

Отчёт по лабораторной работе №4

Моделирование сетей передачи данных

Эмуляция и измерение задержек в глобальных сетях

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

Содержание

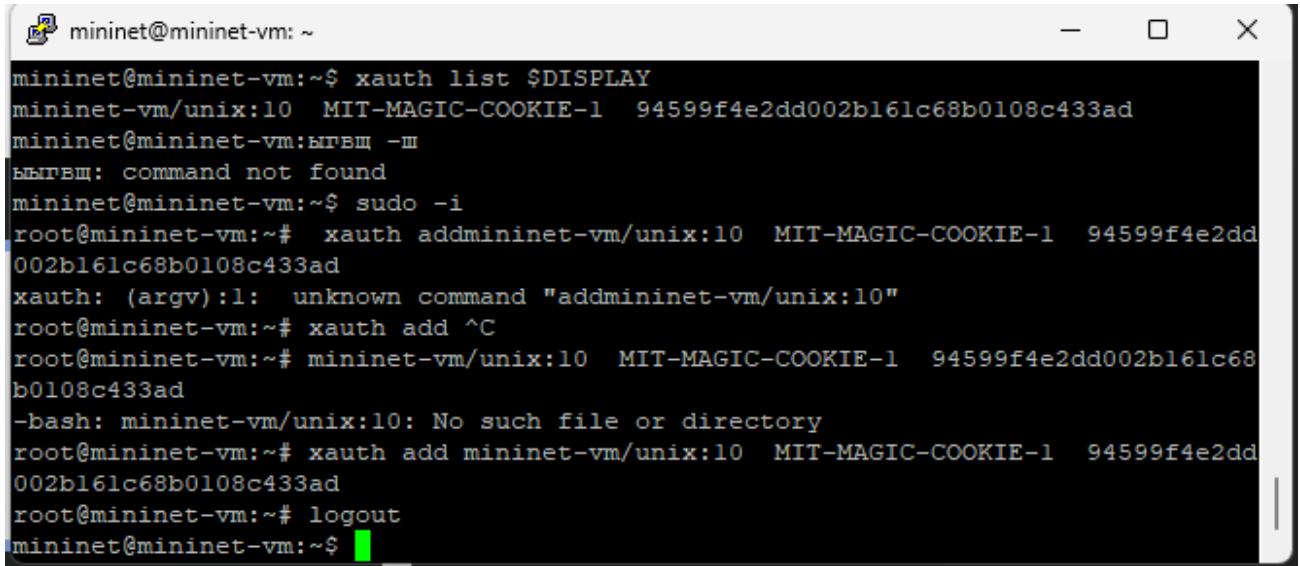
1	Цель работы	1
2	Выполнение лабораторной работы.....	1
3	Вывод.....	33
4	Список литературы. Библиография	33

1 Цель работы

Основной целью работы является знакомство с NETEM — инструментом для тестирования производительности приложений в виртуальной сети, а также получение навыков проведения интерактивного и воспроизводимого экспериментов по измерению задержки и её дрожания (jitter) в моделируемой сети в среде Mininet.

2 Выполнение лабораторной работы

В виртуальной машине mininet исправим права запуска X-соединения (рис. 1):



```
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  94599f4e2dd002b161c68b0108c433ad
mininet@mininet-vm:~$ xauth addmininet-vm/unix:10  MIT-MAGIC-COOKIE-1  94599f4e2dd002b161c68b0108c433ad
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth addmininet-vm/unix:10  MIT-MAGIC-COOKIE-1  94599f4e2dd002b161c68b0108c433ad
xauth: (argv):1: unknown command "addmininet-vm/unix:10"
root@mininet-vm:~# xauth add ^C
root@mininet-vm:~# mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  94599f4e2dd002b161c68b0108c433ad
-bash: mininet-vm/unix:10: No such file or directory
root@mininet-vm:~# xauth add mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  94599f4e2dd002b161c68b0108c433ad
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):

```
"controller: c0" (root)@mininet-vm
root@mininet-vm:/home/mininet# 
"host: h2" (root)@mininet-vm
root@mininet-vm:/home/mininet# 
"host: h1" (root)@mininet-vm
root@mininet-vm:/home/mininet# 

mininet@mininet-vm: ~
*** 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: 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):

The image shows two terminal windows side-by-side. Both windows have a title bar with the host name and are running the command `ifconfig`.

Terminal h1:

```
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 7a:85:08:39:57:be 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 713 bytes 252652 (252.6 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 713 bytes 252652 (252.6 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

Terminal h2:

```
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 fa:e6:a6:4b:f6:b6 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 1069 bytes 272276 (272.2 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 1069 bytes 272276 (272.2 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

Рис. 3: Отображение информации о сетевых интерфейсах и IP-адресов

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

The screenshot shows two terminal windows side-by-side. Both are titled "host h1" and "host h2" at the top, indicating they are running on the same host machine (mininet-vm). Each window displays the output of a ping command to the other host.

```

X "host h1"@mininet-vm
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=0.034 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.057 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.048 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.034 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.049 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.042 ms

--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5127ms
rtt min/avg/max/mdev = 0.034/0.044/0.057/0.008 ms
root@mininet-vm:/home/mininet# 

X "host h2"@mininet-vm
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=0.031 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.029 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.031 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.053 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.048 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.025 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5120ms
rtt min/avg/max/mdev = 0.025/0.036/0.053/0.010 ms
root@mininet-vm:/home/mininet#

```

Рис. 4: Проверка подключения между хостами h1 и h2

На хосте h1 добавим задержку в 100 мс к выходному интерфейсу (рис. 5):

The screenshot shows a single terminal window titled "host h1" at the top. It displays the output of a ping command to host h2, followed by the command to add a queue discipline (qdisc) with a delay of 100ms to the h1-eth0 interface.

```

X "host h1"@mininet-vm
-ping(8) options: -c count, -n no wait, -N nodeinfo opt, -s size, -t timeout, -T tos, -I interface, -f flowlabel, -6 use IPv6
For more details see ping(8).
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=0.034 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.057 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.048 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.034 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.049 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.042 ms

--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5127ms
rtt min/avg/max/mdev = 0.034/0.044/0.057/0.008 ms
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms
root@mininet-vm:/home/mininet#

```

Рис. 5: Добавление задержки в 100 мс к выходному интерфейсу на хосте h1

Проверим, что соединение от хоста h1 к хосту h2 имеет задержку 100 мс, используя команду ping с параметром -c 6 с хоста h1 (рис. 6):

```
X "host: h1"@mininet-vm
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.042 ms
--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5127ms
rtt min/avg/max/mdev = 0.034/0.044/0.057/0.008 ms
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms
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=203 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=102 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=100 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=101 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5010ms
rtt min/avg/max/mdev = 100.187/118.025/202.950/37.984 ms
root@mininet-vm:/home/mininet#
```

Рис. 6: Проверка

Для эмуляции глобальной сети с двунаправленной задержкой необходимо к соответствующему интерфейсу на хосте h2 также добавить задержку в 100 миллисекунд (рис. 7):

```
X "host: h2"@mininet-vm
loop txqueuelen 1000  (Local Loopback)
RX packets 1069 bytes 272276 (272.2 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 1069 bytes 272276 (272.2 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=0.031 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.029 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.031 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.053 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.048 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.025 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5120ms
rtt min/avg/max/mdev = 0.025/0.036/0.053/0.010 ms
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h2-eth0 root netem delay 100ms
root@mininet-vm:/home/mininet#
```

Рис. 7: Добавление задержки в 100 мс к выходному интерфейсу на хосте h2

Проверим, что соединение между хостом h1 и хостом h2 имеет RTT в 200 мс (100 мс от хоста h1 к хосту h2 и 100 мс от хоста h2 к хосту h1), повторив команду ping с параметром -c 6 на терминале хоста h1 (рис. 8):

```
X "host: h1"@mininet-vm
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=102 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=102 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=102 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=100 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5011ms
rtt min/avg/max/mdev = 100.113/101.432/102.403/0.758 ms
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=203 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=202 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=203 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=201 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=201 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=201 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 200.570/201.731/202.899/0.876 ms
root@mininet-vm:/home/mininet#
```

Рис. 8: Проверка

Изменим задержку со 100 мс до 50 мс для отправителя h1 и для получателя h2 (рис. 9):

```
X "host: h1"@mininet-vm
s
root@mininet-vm:/home/mininet# sudo tc qdisc change dev h1-eth0 root netem delay 50m
s
X "host: h2"@mininet-vm
root@mininet-vm:/home/mininet# sudo tc qdisc change dev h2-eth0 root netem delay 50m
s
root@mininet-vm:/home/mininet#
```

Рис. 9: Изменение задержки со 100 мс до 50 мс

Проверим, что соединение от хоста h1 к хосту h2 имеет задержку 100 мс, используя команду ping с параметром -c 6 с терминала хоста h1 (рис. 10):

```

X "host h1"@mininet-vm
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=201 ms
--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 200.570/201.731/202.899/0.876 ms
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 50ms
Error: Exclusivity flag on, cannot modify.
root@mininet-vm:/home/mininet# sudo tc qdisc change dev h1-eth0 root netem delay 50m
s
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=103 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=104 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=104 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=102 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5010ms
rtt min/avg/max/mdev = 100.788/102.359/104.263/1.306 ms
root@mininet-vm:/home/mininet#

```

Рис. 10: Проверка

Восстановим конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса (рис. 11):

```

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

```

Рис. 11: Восстановление конфигураций по умолчанию

Добавим на узле h1 задержку в 100 мс со случайным отклонением 10 мс (рис. 12):

```

X "host h1"@mininet-vm
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=104 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=104 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=102 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5010ms
rtt min/avg/max/mdev = 100.788/102.359/104.263/1.306 ms
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
root@mininet-vm:/home/mininet# ^C
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms
10ms
root@mininet-vm:/home/mininet#

```

Рис. 12: Добавление на узле h1 задержки в 100 мс со случайным отклонением 10 мс

Проверим, что соединение от хоста h1 к хосту h2 имеет задержку 100 мс со случайным отклонением ±10 мс, используя в терминале хоста h1 команду ping с параметром -c 6 (рис. 13):

```
X "host: h1"@mininet-vm
10ms
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=99.5 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=92.8 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=107 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=99.4 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=104 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=93.7 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5012ms
rtt min/avg/max/mdev = 92.797/99.342/106.859/5.024 ms
root@mininet-vm:/home/mininet#
```

Рис. 13: Проверка

Восстановим конфигурацию интерфейса по умолчанию на узле h1 (рис. 14):

```
X "host: h1"@mininet-vm
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=99.5 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=92.8 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=107 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=99.4 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=104 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=93.7 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5012ms
rtt min/avg/max/mdev = 92.797/99.342/106.859/5.024 ms
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
root@mininet-vm:/home/mininet#
```

Рис. 14: Восстановление конфигурации интерфейса по умолчанию

Добавим на интерфейсе хоста h1 задержку в 100 мс с вариацией ± 10 мс и значением корреляции в 25%. Убедимся, что все пакеты, покидающие устройство h1 на интерфейсе h1- eth0, будут иметь время задержки 100 мс со случайным отклонением ± 10 мс, при этом время передачи следующего пакета зависит от предыдущего значения на 25%. Используем для этого в терминале хоста h1 команду ping с параметром -c 20 (рис. 15):

```
X "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=102 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=91.5 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=105 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=102 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=92.2 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=91.5 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=94.0 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=93.7 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=98.6 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=91.2 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=93.8 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=106 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=103 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=99.0 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=102 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=98.9 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=102 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=110 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=99.6 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=104 ms

--- 10.0.0.2 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19036ms
rtt min/avg/max/mdev = 91.248/98.926/110.048/5.324 ms
root@mininet-vm:/home/mininet# █
```

Рис. 15: Проверка

Восстановим конфигурацию интерфейса по умолчанию на узле h1 (рис. 16):

```
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

Зададим нормальное распределение задержки на узле h1 в эмулируемой сети (рис. 17):

```
X "host: h1"@mininet-vm
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.057 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.070 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.139 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.071 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.065 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.067 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=0.061 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=0.071 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=0.063 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=0.077 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=0.062 ms

--- 10.0.0.2 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19388ms
rtt min/avg/max/mdev = 0.048/0.219/2.460/0.528 ms
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
Error: Invalid qdisc name.
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
Error: Invalid qdisc name.
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms
20ms distribution normal
root@mininet-vm:/home/mininet#
```

Рис. 17: Настройка нормального распределения задержки на узле h1 в эмулируемой сети

Убедимся, что все пакеты, покидающие хост h1 на интерфейсе h1-eth0, будут иметь время задержки, которое распределено в диапазоне 100 мс ±20 мс. Используем для этого команду ping на терминале хоста h1 с параметром -c 10 (рис. 18):

```
X "host: h1"@mininet-vm
Error: Invalid qdisc name.
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
Error: Invalid qdisc name.
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms
20ms distribution normal
root@mininet-vm:/home/mininet# ping -c 10 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=105 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=119 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=109 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=61.5 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=88.9 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=75.0 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=74.8 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=133 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=92.8 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=84.4 ms

--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9021ms
rtt min/avg/max/mdev = 61.549/94.374/133.395/21.013 ms
root@mininet-vm:/home/mininet#
```

Рис. 18: Проверка

Восстановим конфигурацию интерфейса по умолчанию на узле h1 (рис. 19):

```
X "host h1"@mininet-vm
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
Error: Invalid qdisc name.
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms
20ms distribution normal
root@mininet-vm:/home/mininet# ping -c 10 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=105 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=119 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=109 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=61.5 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=88.9 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=75.0 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=74.8 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=133 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=92.8 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=84.4 ms

--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9021ms
rtt min/avg/max/mdev = 61.549/94.374/133.395/21.013 ms
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

Завершим работу mininet в интерактивном режиме (рис. 20):

```
mininet@mininet-vm: ~
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Running terms on localhost:10.0
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 8 terms
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 5114.515 seconds
```

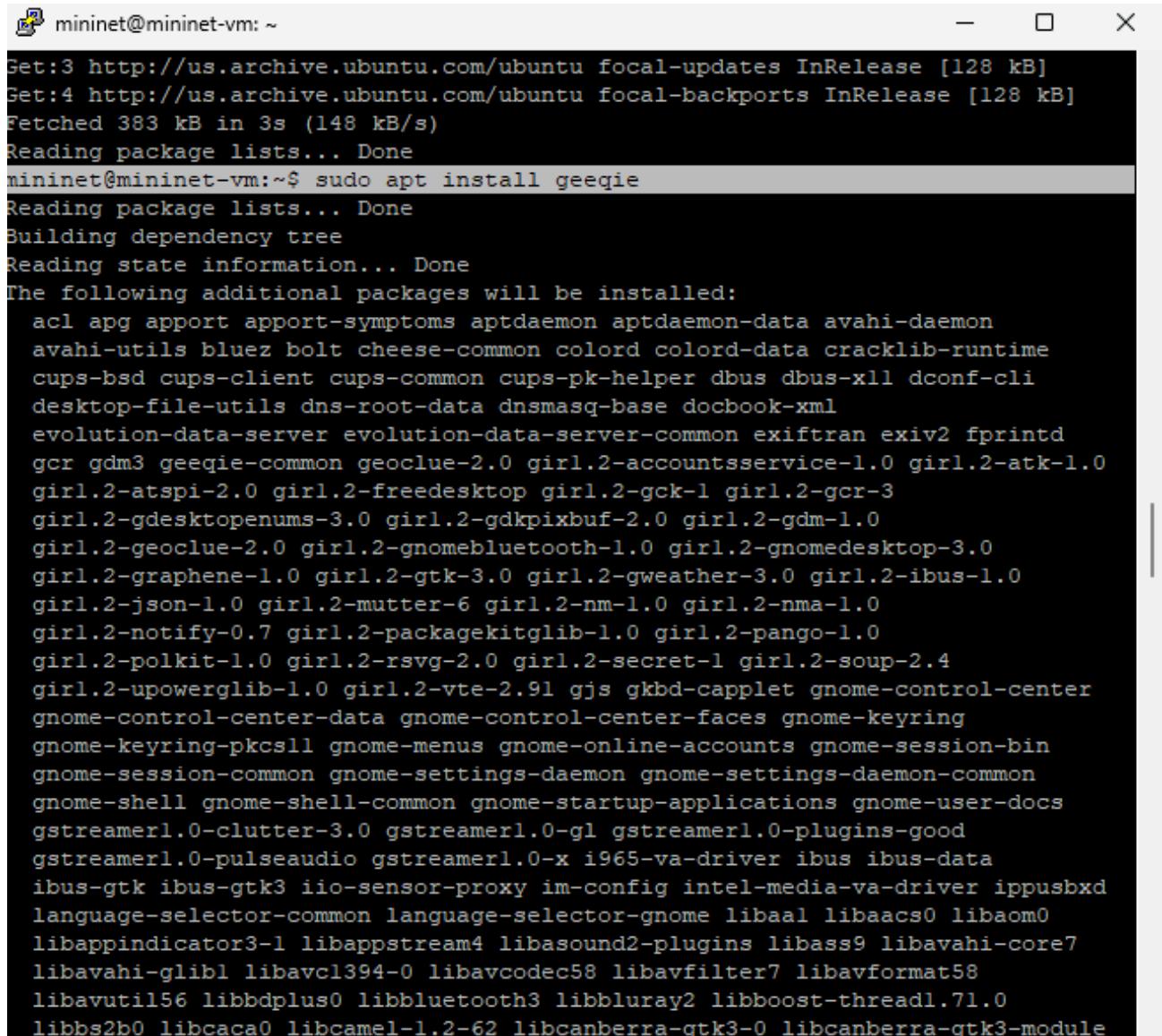
Рис. 20: Завершение работы mininet в интерактивном режиме

Обновим репозитории программного обеспечения на виртуальной машине (рис. 21):

```
mininet@mininet-vm:~$ sudo apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [128 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [128 kB]
Fetched 383 kB in 3s (148 kB/s)
Reading package lists... Done
mininet@mininet-vm:~$
```

Рис. 21: Обновление репозиториев программного обеспечения на виртуальной машине

Установим пакет geeqie для просмотра файлов png (рис. 22):



```
mininet@mininet-vm: ~
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [128 kB]
Fetched 383 kB in 3s (148 kB/s)
Reading package lists... Done
mininet@mininet-vm:~$ sudo apt install geeqie
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
acl apg apport apport-symptoms aptdaemon aptdaemon-data avahi-daemon
avahi-utils bluez bolt cheese-common colord colord-data cracklib-runtime
cups-bsd cups-client cups-common cups-pk-helper dbus dbus-xll dconf-cli
desktop-file-utils dns-root-data dnsmasq-base docbook-xml
evolution-data-server evolution-data-server-common exiftran exiv2 fprintd
gcr gdm3 geeqie-common geoclue-2.0 girl.2-accountsservice-1.0 girl.2-atk-1.0
girl.2-atspi-2.0 girl.2-freedesktop girl.2-gck-1 girl.2-gcr-3
girl.2-gdesktoopenums-3.0 girl.2-gdkpixbuf-2.0 girl.2-gdm-1.0
girl.2-geoclue-2.0 girl.2-gnomebluetooth-1.0 girl.2-gnomedesktop-3.0
girl.2-graphene-1.0 girl.2-gtk-3.0 girl.2-gweather-3.0 girl.2-ibus-1.0
girl.2-json-1.0 girl.2-mutter-6 girl.2-nm-1.0 girl.2-nma-1.0
girl.2-notify-0.7 girl.2-packagekitglib-1.0 girl.2-pango-1.0
girl.2-polkit-1.0 girl.2-rsvg-2.0 girl.2-secret-1 girl.2-soup-2.4
girl.2-upowerglib-1.0 girl.2-vte-2.91 gjs gkbd-capplet gnome-control-center
gnome-control-center-data gnome-control-center-faces gnome-keyring
gnome-keyring-pkcs11 gnome-menus gnome-online-accounts gnome-session-bin
gnome-session-common gnome-settings-daemon gnome-settings-daemon-common
gnome-shell gnome-shell-common gnome-startup-applications gnome-user-docs
gstreamer1.0-clutter-3.0 gstreamer1.0-gl gstreamer1.0-plugins-good
gstreamer1.0-pulseaudio gstreamer1.0-x i965-va-driver ibus ibus-data
ibus-gtk ibus-gtk3 iio-sensor-proxy im-config intel-media-va-driver ippusbxd
language-selector-common language-selector-gnome libaal libaacs0 libaoem0
libappindicator3-1 libappstream4 libasound2-plugins libass9 libavahi-core7
libavahi-glib1 libavcl394-0 libavcodec58 libavfilter7 libavformat58
libavutil56 libbdplus0 libbluetooth3 libbluray2 libboost-thread1.71.0
libbs2b0 libcaca0 libcamel-1.2-62 libcanberra-gtk3-0 libcanberra-gtk3-module
```

Рис. 22: Установка пакета geeqie

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

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

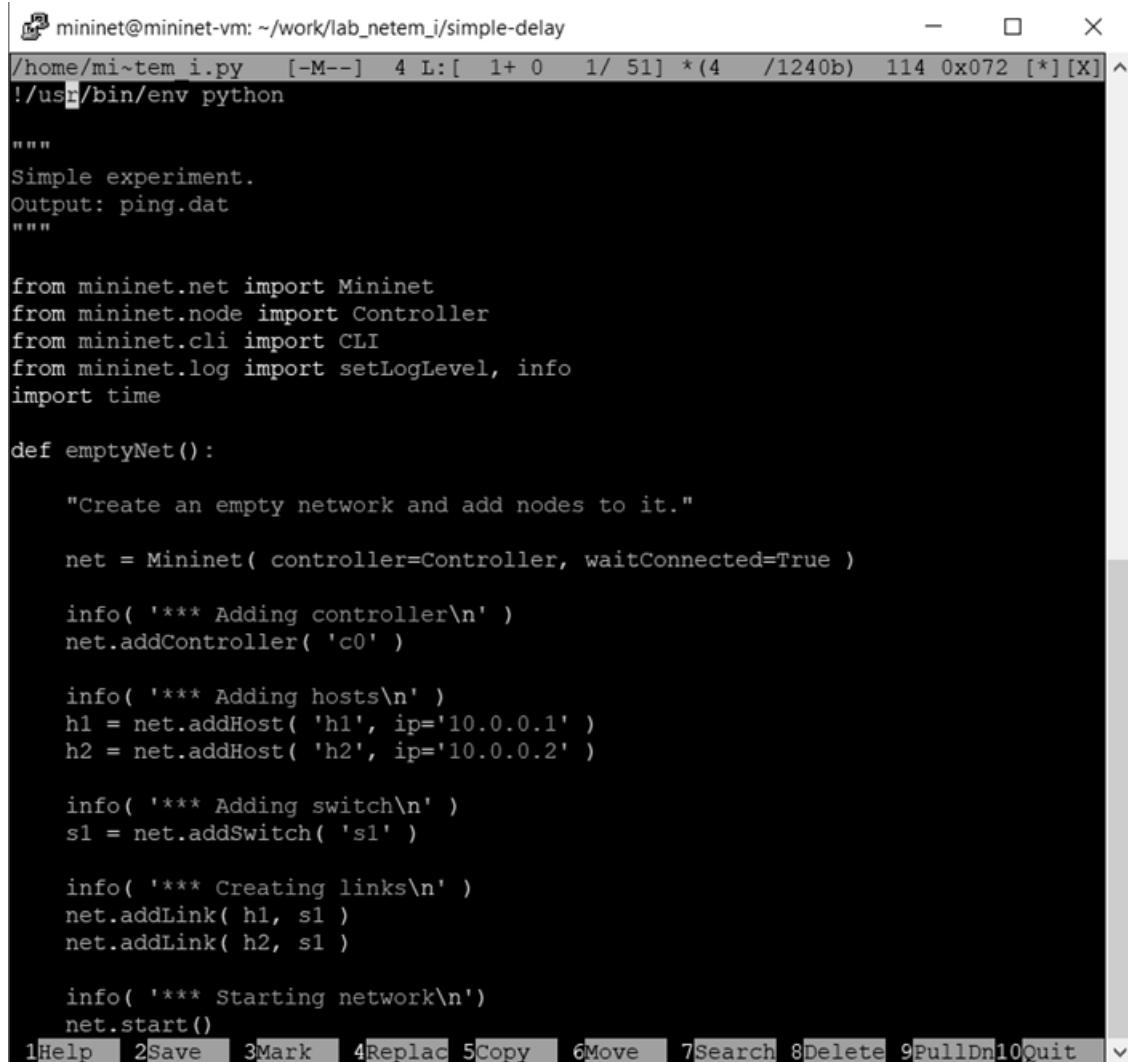
Рис. 23: Создание нового каталога

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

```
mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_i/simple-delay  
mininet@mininet-vm:~$ cd ~/work/lab_netem_i/simple-delay  
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ ls  
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ touch lab_netem_i.py  
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ ls  
lab_netem_i.py  
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nano lab_netem_i.py
```

Рис. 24: Создание каталога simple-delay

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



```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay  
/home/mininet-i.py [-M--] 4 L:[ 1+ 0 1/ 51 ] *(4 /1240b) 114 0x072 [*][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()
```

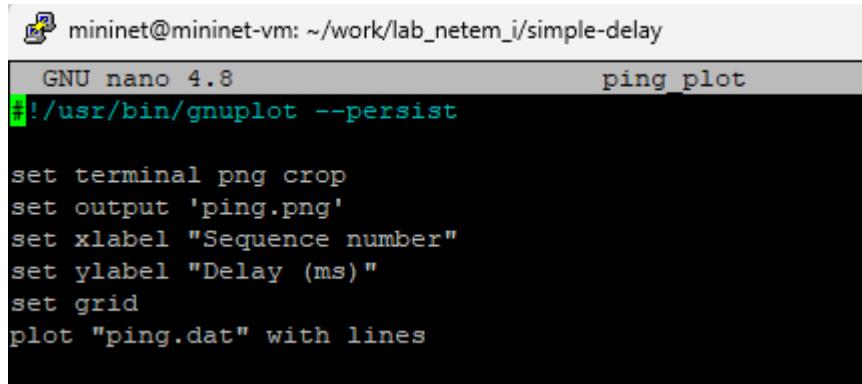
Рис. 25: Создание скрипта lab_netem_i.py для эксперимента

Создадим файл ping_plot (рис. 26):

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ touch ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nano ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ █
```

Рис. 26: Создание файла ping_plot

Затем создадим скрипт для визуализации ping_plot результатов эксперимента (рис. 27):



```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
GNU nano 4.8          ping plot
#!/usr/bin/gnuplot --persist

set terminal png crop
set output 'ping.png'
set xlabel "Sequence number"
set ylabel "Delay (ms)"
set grid
plot "ping.dat" with lines
```

Рис. 27: Создание скрипта ping_plot для визуализации результатов эксперимента

Зададим права доступа к файлу скрипта (рис. 28):

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ chmod +x ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ █
```

Рис. 28: Настройка прав доступа к файлу скрипта

Создадим файла Makefile (рис. 29):

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ touch Makefile
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nano m
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ ls
lab_netem_i.py Makefile ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nano Makefile
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nano Makefile
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ █
```

Рис. 29: Создание файла Makefile

Внутри файла Makefile поместим скрипт для управления процессом проведения эксперимента (рис. 30):

```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
GNU nano 4.8                                     Makefile
all: ping.dat ping.png

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

ping.png: ping.dat
    ./ping_plot

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

Рис. 30: Добавления скрипта в Makefile для управления процессом проведения эксперимента

Выполним эксперимент (рис. 31):

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make
sudo python lab_netem_i.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 100ms',)
*** h2 : ('tc qdisc add dev h2-eth0 root netem delay 100ms')
*** Ping
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "time=' | awk '{print $5, $7}' | sed -e \'s/time=/g\' -e \'s/icmp_seq=/g\' > ping.dat')

*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$
```

Рис. 31: Выполнение эксперимента

Просмотрим построенный в результате выполнения скриптов график (рис. 32):

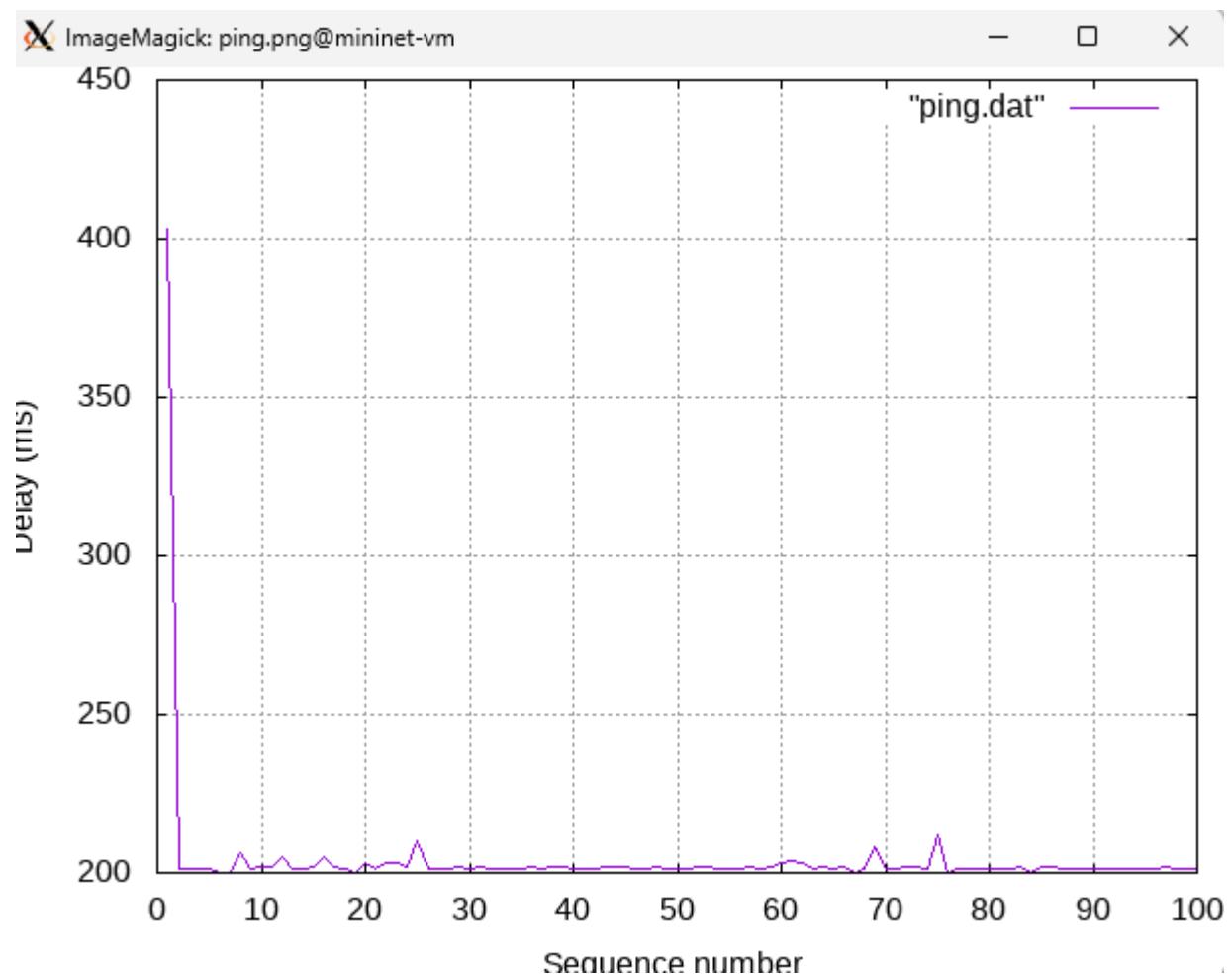


Рис. 32: Просмотр графика

Из файла ping.dat удалим первую строку и заново построим график (рис. 33 - рис. 34):

GNU nano 4.8 ping.dat Modified

2 201
3 201
4 201
5 201
6 200
7 200
8 206
9 201
10 202
11 202
12 205
13 201
14 201
15 202
16 205
17 202
18 201
19 200
20 203
21 201
22 203
23 203
24 202
25 210
26 201
27 201
28 201
29 202
30 201
31 202
32 201
33 201
34 201
35 201
36 202
37 201
38 202
39 202
40 201
41 201
42 201
43 202
44 202

[Read 100 lines]

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

Рис. 33: Удаление первой строчки из файла ping.dat

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make ping.png
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make ping.png
make: 'ping.png' is up to date.
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ display ping.png
```

Рис. 34: Повторное построение графика

Просмотрим заново построенный график (рис. 35):

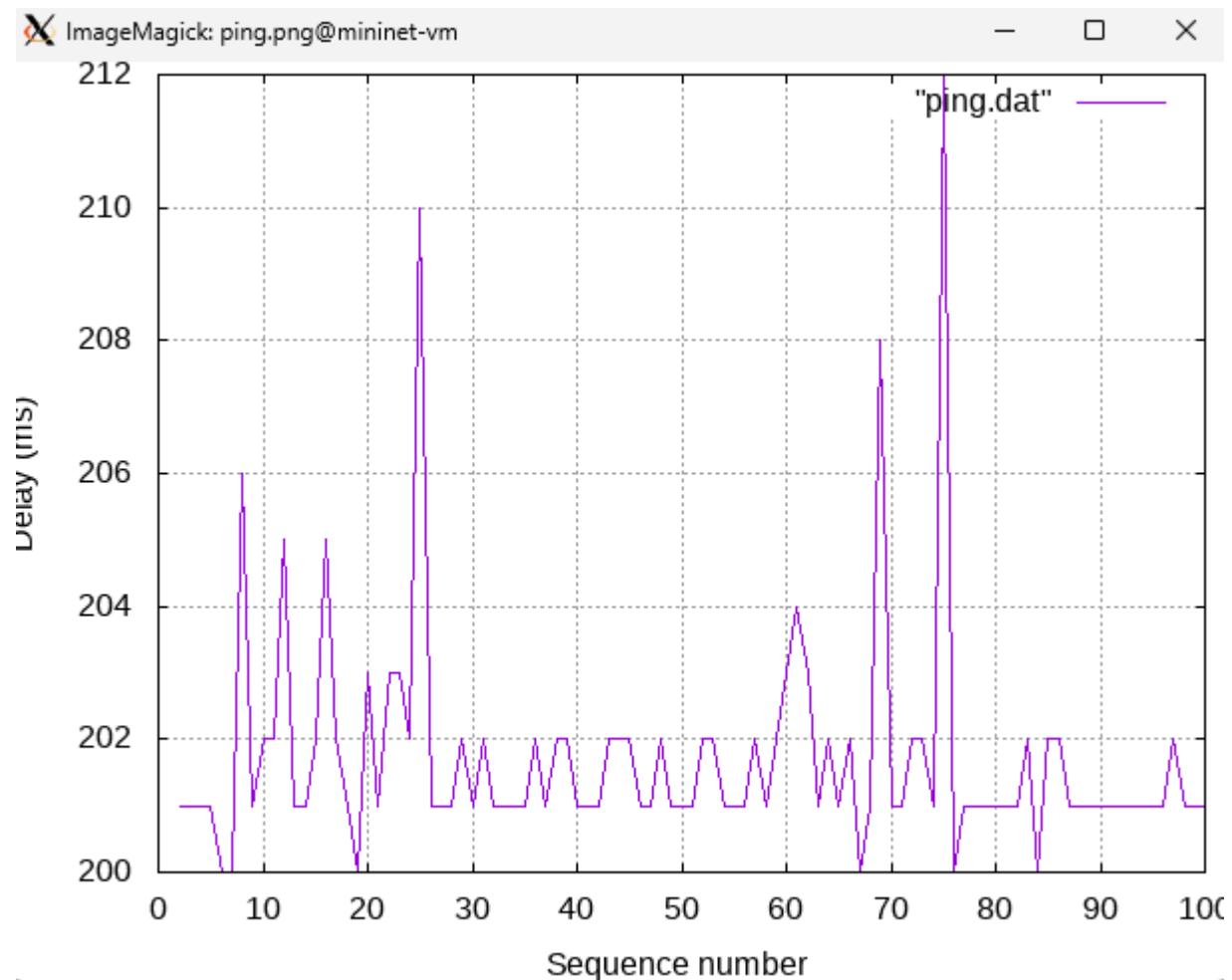


Рис. 35: Просмотр графика

Разработаем скрипт для вычисления на основе данных файла ping.dat минимального, среднего, максимального и стандартного отклонения времени приёма-передачи. Также добавим правило запуска скрипта в Makefile (рис. 36 - рис. 38):

```
GNU nano 4.8                                         Makefile                                         Modified
all: ping.dat ping.png
#
#ping.dat:
#    sudo python lab_netem_i.py
#    sudo chown mininet:mininet ping.dat
#
#ping.png: ping.dat
#    ./ping_plot
#
#clean:
#    -rm -f *.dat *.png

import numpy as np

def calculate_stat(data):
    times = np.array([float(line.split()[1]) for line in data])
    min_time = np.min(times)
    avg_time = np.mean(times)
    max_time = np.max(times)
    std_dev = np.std(times)

    return min_time, avg_time, max_time, std_dev

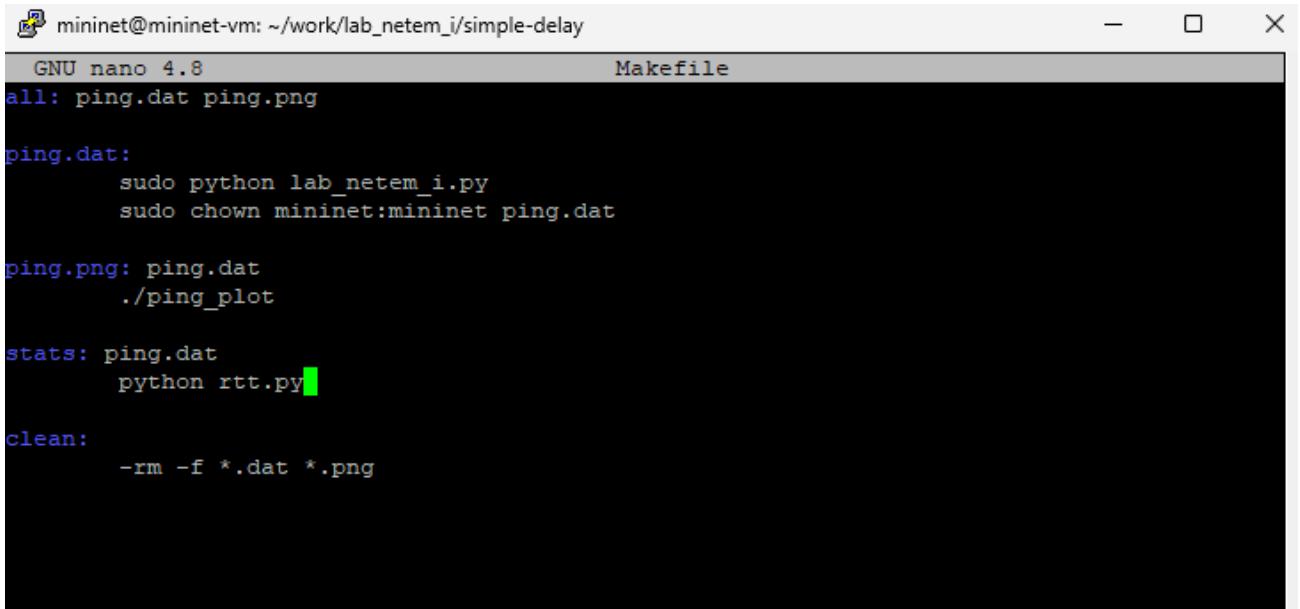
def read_file():
    with open('ping.dat', 'r') as file:
        data = file.readlines()

    min_time, avg_time, max_time, std_dev = calc_stat(data)

    print(f"Min time: {min_time} ms")
    print(f"Avg time: {avg_time} ms")
    print(f"Max time: {max_time} ms")
    print(f"Std dev: {std_dev} ms")

if __name__ == "__main__":
    read_file()
```

Рис. 36: Разработка скрипта для вычисления на основе данных файла *ping.dat* минимального, среднего, максимального и стандартного отклонения времени приёма-передачи



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

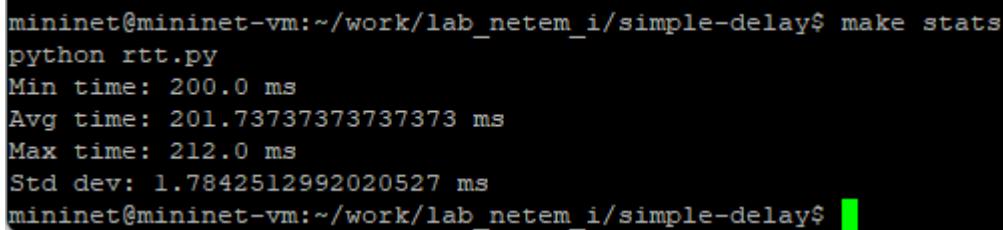
ping.dat:
    sudo python lab_netem_i.py
    sudo chown mininet:mininet ping.dat

ping.png: ping.dat
    ./ping_plot

stats: ping.dat
    python rtt.py

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

Рис. 37: Добавление правила запуска скрипта в Makefile



```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make stats
python rtt.py
Min time: 200.0 ms
Avg time: 201.737373737373 ms
Max time: 212.0 ms
Std dev: 1.7842512992020527 ms
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$
```

Рис. 38: Проверка

Очистим каталог от результатов проведения экспериментов.

Самостоятельно реализуем воспроизводимые эксперименты по изменению задержки, джиттера, значения корреляции для джиттера и задержки, распределения времени задержки в эмулируемой глобальной сети. Построим графики. Вычислим минимальное, среднее, максимальное и стандартное отклонение времени приёма-передачи для каждого случая (рис. 39 - рис. 50):

The screenshot shows a terminal window titled "mininet@mininet-vm: ~/work/lab_netem_i/simple-delay". The window contains a file named "lab_netem_i.py" which is being edited in a nano text editor. The code is a Python script for setting up a network experiment using Mininet and netem. It defines a function "emptyNet" that creates a network with two hosts (h1 and h2) connected to a switch (s1), and applies netem delay to their interfaces. It also includes a ping command at the end.

```
GNU nano 4.8          lab_netem_i.py          Modified
#!/usr/bin/env python

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

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

def emptyNet():

    "Create an empty network and add nodes to it."

    net = Mininet( controller=Controller, waitConnected=True )

    info( "*** Adding controller\n" )
    net.addController( 'c0' )

    info( "*** Adding hosts\n" )
    h1 = net.addHost( 'h1', ip='10.0.0.1' )
    h2 = net.addHost( 'h2', ip='10.0.0.2' )

    info( "*** Adding switch\n" )
    s1 = net.addSwitch( 's1' )

    info( "*** Creating links\n" )
    net.addLink( h1, s1 )
    net.addLink( h2, s1 )

    info( "*** Starting network\n" )
    net.start()

    info( "*** Set delay\n" )
    h1.cmdPrint( 'tc qdisc add dev h1-eth0 root netem delay 50ms' )
    h2.cmdPrint( 'tc qdisc add dev h2-eth0 root netem delay 50ms' )

    time.sleep(10) # Wait 10 seconds

    info( "*** Ping\n" )

^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. 39: Воспроизводимый эксперимент по изменению задержки

```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ cd ~/work/lab_netem_i/simple-delay
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ ls
lab netem i.py Makefile ping.dat ping plot ping.png rtt.py
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nano lab netem i.py
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make clean
rm -f *.dat *.png
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make
sudo python lab netem i.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 50ms',)
*** h2 : ('tc qdisc add dev h2-eth0 root netem delay 50ms',)
*** Ping
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "time=" | awk \'{print $5, $7}\' | sed -e \'s/time=/\\g\\' -e \'/s/icmp_seq=/\\g\\' > ping.dat')

*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ 
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make stats
python rtt.py
Min time: 100.0 ms
Avg time: 102.41 ms
Max time: 208.0 ms
Std dev: 10.64621528995164 ms
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$
```

Рис. 40: Воспроизводимый эксперимент по изменению задержки

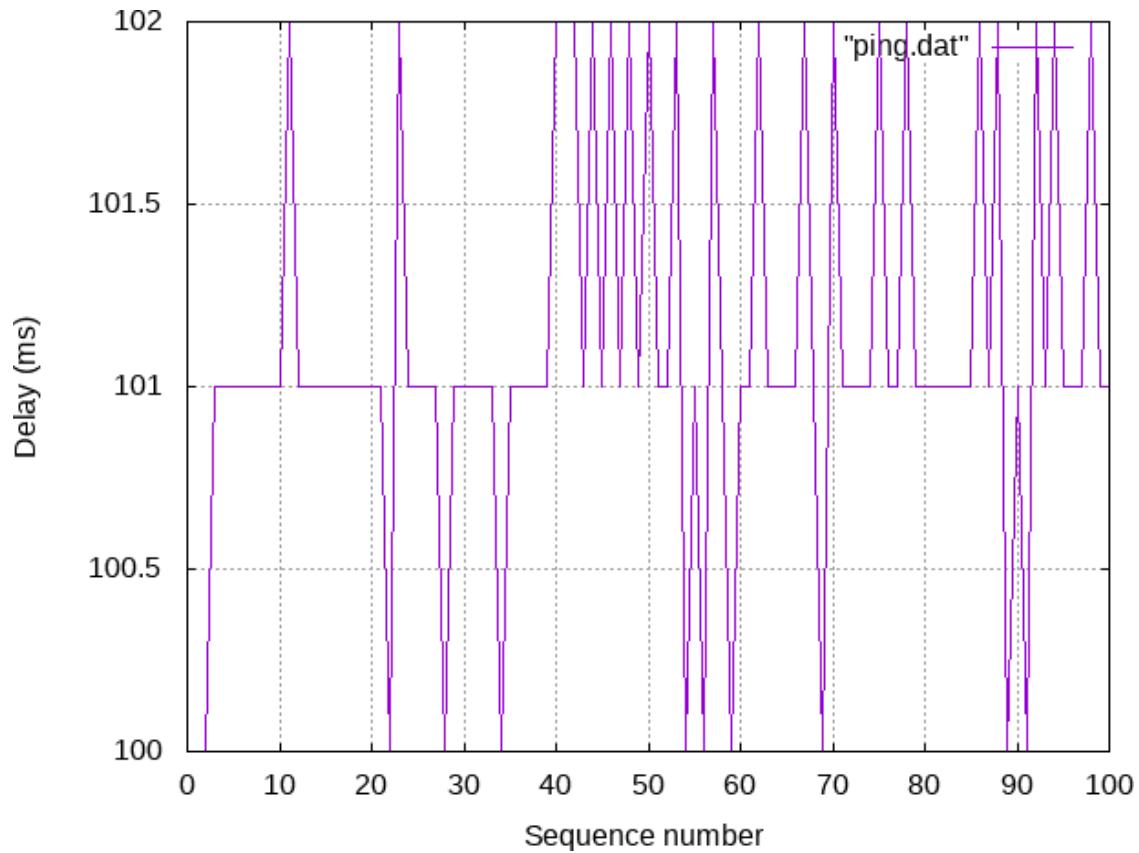


Рис. 41: Просмотр графика

The screenshot shows a terminal window titled "mininet@mininet-vm: ~/work/lab_netem_i/simple-delay". The window contains a Python script named "lab_netem_i.py". The script imports Mininet, Controller, CLI, log, and time modules. It defines an "emptyNet" function that creates a network with two hosts (h1 and h2) connected to a switch (s1), and sets delays for their interfaces. It then starts the network and performs a ping. The terminal also displays a set of keyboard shortcuts at the bottom.

```
GNU nano 4.8                               lab_netem_i.py                                Modified
#!/usr/bin/env python

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

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

def emptyNet():

    "Create an empty network and add nodes to it."

    net = Mininet( controller=Controller, waitConnected=True )

    info( "*** Adding controller\n" )
    net.addController( 'c0' )

    info( "*** Adding hosts\n" )
    h1 = net.addHost( 'h1', ip='10.0.0.1' )
    h2 = net.addHost( 'h2', ip='10.0.0.2' )

    info( "*** Adding switch\n" )
    s1 = net.addSwitch( 's1' )

    info( "*** Creating links\n" )
    net.addLink( h1, s1 )
    net.addLink( h2, s1 )

    info( "*** Starting network\n" )
    net.start()

    info( "*** Set delay\n" )
    h1.cmdPrint( 'tc qdisc add dev h1-eth0 root netem delay 100ms 10ms' )
    h2.cmdPrint( 'tc qdisc add dev h2-eth0 root netem delay 100ms 10ms' )

    time.sleep(10) # Wait 10 seconds

    info( "*** Ping\n" )

^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. 42: Воспроизводимый эксперимент по изменению джиттера

```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make stats
python rtt.py
Min time: 100.0 ms
Avg time: 102.41 ms
Max time: 208.0 ms
Std dev: 10.64621528995164 ms
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ ls
lab_netem_i.py Makefile ping.dat ping_plot ping.png rtt.py
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nano lab_netem_i.py
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make clean
rm -f *.dat *.png
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make
sudo python lab_netem_i.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 100ms 10ms',)
*** h2 : ('tc qdisc add dev h2-eth0 root netem delay 100ms',)
*** Ping
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "time=" | awk \'{print $5, $7}\' | sed -e \'s/time=//g\' -e \'s/icmp_seq//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$
```

Рис. 43: Воспроизводимый эксперимент по изменению джиттера

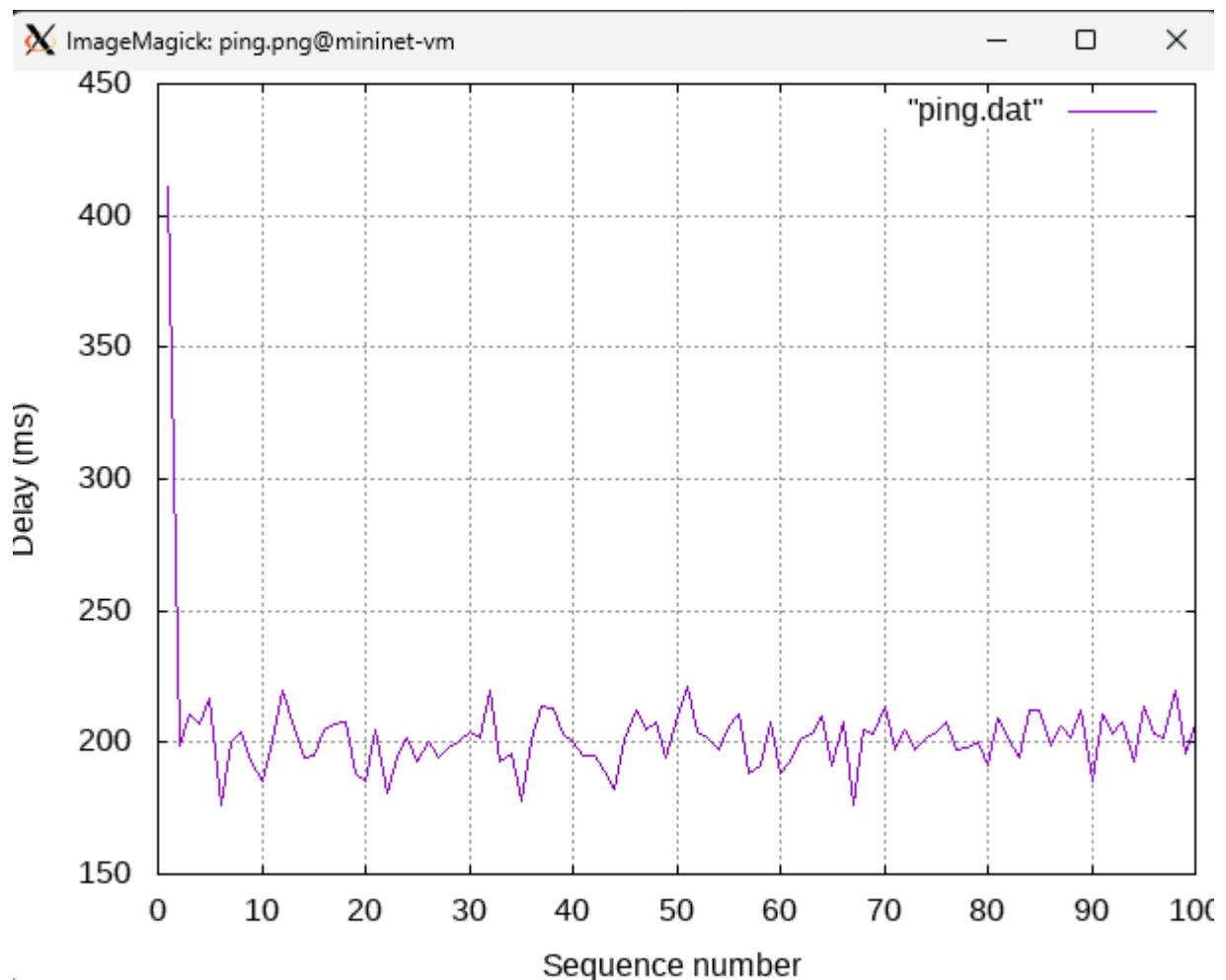


Рис. 44: Просмотр графика

The screenshot shows a terminal window titled "mininet@mininet-vm: ~/work/lab_netem_i/simple-delay". The window contains a file named "lab_netem_i.py" which is being edited in nano 4.8. The code is a script for setting up a network with two hosts (h1, h2) connected to a switch (s1). It uses the netem module to add latency to the link between h1 and s1. The script includes comments explaining its purpose and how it sets up the network.

```
GNU nano 4.8                               lab_netem_i.py                                Modified
#!/usr/bin/env python

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

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

def emptyNet():

    "Create an empty network and add nodes to it."

    net = Mininet( controller=Controller, waitConnected=True )

    info( "*** Adding controller\n" )
    net.addController( 'c0' )

    info( "*** Adding hosts\n" )
    h1 = net.addHost( 'h1', ip='10.0.0.1' )
    h2 = net.addHost( 'h2', ip='10.0.0.2' )

    info( "*** Adding switch\n" )
    s1 = net.addSwitch( 's1' )

    info( "*** Creating links\n" )
    net.addLink( h1, s1 )
    net.addLink( h2, s1 )

    info( "*** Starting network\n" )
    net.start()

    info( "*** Set delay\n" )
    h1.cmdPrint( 'tc qdisc add dev h1-eth0 root netem delay 100ms 10ms 25%' )
    h2.cmdPrint( 'tc qdisc add dev h2-eth0 root netem delay 100ms' )

    time.sleep(10) # Wait 10 seconds

    info( "*** Ping\n" )

^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. 45: Воспроизведимый эксперимент по изменению значения корреляции для джиттера и задержки

```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ display ping.png
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nano lab_netem_i.py
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make clean
rm -f *.dat *.png
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make
sudo python lab_netem_i.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 100ms 10ms 25%',)
*** h2 : ('tc qdisc add dev h2-eth0 root netem delay 100ms',)
*** Ping
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "time=' | awk \'{print $5, $7}\' | sed -e \'s/time=/\\g\\' -e \'/s/icmp_seq=/\\g\\' > ping.dat')

*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$
```

Рис. 46: Воспроизводимый эксперимент по изменению значения корреляции для джиттера и задержки

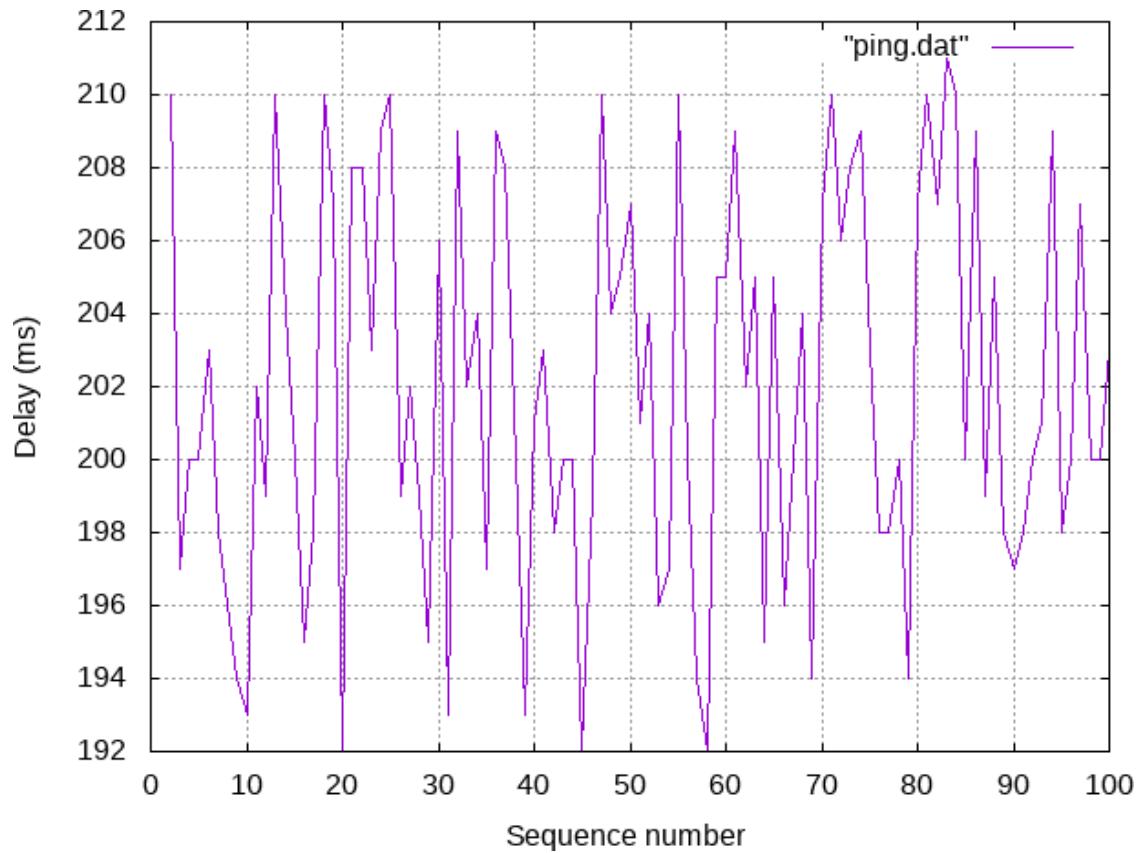


Рис. 47: Просмотр графика

The screenshot shows a terminal window titled "mininet@mininet-vm: ~/work/lab_netem_i/simple-delay". The window contains a Python script named "lab_netem_i.py". The script imports necessary modules from mininet and sets up a network with two hosts (h1, h2) connected to a switch (s1). It then configures traffic classes (qdisc) on both hosts to use the NetEM discipline with a normal distribution of delays. Finally, it performs a ping between h1 and h2. A command-line interface menu is visible at the bottom of the terminal.

```
GNU nano 4.8                                         lab_netem_i.py                                         Modified
#!/usr/bin/env python

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

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

def emptyNet():

    "Create an empty network and add nodes to it."

    net = Mininet( controller=Controller, waitConnected=True )

    info( '*** Adding controller\n' )
    net.addController( 'c0' )

    info( '*** Adding hosts\n' )
    h1 = net.addHost( 'h1', ip='10.0.0.1' )
    h2 = net.addHost( 'h2', ip='10.0.0.2' )

    info( '*** Adding switch\n' )
    s1 = net.addSwitch( 's1' )

    info( '*** Creating links\n' )
    net.addLink( h1, s1 )
    net.addLink( h2, s1 )

    info( '*** Starting network\n' )
    net.start()

    info( '*** Set delay\n' )
    h1.cmdPrint( 'tc qdisc add dev h1-eth0 root netem delay 100ms 10ms 25% distribution normal' )
    h2.cmdPrint( 'tc qdisc add dev h2-eth0 root netem delay 100ms' )

    time.sleep(10) # Wait 10 seconds

    info( '*** Ping\n' )

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

Рис. 48: Воспроизводимый эксперимент по изменению распределения времени задержки в эмулируемой глобальной сети

```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ 
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ display ping.png
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nano lab_netem_i.py
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ display ping.png
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nano lab_netem_i.py
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make clean
rm -f *.dat *.png
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make
sudo python lab_netem_i.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 100ms 10ms 25% distribution normal',)
*** h2 : ('tc qdisc add dev h2-eth0 root netem delay 100ms',)
*** Ping
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "time=' | awk \'(print $5, $7)\' | sed -e \'s/time=/g\' -e \'s/icmp_seq=/g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
...
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$
```

Рис. 49: Воспроизводимый эксперимент по изменению распределения времени задержки в эмулируемой глобальной сети

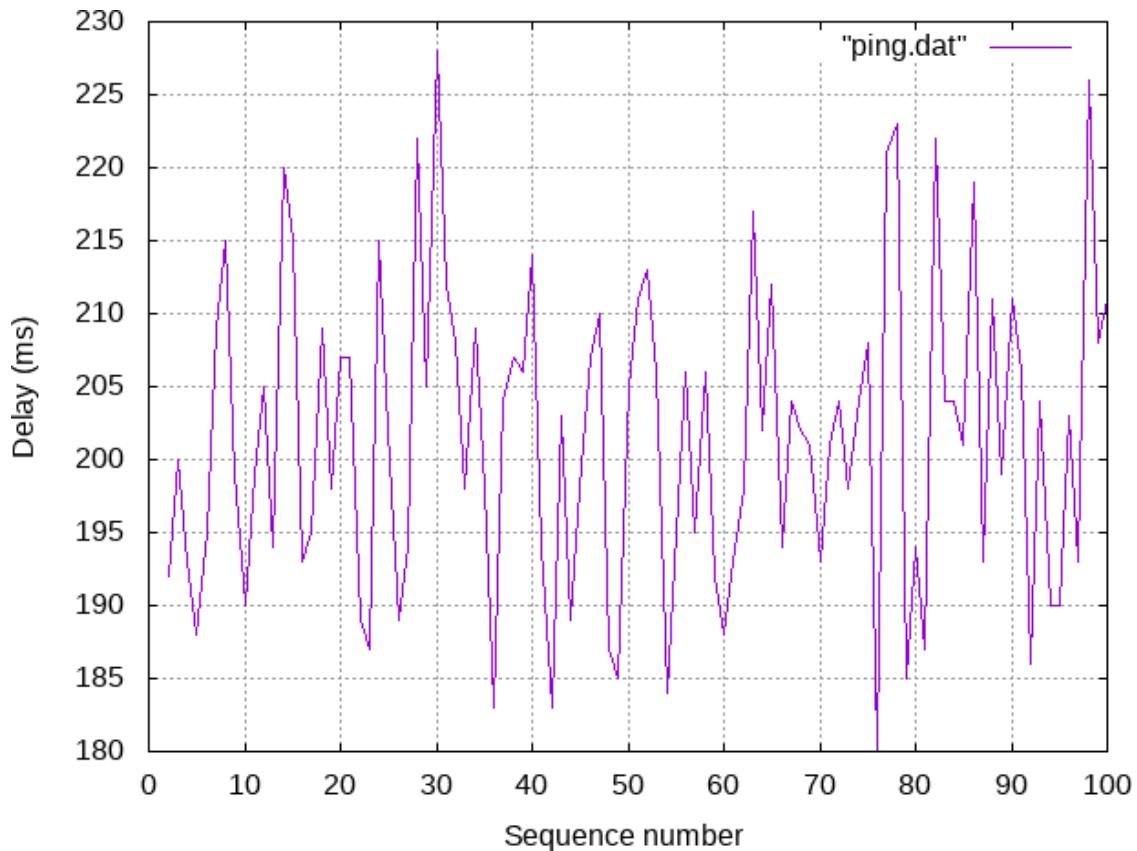


Рис. 50: Просмотр графика

3 Вывод

В ходе выполнения лабораторной работы познакомились с NETEM — инструментом для тестирования производительности приложений в виртуальной сети, а также получили навыки проведения интерактивного и воспроизводимого экспериментов по измерению задержки и её дрожания (jitter) в моделируемой сети в среде Mininet.

4 Список литературы. Библиография

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