

Отчёт по лабораторной работе №5

Моделирование сетей передачи данных

Эмуляция и измерение потерь пакетов в глобальных сетях

Выполнил: Исаев Булат Абубакарович,
НПИбд-01-22, 1132227131

Содержание

1	Цель работы	1
2	Выполнение лабораторной работы.....	1
3	Вывод.....	22
4	Список литературы. Библиография	22

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. 1):

```
mininet@mininet-vm: ~  
login as: mininet  
mininet@192.168.56.103's password:  
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-42-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
New release '22.04.5 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Fri Oct 31 15:38:26 2025 from 192.168.56.1  
mininet@mininet-vm:~$ xauth list $DISPLAY  
mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  bced047e0c3bcbb01e84e95f729f6f92  
mininet@mininet-vm:~$ xauth add mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  bced047e0c3bcbb01e84e95f729f6f92  
mininet@mininet-vm:~$ sudo -i  
root@mininet-vm:~# xauth add mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  bced047e0c3bcbb01e84e95f729f6f92  
root@mininet-vm:~# logout  
mininet@mininet-vm:~$
```

Рис. 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):

```
"host: h1"@mininet-vm
root@mininet-vm:/home/mininet#

mininet@mininet-vm: ~
mininet@mininet-vm:~$ xauth add mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 bced047e0c3bcbb01e84e95f729f6f92
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth add mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 bced047e0c3bcbb01e84e95f729f6f92
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:
mininet>
root@mininet-vm:/home/mininet#
```

Рис. 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. 3):

```
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 d6:0d:6a:44:d6:54 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 1091 bytes 248124 (248.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1091 bytes 248124 (248.1 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 netmask 255.0.0.0 broadcast 10.255.255.255
    ether da:10:ad:8c:81:0e 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 1175 bytes 256484 (256.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1175 bytes 256484 (256.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

Рис. 3: Отображение информации их сетевых интерфейсов и IP-адресов

Проверим подключение между хостами h1 и h2 с помощью команды ping с параметром -с 6 (рис. 4):

```
"host: h1"@mininet-vm
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 1091  bytes 248124 (248.1 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 1091  bytes 248124 (248.1 KB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

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=3.57 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.333 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.163 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.078 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.081 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.135 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5102ms
rtt min/avg/max/mdev = 0.078/0.726/3.566/1.272 ms
root@mininet-vm:/home/mininet#

"host: h2"@mininet-vm
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 1175  bytes 256484 (256.4 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 1175  bytes 256484 (256.4 KB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

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=2.76 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.125 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.078 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.198 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.081 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.092 ms

--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5082ms
rtt min/avg/max/mdev = 0.078/0.556/2.764/0.988 ms
root@mininet-vm:/home/mininet#
```

Рис. 4: Проверка подключения между хостами h1 и h2

На хосте h1 добавим 10% потерь пакетов к интерфейсу h1-eth0 (рис. 5):

```
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 1091 bytes 248124 (248.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1091 bytes 248124 (248.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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=3.57 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.333 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.163 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.078 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.081 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.135 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5102ms
rtt min/avg/max/mdev = 0.078/0.726/3.566/1.272 ms
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem loss 10%
root@mininet-vm:/home/mininet#
```

Рис. 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. 6):

```
64 bytes from 10.0.0.2: icmp_seq=81 ttl=64 time=0.102 ms
64 bytes from 10.0.0.2: icmp_seq=82 ttl=64 time=0.081 ms
64 bytes from 10.0.0.2: icmp_seq=83 ttl=64 time=0.088 ms
64 bytes from 10.0.0.2: icmp_seq=84 ttl=64 time=0.105 ms
64 bytes from 10.0.0.2: icmp_seq=86 ttl=64 time=0.115 ms
64 bytes from 10.0.0.2: icmp_seq=87 ttl=64 time=0.075 ms
64 bytes from 10.0.0.2: icmp_seq=88 ttl=64 time=0.090 ms
64 bytes from 10.0.0.2: icmp_seq=89 ttl=64 time=0.081 ms
64 bytes from 10.0.0.2: icmp_seq=90 ttl=64 time=0.084 ms
64 bytes from 10.0.0.2: icmp_seq=91 ttl=64 time=0.095 ms
64 bytes from 10.0.0.2: icmp_seq=92 ttl=64 time=0.070 ms
64 bytes from 10.0.0.2: icmp_seq=93 ttl=64 time=0.219 ms
64 bytes from 10.0.0.2: icmp_seq=94 ttl=64 time=0.070 ms
64 bytes from 10.0.0.2: icmp_seq=95 ttl=64 time=0.071 ms
64 bytes from 10.0.0.2: icmp_seq=96 ttl=64 time=0.078 ms
64 bytes from 10.0.0.2: icmp_seq=97 ttl=64 time=0.072 ms
64 bytes from 10.0.0.2: icmp_seq=98 ttl=64 time=0.085 ms
64 bytes from 10.0.0.2: icmp_seq=99 ttl=64 time=0.086 ms
64 bytes from 10.0.0.2: icmp_seq=100 ttl=64 time=0.119 ms

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 90 received, 10% packet loss, time 101342ms
rtt min/avg/max/mdev = 0.047/0.098/0.298/0.041 ms
root@mininet-vm:/home/mininet#
```

Рис. 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. 7):

```
"host: h2"@mininet-vm
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 1175  bytes 256484 (256.4 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 1175  bytes 256484 (256.4 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

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=2.76 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.125 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.078 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.198 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.081 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.092 ms

--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5082ms
rtt min/avg/max/mdev = 0.078/0.556/2.764/0.988 ms
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h2-eth0 root netem loss 10%
root@mininet-vm:/home/mininet#
```

Рис. 7: Добавление 10% потерь пакетов на хосте h2

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 (рис. 8):


```
"host: h1"@mininet-vm
64 bytes from 10.0.0.2: icmp_seq=76 ttl=64 time=0.212 ms
64 bytes from 10.0.0.2: icmp_seq=77 ttl=64 time=0.080 ms
64 bytes from 10.0.0.2: icmp_seq=78 ttl=64 time=0.132 ms
64 bytes from 10.0.0.2: icmp_seq=81 ttl=64 time=0.826 ms
64 bytes from 10.0.0.2: icmp_seq=82 ttl=64 time=0.139 ms
64 bytes from 10.0.0.2: icmp_seq=84 ttl=64 time=0.081 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=87 ttl=64 time=0.077 ms
64 bytes from 10.0.0.2: icmp_seq=88 ttl=64 time=0.093 ms
64 bytes from 10.0.0.2: icmp_seq=90 ttl=64 time=0.123 ms
64 bytes from 10.0.0.2: icmp_seq=91 ttl=64 time=0.113 ms
64 bytes from 10.0.0.2: icmp_seq=92 ttl=64 time=0.637 ms
64 bytes from 10.0.0.2: icmp_seq=93 ttl=64 time=0.077 ms
64 bytes from 10.0.0.2: icmp_seq=94 ttl=64 time=0.078 ms
64 bytes from 10.0.0.2: icmp_seq=95 ttl=64 time=0.060 ms
64 bytes from 10.0.0.2: icmp_seq=96 ttl=64 time=0.102 ms
64 bytes from 10.0.0.2: icmp_seq=97 ttl=64 time=0.076 ms
64 bytes from 10.0.0.2: icmp_seq=98 ttl=64 time=0.115 ms
64 bytes from 10.0.0.2: icmp_seq=100 ttl=64 time=0.082 ms

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 81 received, 19% packet loss, time 101337ms
rtt min/avg/max/mdev = 0.047/0.202/5.940/0.661 ms
root@mininet-vm:/home/mininet#
```

Рис. 8: Проверка

Восстановим конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса (рис. 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. 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. 10):


```
"host: h1"@mininet-vm

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 81 received, 19% packet loss, time 101337ms
rtt min/avg/max/mdev = 0.047/0.202/5.940/0.661 ms
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
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=2.75 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=1.01 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.512 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.074 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.097 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.076 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5076ms
rtt min/avg/max/mdev = 0.074/0.753/2.754/0.955 ms
root@mininet-vm:/home/mininet#
```

Рис. 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. 11):

```
"host: h1"@mininet-vm

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 81 received, 19% packet loss, time 101337ms
rtt min/avg/max/mdev = 0.047/0.202/5.940/0.661 ms
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
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=2.75 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=1.01 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.512 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.074 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.097 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.076 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5076ms
rtt min/avg/max/mdev = 0.074/0.753/2.754/0.955 ms
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem loss 50% 50%
root@mininet-vm:/home/mininet#
```

Рис. 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. 12):

```
"host: h1"@mininet-vm
root@mininet-vm:/home/mininet# ping -c 50 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=1.26 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=1.21 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.291 ms
64 bytes from 10.0.0.2: icmp_seq=31 ttl=64 time=2.45 ms
64 bytes from 10.0.0.2: icmp_seq=32 ttl=64 time=0.207 ms
64 bytes from 10.0.0.2: icmp_seq=33 ttl=64 time=0.129 ms
64 bytes from 10.0.0.2: icmp_seq=34 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=36 ttl=64 time=0.094 ms
64 bytes from 10.0.0.2: icmp_seq=37 ttl=64 time=0.116 ms
64 bytes from 10.0.0.2: icmp_seq=38 ttl=64 time=0.125 ms
64 bytes from 10.0.0.2: icmp_seq=39 ttl=64 time=0.092 ms
64 bytes from 10.0.0.2: icmp_seq=40 ttl=64 time=0.103 ms
64 bytes from 10.0.0.2: icmp_seq=41 ttl=64 time=0.102 ms
64 bytes from 10.0.0.2: icmp_seq=42 ttl=64 time=0.135 ms
64 bytes from 10.0.0.2: icmp_seq=44 ttl=64 time=0.135 ms
64 bytes from 10.0.0.2: icmp_seq=46 ttl=64 time=0.097 ms
64 bytes from 10.0.0.2: icmp_seq=47 ttl=64 time=0.146 ms
64 bytes from 10.0.0.2: icmp_seq=49 ttl=64 time=0.101 ms

--- 10.0.0.2 ping statistics ---
50 packets transmitted, 18 received, 64% packet loss, time 50008ms
rtt min/avg/max/mdev = 0.043/0.379/2.453/0.612 ms
root@mininet-vm:/home/mininet#
```

Рис. 12: Проверка

Восстановим конфигурацию интерфейса по умолчанию на узле h1 (рис. 13):

```
"host: h1"@mininet-vm
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=1.26 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=1.21 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.291 ms
64 bytes from 10.0.0.2: icmp_seq=31 ttl=64 time=2.45 ms
64 bytes from 10.0.0.2: icmp_seq=32 ttl=64 time=0.207 ms
64 bytes from 10.0.0.2: icmp_seq=33 ttl=64 time=0.129 ms
64 bytes from 10.0.0.2: icmp_seq=34 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=36 ttl=64 time=0.094 ms
64 bytes from 10.0.0.2: icmp_seq=37 ttl=64 time=0.116 ms
64 bytes from 10.0.0.2: icmp_seq=38 ttl=64 time=0.125 ms
64 bytes from 10.0.0.2: icmp_seq=39 ttl=64 time=0.092 ms
64 bytes from 10.0.0.2: icmp_seq=40 ttl=64 time=0.103 ms
64 bytes from 10.0.0.2: icmp_seq=41 ttl=64 time=0.102 ms
64 bytes from 10.0.0.2: icmp_seq=42 ttl=64 time=0.135 ms
64 bytes from 10.0.0.2: icmp_seq=44 ttl=64 time=0.135 ms
64 bytes from 10.0.0.2: icmp_seq=46 ttl=64 time=0.097 ms
64 bytes from 10.0.0.2: icmp_seq=47 ttl=64 time=0.146 ms
64 bytes from 10.0.0.2: icmp_seq=49 ttl=64 time=0.101 ms

--- 10.0.0.2 ping statistics ---
50 packets transmitted, 18 received, 64% packet loss, time 50008ms
rtt min/avg/max/mdev = 0.043/0.379/2.453/0.612 ms
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
root@mininet-vm:/home/mininet#
```

Рис. 13: Восстановление конфигурации интерфейса по умолчанию

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

```

"host: h1"@mininet-vm
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=1.26 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=1.21 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.291 ms
64 bytes from 10.0.0.2: icmp_seq=31 ttl=64 time=2.45 ms
64 bytes from 10.0.0.2: icmp_seq=32 ttl=64 time=0.207 ms
64 bytes from 10.0.0.2: icmp_seq=33 ttl=64 time=0.129 ms
64 bytes from 10.0.0.2: icmp_seq=34 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=36 ttl=64 time=0.094 ms
64 bytes from 10.0.0.2: icmp_seq=37 ttl=64 time=0.116 ms
64 bytes from 10.0.0.2: icmp_seq=38 ttl=64 time=0.125 ms
64 bytes from 10.0.0.2: icmp_seq=39 ttl=64 time=0.092 ms
64 bytes from 10.0.0.2: icmp_seq=40 ttl=64 time=0.103 ms
64 bytes from 10.0.0.2: icmp_seq=41 ttl=64 time=0.102 ms
64 bytes from 10.0.0.2: icmp_seq=42 ttl=64 time=0.135 ms
64 bytes from 10.0.0.2: icmp_seq=44 ttl=64 time=0.135 ms
64 bytes from 10.0.0.2: icmp_seq=46 ttl=64 time=0.097 ms
64 bytes from 10.0.0.2: icmp_seq=47 ttl=64 time=0.146 ms
64 bytes from 10.0.0.2: icmp_seq=49 ttl=64 time=0.101 ms

--- 10.0.0.2 ping statistics ---
50 packets transmitted, 18 received, 64% packet loss, time 50008ms
rtt min/avg/max/mdev = 0.043/0.379/2.453/0.612 ms
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem corrupt 0.01%

```

Рис. 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. 15):

```

"host: h1"@mininet-vm
Cannot find device "h2-eth0"
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 41988 connected to 10.0.0.2 port 5201
[ ID] Interval           Transfer    Bitrate        Retr   Cwnd
[ 7]  0.00-1.01      sec  1.25 GBytes  10.7 Gbits/sec    1   5.79 MBytes
[ 7]  1.01-2.01      sec  1.33 GBytes  11.4 Gbits/sec    2   2.93 MBytes
[ 7]  2.01-3.00      sec  1.26 GBytes  10.9 Gbits/sec    1   2.32 MBytes
[ 7]  3.00-4.00      sec  1.25 GBytes  10.7 Gbits/sec    1   2.00 MBytes
[ 7]  4.00-5.00      sec  1.40 GBytes  12.1 Gbits/sec    1   1.67 MBytes
[ 7]  5.00-6.00      sec  1.28 GBytes  10.9 Gbits/sec    3   1.09 MBytes
[ 7]  6.00-7.00      sec  1.38 GBytes  11.8 Gbits/sec    2   1.20 MBytes
[ 7]  7.00-8.00      sec  1.23 GBytes  10.6 Gbits/sec    3   2.20 MBytes
[ 7]  8.00-9.01      sec  1.40 GBytes  11.9 Gbits/sec    5     894 KBytes
[ 7]  9.01-10.00     sec  1.54 GBytes  13.3 Gbits/sec    2   1.44 MBytes
- - - - -
[ ID] Interval           Transfer    Bitrate        Retr
[ 7]  0.00-10.00      sec  13.3 GBytes  11.4 Gbits/sec    21
[ 7]  0.00-10.01      sec  13.3 GBytes  11.4 Gbits/sec
sender
receiver

iperf Done.
root@mininet-vm:/home/mininet#

```

Рис. 15: Проверка конфигурации с помощью инструмента iPerf3 для проверки повторных передач

Восстановим для узла h1 конфигурацию по умолчанию (рис. 16):

```
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 41988 connected to 10.0.0.2 port 5201
[ ID] Interval            Transfer      Bitrate      Retr  Cwnd
[ 7]  0.00-1.01      sec  1.25 GBytes  10.7 Gbits/sec    1   5.79 MBytes
[ 7]  1.01-2.01      sec  1.33 GBytes  11.4 Gbits/sec    2   2.93 MBytes
[ 7]  2.01-3.00      sec  1.26 GBytes  10.9 Gbits/sec    1   2.32 MBytes
[ 7]  3.00-4.00      sec  1.25 GBytes  10.7 Gbits/sec    1   2.00 MBytes
[ 7]  4.00-5.00      sec  1.40 GBytes  12.1 Gbits/sec    1   1.67 MBytes
[ 7]  5.00-6.00      sec  1.28 GBytes  10.9 Gbits/sec    3   1.09 MBytes
[ 7]  6.00-7.00      sec  1.38 GBytes  11.8 Gbits/sec    2   1.20 MBytes
[ 7]  7.00-8.00      sec  1.23 GBytes  10.6 Gbits/sec    3   2.20 MBytes
[ 7]  8.00-9.01      sec  1.40 GBytes  11.9 Gbits/sec    5    894 KBytes
[ 7]  9.01-10.00     sec  1.54 GBytes  13.3 Gbits/sec    2   1.44 MBytes
- - - - -
[ ID] Interval            Transfer      Bitrate      Retr
[ 7]  0.00-10.00     sec  13.3 GBytes  11.4 Gbits/sec    21
[ 7]  0.00-10.01     sec  13.3 GBytes  11.4 Gbits/sec
sender
receiver

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

Рис. 16: Восстановление конфигурации интерфейса по умолчанию

Добавим на интерфейсе узла h1 правило из лабораторной работы (рис. 17):

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 1
0ms reorder 25% 50%
root@mininet-vm:/home/mininet#
```

Рис. 17: Добавление на узле h1 правила из лабораторной работы

Проверим, что на соединении от хоста h1 к хосту h2 имеются потери пакетов, используя команду ring 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. 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. 18: Проверка

Восстановим конфигурацию интерфейса по умолчанию на узле h1 (рис. 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. 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. 20):


```
host: h1" @mininet-vm
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.033 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.086 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.055 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.051 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.056 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.080 ms
64 bytes from 10.0.0.1: icmp_seq=7 ttl=64 time=0.077 ms
64 bytes from 10.0.0.1: icmp_seq=8 ttl=64 time=0.064 ms
64 bytes from 10.0.0.1: icmp_seq=9 ttl=64 time=0.095 ms
64 bytes from 10.0.0.1: icmp_seq=10 ttl=64 time=0.065 ms
64 bytes from 10.0.0.1: icmp_seq=11 ttl=64 time=0.091 ms
64 bytes from 10.0.0.1: icmp_seq=12 ttl=64 time=0.060 ms
64 bytes from 10.0.0.1: icmp_seq=13 ttl=64 time=0.091 ms
64 bytes from 10.0.0.1: icmp_seq=14 ttl=64 time=0.068 ms
64 bytes from 10.0.0.1: icmp_seq=15 ttl=64 time=0.055 ms
64 bytes from 10.0.0.1: icmp_seq=16 ttl=64 time=0.644 ms
64 bytes from 10.0.0.1: icmp_seq=17 ttl=64 time=0.112 ms
64 bytes from 10.0.0.1: icmp_seq=18 ttl=64 time=0.087 ms
64 bytes from 10.0.0.1: icmp_seq=19 ttl=64 time=0.071 ms
64 bytes from 10.0.0.1: icmp_seq=20 ttl=64 time=0.090 ms

--- 10.0.0.1 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19436ms
rtt min/avg/max/mdev = 0.033/0.101/0.644/0.125 ms
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem duplicate 50%
root@mininet-vm:/home/mininet#
```

Рис. 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. 21):

```
host: h1@mininet-vm
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=2.67 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.845 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.920 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.328 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.350 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.071 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.113 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.114 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.066 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.067 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.100 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.087 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.099 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.099 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.082 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.236 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.237 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.107 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.108 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.109 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.105 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.106 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.065 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=0.132 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=0.098 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=0.119 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=0.120 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=0.090 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=0.091 ms (DUP!)
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=0.090 ms

--- 10.0.0.2 ping statistics ---
20 packets transmitted, 20 received, +10 duplicates, 0% packet loss, time 19398ms
rtt min/avg/max/mdev = 0.065/0.260/2.670/0.490 ms
root@mininet-vm:/home/mininet#
```

Рис. 21: Проверка

Восстановим конфигурацию интерфейса по умолчанию на узле h1 (рис. 22):

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

Рис. 22: Восстановление конфигурации интерфейса по умолчанию

Для каждого воспроизводимого эксперимента exрname создадим свой каталог, в котором будут размещаться файлы эксперимента (рис. 23):

```
mininet@mininet-vm:~$ mkdir -p ~/work/lab netem ii/exрname
mininet@mininet-vm:~$
```

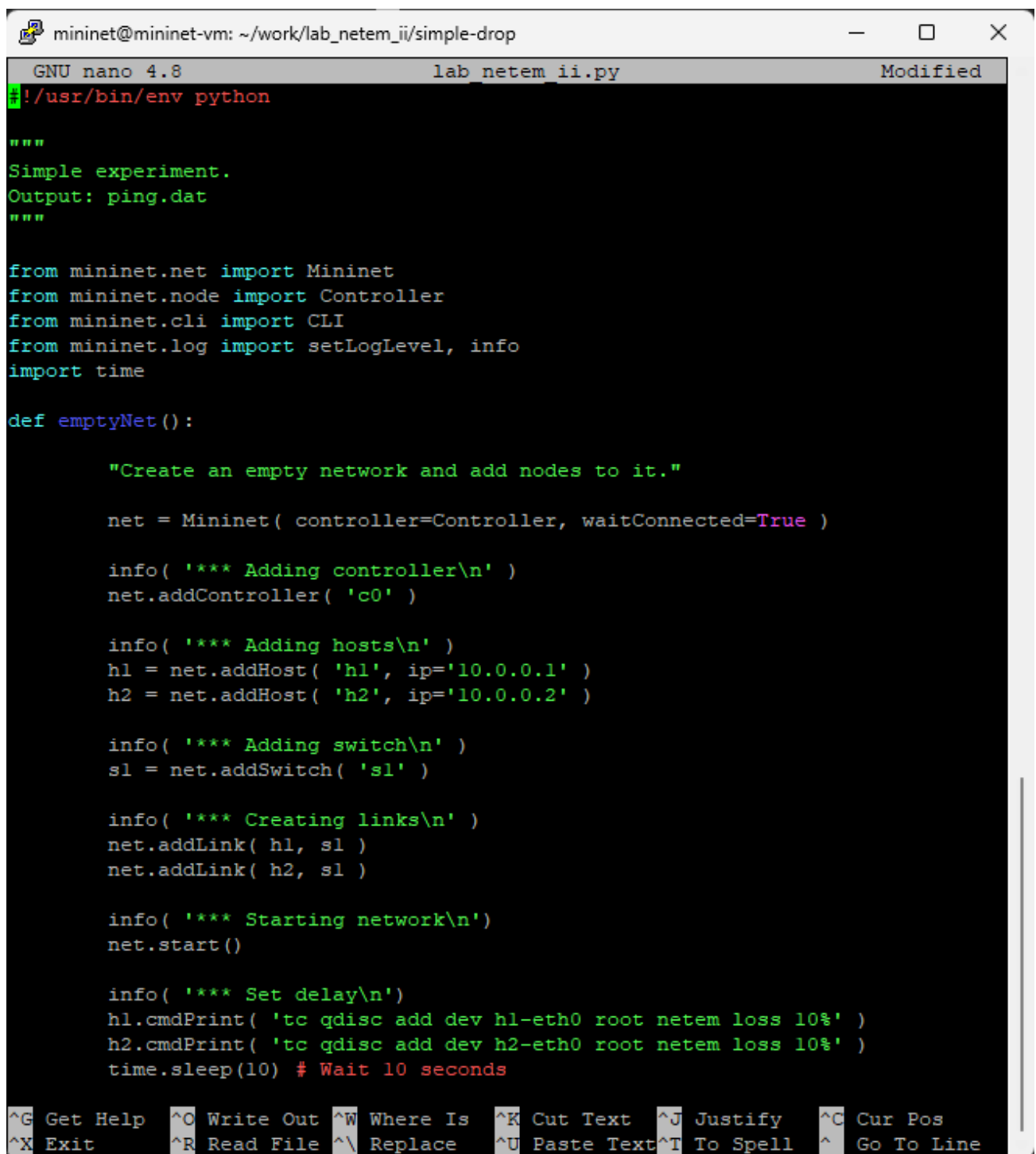
Рис. 23: Создание каталога exрname

В виртуальной среде mininet в своём рабочем каталоге с проектами создадим каталог simple-drop и перейдём в него (рис. 24):


```
mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_ii/simple-drop
mininet@mininet-vm:~$ cd ^C
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. 24: Создание каталога simple-drop и дальнейшее его открытие

Создадим скрипт для эксперимента lab_netem_ii.py (рис. 25):



The image shows a terminal window titled "mininet@mininet-vm: ~/work/lab_netem_ii/simple-drop". Inside the terminal, the GNU nano 4.8 editor is open, editing a file named "lab_netem_ii.py". The script content is as follows:

```
#!/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
```

The bottom of the terminal shows the nano editor's command shortcuts:

^G Get Help	^O Write Out	^W Where Is	^K Cut Text	^J Justify	^C Cur Pos
^X Exit	^R Read File	^_ Replace	^U Paste Text	^T To Spell	^_ Go To Line

Рис. 25: Создание скрипта для эксперимента

Затем создадим скрипт, чтобы на экран или в отдельный файл выводилась информация о потерях пакетов (рис. 26):

```
mininet@mininet-vm: ~/work/lab_netem_ii/simple-drop
GNU nano 4.8 stats.py Modified
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. 26: Создание нового скрипта для вывода информации о потере пакетов

Создадим Makefile для управления процессом проведения эксперимента (рис. 27):

```
mininet@mininet-vm: ~/work/lab_netem_ii/simple-drop
GNU nano 4.8
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
```

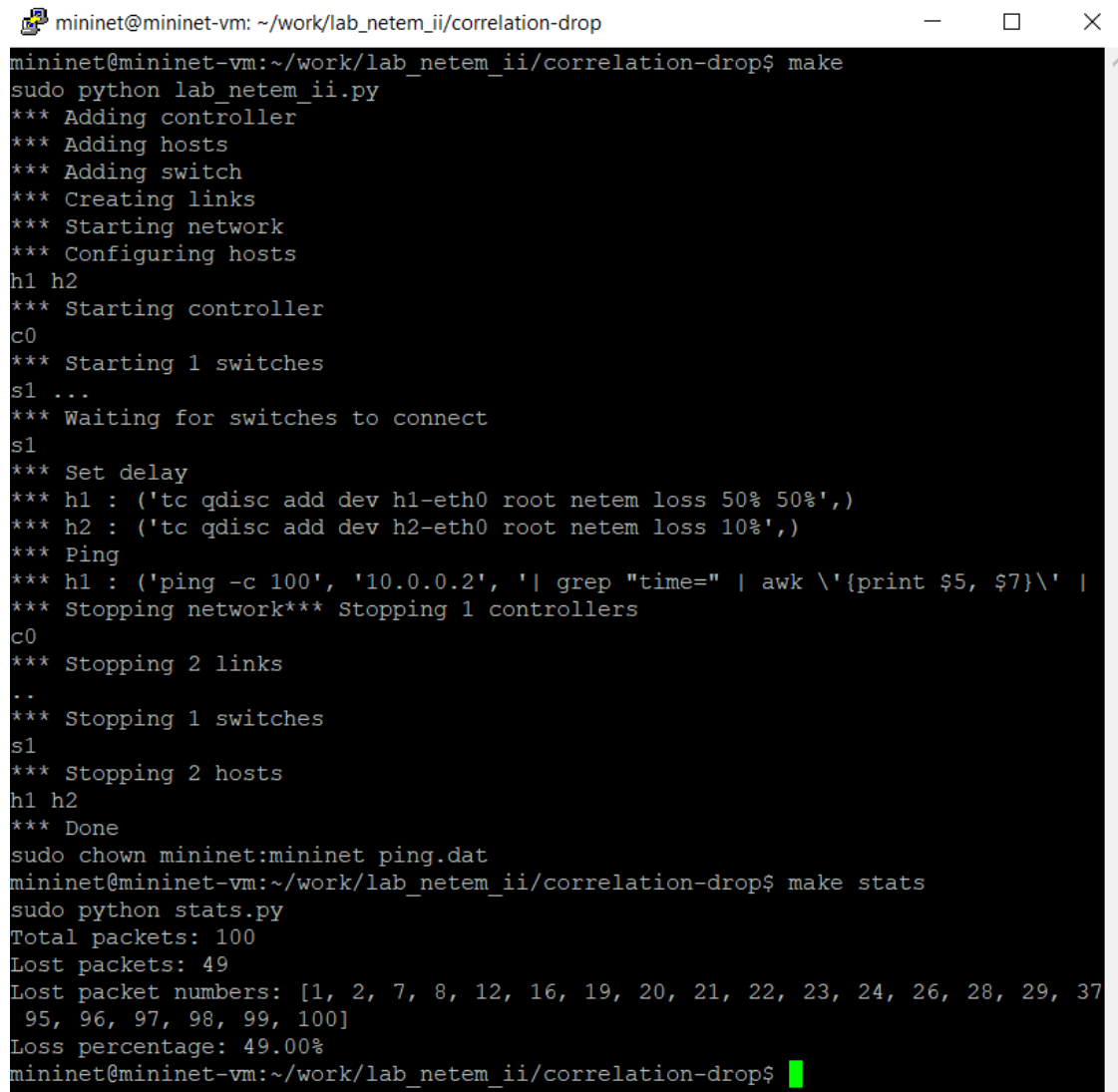
Рис. 27: Создание Makefile и помещение в него скрипта

Выполним эксперимент и далее очистим каталог от результатов проведения экспериментов (рис. 28):

```
mininet@mininet-vm:~/work/lab_netem_ii/simple-drop$ make
make: Nothing to be done for 'all'.
mininet@mininet-vm:~/work/lab_netem_ii/simple-drop$ make stats
sudo python stats.py
\Total packets: (total_packets)
Lost packets: (lost_packet_count)
Lost packet numbers: [21, 23, 25, 26, 28, 49, 50, 51, 56, 60, 64, 80, 81, 83, 87, 91, 100]
Loss percentage: (loss_percentage:.2f)%
mininet@mininet-vm:~/work/lab_netem_ii/simple-drop$
```

Рис. 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. 29 - рис. 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. 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. 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. 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. 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/>