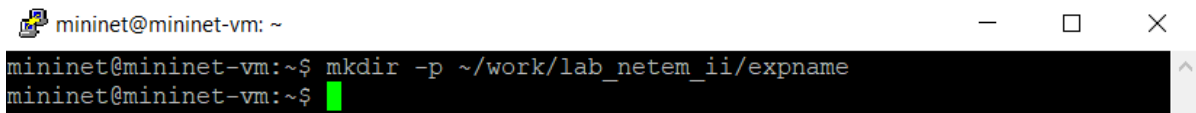
```

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

```

Рис. 2.22: Восстановление конфигурации интерфейса по умолчанию

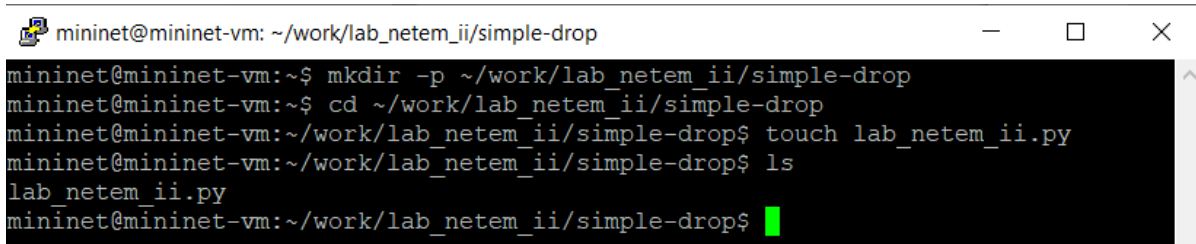
Для каждого воспроизводимого эксперимента exrname создадим свой каталог, в котором будут размещаться файлы эксперимента (рис. 2.23):



```
mininet@mininet-vm: ~  
mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_ii/expname  
mininet@mininet-vm:~$
```

Рис. 2.23: Создание каталога expname

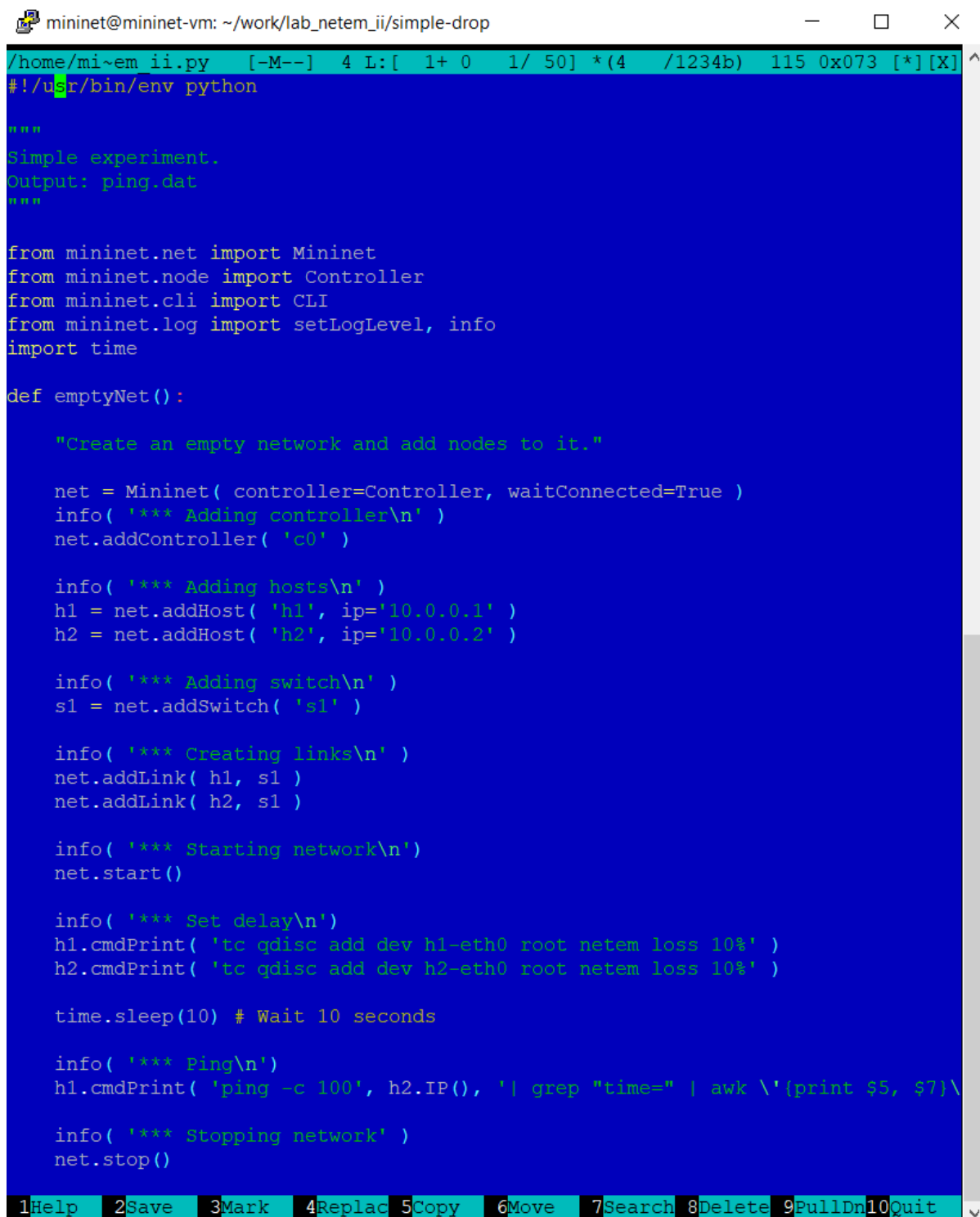
В виртуальной среде mininet в своём рабочем каталоге с проектами создадим каталог simple-drop и перейдём в него (рис. 2.24):



```
mininet@mininet-vm: ~/work/lab_netem_ii/simple-drop  
mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_ii/simple-drop  
mininet@mininet-vm:~$ cd ~/work/lab_netem_ii/simple-drop  
mininet@mininet-vm:~/work/lab_netem_ii/simple-drop$ touch lab_netem_ii.py  
mininet@mininet-vm:~/work/lab_netem_ii/simple-drop$ ls  
lab_netem_ii.py  
mininet@mininet-vm:~/work/lab_netem_ii/simple-drop$
```

Рис. 2.24: Создание каталога simple-drop и дальнейшее его открытие

Создадим скрипт для эксперимента lab\_netem\_ii.py (рис. 2.25):



```
mininet@mininet-vm: ~/work/lab_netem_ii/simple-drop
/home/mininet/.mininet/ii.py [-M--] 4 L:[ 1+ 0 1/ 50] *(4 /1234b) 115 0x073 [*][X] ^
#!/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}\'' )

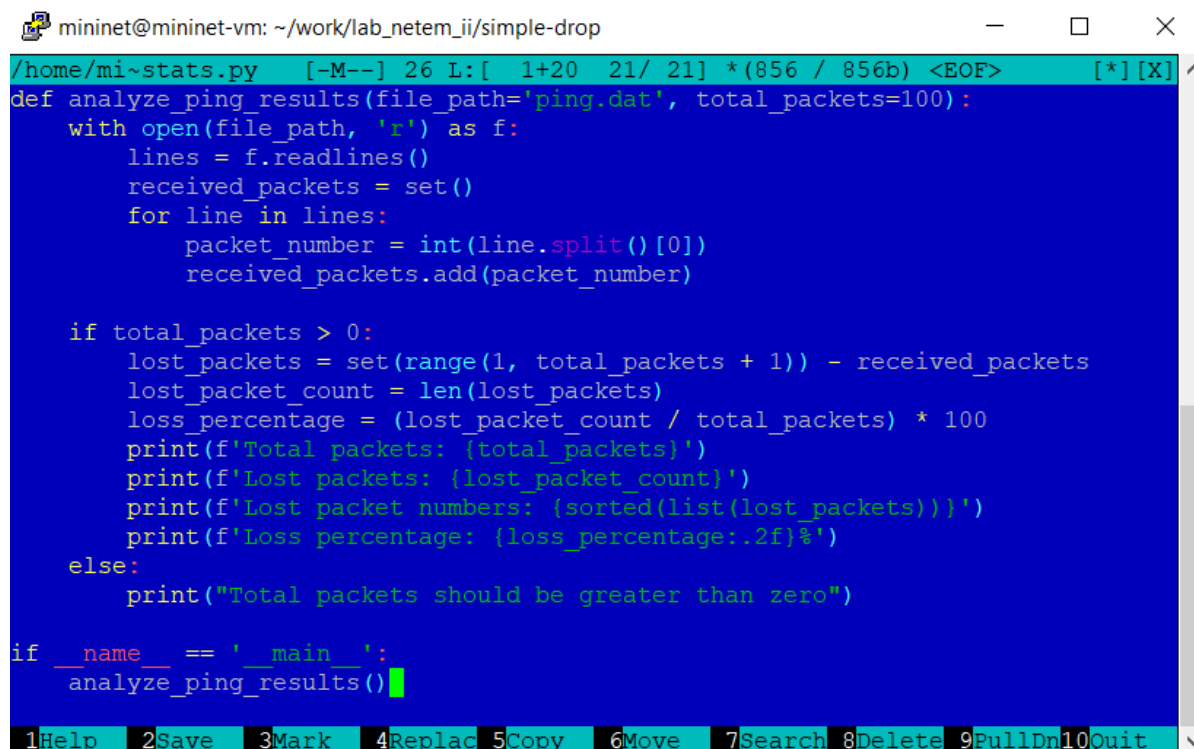
    info( '*** Stopping network\n' )
    net.stop()

1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn10Quit
```

Рис. 2.25: Создание скрипта для эксперимента

Затем создадим скрипт, чтобы на экран или в отдельный файл выводилась

информация о потерях пакетов (рис. 2.26):



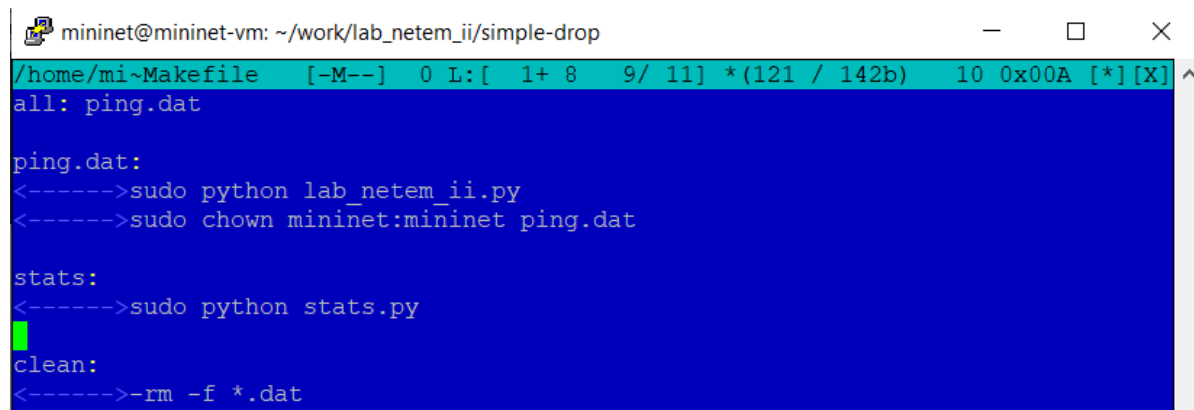
```
mininet@mininet-vm: ~/work/lab_netem_ii/simple-drop
/home/mi~stats.py  [-M--] 26 L:[ 1+20 21/ 21] *(856 / 856b) <EOF>  [*] [X] ^
def analyze_ping_results(file_path='ping.dat', total_packets=100):
    with open(file_path, 'r') as f:
        lines = f.readlines()
        received_packets = set()
        for line in lines:
            packet_number = int(line.split()[0])
            received_packets.add(packet_number)

    if total_packets > 0:
        lost_packets = set(range(1, total_packets + 1)) - received_packets
        lost_packet_count = len(lost_packets)
        loss_percentage = (lost_packet_count / total_packets) * 100
        print(f'Total packets: {total_packets}')
        print(f'Lost packets: {lost_packet_count}')
        print(f'Lost packet numbers: {sorted(list(lost_packets))}')
        print(f'Loss percentage: {loss_percentage:.2f}%')
    else:
        print("Total packets should be greater than zero")

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

Рис. 2.26: Создание нового скрипта для вывода информации о потере пакетов

Создадим Makefile для управления процессом проведения эксперимента (рис. 2.27):



```
mininet@mininet-vm: ~/work/lab_netem_ii/simple-drop
/home/mi~Makefile  [-M--] 0 L:[ 1+ 8 9/ 11] *(121 / 142b) 10 0x00A [*] [X] ^
all: ping.dat

ping.dat:
<----->sudo python lab_netem_ii.py
<----->sudo chown mininet:mininet ping.dat

stats:
<----->sudo python stats.py

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

Рис. 2.27: Создание Makefile и помещение в него скрипта

Выполним эксперимент и далее очистим каталог от результатов проведения экспериментов (рис. 2.28):

```

mininet@mininet-vm: ~/work/lab_netem_ii/simple-drop
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 "time=" | awk \'{print $5, $7}\'} | sed -e \'s/time=//g\' -e \'s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
mininet@mininet-vm:~/work/lab_netem_ii/simple-drop$ make stats
sudo python stats.py
Total packets: 100
Lost packets: 21
Lost packet numbers: [3, 5, 8, 15, 29, 37, 42, 54, 56, 60, 65, 68, 71, 75, 76, 83, 86, 87, 88, 95, 100]
Loss percentage: 21.00%
mininet@mininet-vm:~/work/lab_netem_ii/simple-drop$ make clean
rm -f *.dat
mininet@mininet-vm:~/work/lab_netem_ii/simple-drop$

```

Рис. 2.28: Выполнение эксперимента и последующая очистка каталога

Далее реализуем воспроизводимые эксперименты по исследованию параметров сети, связанных с потерей, изменением порядка и повреждением пакетов при передаче данных (рис. 2.29 - рис. 2.32):

```
mininet@mininet-vm: ~/work/lab_netem_ii/correlation-drop
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%',)
*** 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, $7}\'' |
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
mininet@mininet-vm:~/work/lab_netem_ii/correlation-drop$ make stats
sudo python stats.py
Total packets: 100
Lost packets: 49
Lost packet numbers: [1, 2, 7, 8, 12, 16, 19, 20, 21, 22, 23, 24, 26, 28, 29, 37,
95, 96, 97, 98, 99, 100]
Loss percentage: 49.00%
mininet@mininet-vm:~/work/lab_netem_ii/correlation-drop$
```

Рис. 2.29: Реализация воспроизводимого эксперимента по исследованию параметров сети

```
mininet@mininet-vm: ~/work/lab_netem_ii/package-damage
mininet@mininet-vm:~/work/lab_netem_ii/package-damage$ 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 corrupt 0.01%,')
*** 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, $7}\'' |
sed -e \'/s/time=//g\' -e \'/s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
mininet@mininet-vm:~/work/lab_netem_ii/package-damage$ make stats
sudo python stats.py
Total packets: 100
Lost packets: 8
Lost packet numbers: [15, 17, 18, 26, 38, 82, 85, 95]
Loss percentage: 8.00%
mininet@mininet-vm:~/work/lab_netem_ii/package-damage$
```

Рис. 2.30: Реализация воспроизводимого эксперимента по исследованию параметров сети

```
mininet@mininet-vm: ~/work/lab_netem_ii/reordering-packages
mininet@mininet-vm:~/work/lab_netem_ii/reordering-packages$ 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% 50%,)
*** 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, $7}\'' |
sed -e \'/s/time=//g\' -e \'/s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
mininet@mininet-vm:~/work/lab_netem_ii/reordering-packages$ make stats
sudo python stats.py
Total packets: 100
Lost packets: 11
Lost packet numbers: [6, 27, 30, 31, 34, 52, 65, 68, 84, 89, 96]
Loss percentage: 11.00%
mininet@mininet-vm:~/work/lab_netem_ii/reordering-packages$
```

Рис. 2.31: Реализация воспроизводимого эксперимента по исследованию параметров сети



```
mininet@mininet-vm: ~/work/lab_netem_ii/duplicating-packages
mininet@mininet-vm:~/work/lab_netem_ii/duplicating-packages$ 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 duplicate 50%,')
*** h2 : ('tc qdisc add dev h2-eth0 root netem',)
*** Ping
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "time=" | awk \'{print $5, $7}\'' |
sed -e \'/s/time=//g\' -e \'/s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
mininet@mininet-vm:~/work/lab_netem_ii/duplicating-packages$ make stats
sudo python stats.py
Total packets: 100
Lost packets: 0
Lost packet numbers: []
Loss percentage: 0.00%
mininet@mininet-vm:~/work/lab_netem_ii/duplicating-packages$
```

Рис. 2.32: Реализация воспроизводимого эксперимента по исследованию параметров сети

## 3 Вывод

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

## 4 Список литературы. Библиография

[1] Mininet: <https://mininet.org/>