

Моделирование сетей передачи данных

Лабораторная работа №4: Эмуляция и измерение задержек в глобальных сетях

Кармацкий Никита Сергеевич

Российский университет дружбы народов, Москва, Россия

Цель лабораторной 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.

```
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  42b83fbe456aae740de47d0ab4cd102f
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth add mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  42b83fbe456aae740de47d0ab
4cd102f
root@mininet-vm:~# logout
```

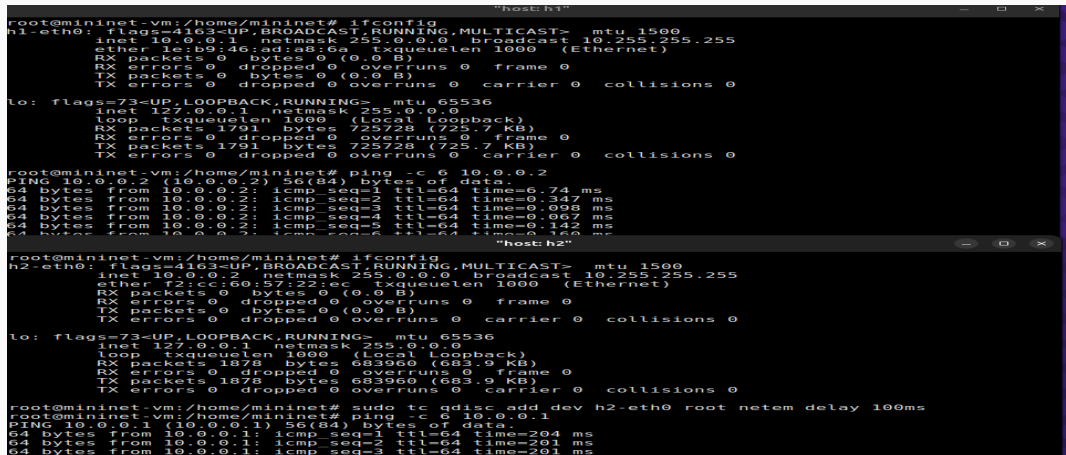
Рис. 1: Исправление прав запуска X-соединения в виртуальной машине mininet

1. Запуск лабораторной топологии

```
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> sudo tc qdisc add dev h1-eth0 root netem delay 100ms
*** Unknown command: sudo tc qdisc add dev h1-eth0 root netem delay 100ms
mininet> █
```

Рис. 2: Создание простейшей топологии

1. Запуск лабораторной топологии



```
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 1e:b9:46:ad:a8:6a 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 1791 bytes 725728 (725.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1791 bytes 725728 (725.7 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=6.74 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.347 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.098 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.067 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.142 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.160 ms

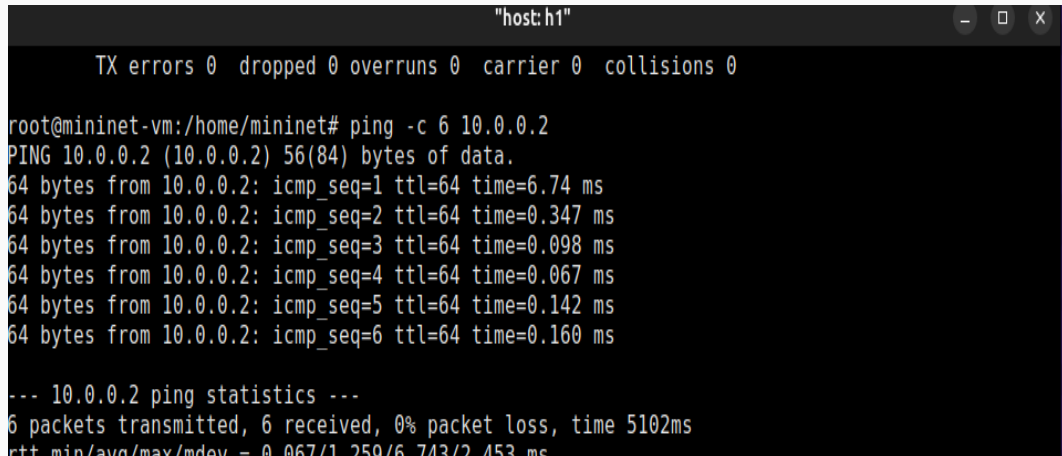
root@mininet-vm:/home/mininet# ifconfig
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.2 netmask 255.0.0.0 broadcast 10.255.255.255
    ether f2:cc:60:57:22:ec txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX 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 1878 bytes 683960 (683.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1878 bytes 683960 (683.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet# sudo tc qdisc add dev h2-eth0 root netem delay 100ms
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=204 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=201 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=201 ms
```

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

1. Запуск лабораторной топологии



```
"host: h1"
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=6.74 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.347 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.098 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.067 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.142 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.160 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5102ms
rtt min/avg/max/mdev = 0.067/1.250/6.743/2.453 ms
```

Рис. 4: Проверка подключения между хостами

2. Интерактивные эксперименты

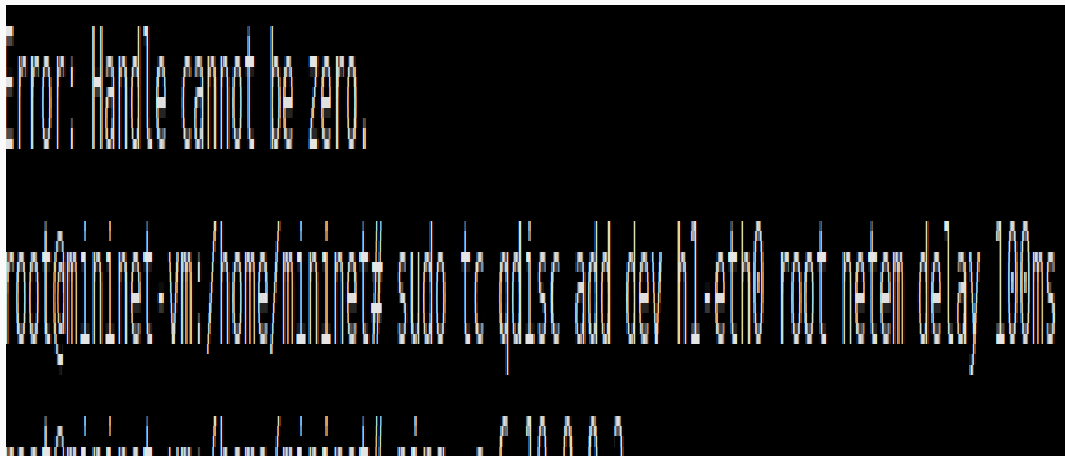


Рис. 5: Добавление задержки в 100 мс к выходному интерфейсу на хосте h1

2. Интерактивные эксперименты

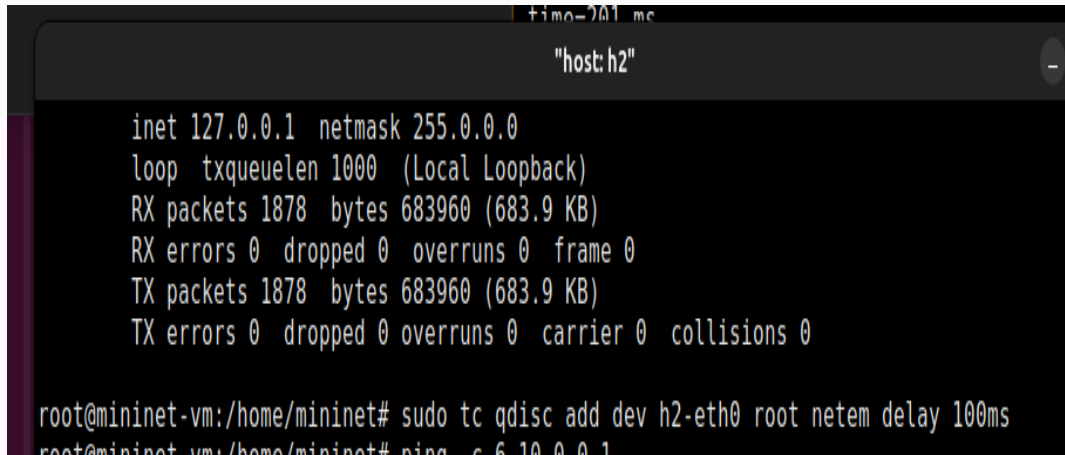
```
host: h1

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5102ms
rtt min/avg/max/mdev = 0.067/1.259/6.743/2.453 ms
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 netem delay 10
Error: Handle cannot be zero.
root@mininet-vm:/home/mininet# tc qdisc add dev h1-eth0 netem delay 100ms
Error: Handle cannot be zero.
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem del
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=105 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=101 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 5008ms
rtt min/avg/max/mdev = 100.578/101.785/105.390/1.702 ms
```

Рис. 6: Проверка

2. Интерактивные эксперименты



```
time-201 ms
"host: h2"

inet 127.0.0.1 netmask 255.0.0.0
loop txqueuelen 1000 (Local Loopback)
RX packets 1878 bytes 683960 (683.9 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 1878 bytes 683960 (683.9 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet# sudo tc qdisc add dev h2-eth0 root netem delay 100ms
root@mininet-vm:/home/mininet# ping -c 6 10.0.0.1
```

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

2. Интерактивные эксперименты

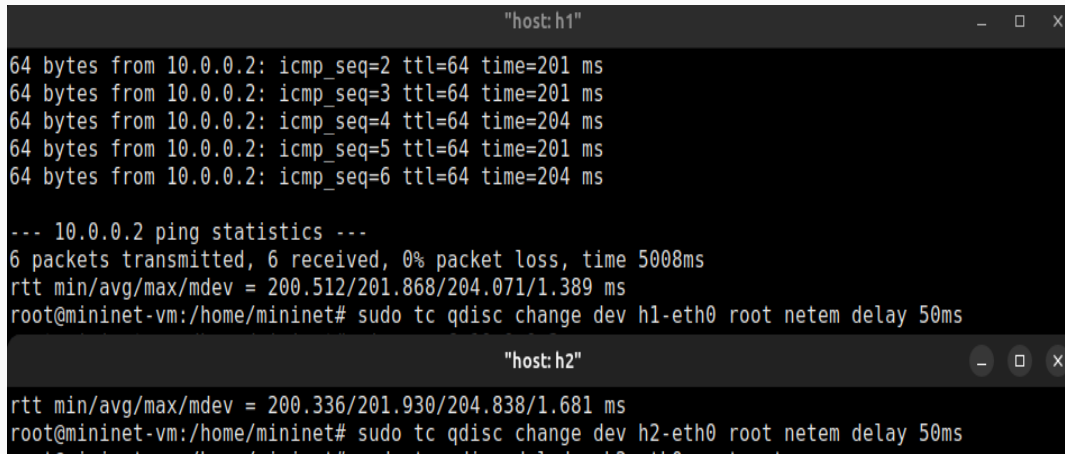
```

"host: h1"
rtt min/avg/max/mdev = 100.578/101.785/105.390/1.702 ms
root@mininet-vm:/home/mininet# ^C
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=201 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=201 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=201 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=204 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=204 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 200.512/201.868/204.071/1.389 ms
root@mininet-vm:/home/mininet# sudo tcpdump -i eth0 -c 10 -s 1500 -w /tmp/ping.pcap
```

Рис. 8: Проверка

2. Интерактивные эксперименты



The image shows two terminal windows. The top window, titled "host: h1", displays the results of a ping command to 10.0.0.2. It shows five successful pings with times ranging from 201 ms to 204 ms. Below this, it shows the ping statistics: 6 packets transmitted, 6 received, 0% packet loss, and a total time of 5008ms. The round-trip times (rtt) are listed as min/avg/max/mdev = 200.512/201.868/204.071/1.389 ms. The bottom window, titled "host: h2", shows the same rtt statistics and then the execution of the command `sudo tc qdisc change dev h2-eth0 root netem delay 50ms` to configure a 50ms delay on the interface.

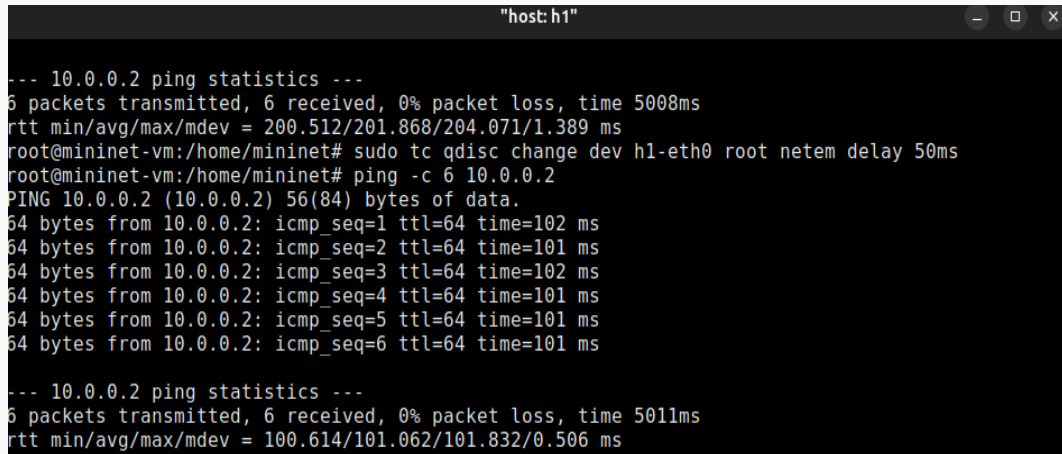
```
"host: h1"
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=201 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=201 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=204 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=201 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=204 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 200.512/201.868/204.071/1.389 ms
root@mininet-vm:/home/mininet# sudo tc qdisc change dev h1-eth0 root netem delay 50ms

"host: h2"
rtt min/avg/max/mdev = 200.336/201.930/204.838/1.681 ms
root@mininet-vm:/home/mininet# sudo tc qdisc change dev h2-eth0 root netem delay 50ms
```

Рис. 9: Изменение задержек до 100 мс до 50мс на хостах

2. Интерактивные эксперименты



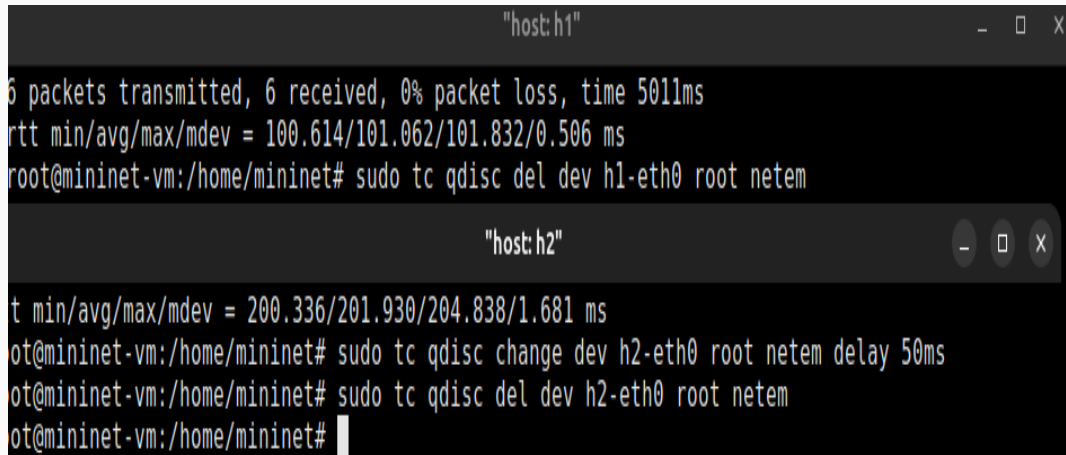
```
"host: h1"

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 200.512/201.868/204.071/1.389 ms
root@mininet-vm:/home/mininet# sudo tc qdisc change dev h1-eth0 root netem delay 50ms
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=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=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 5011ms
rtt min/avg/max/mdev = 100.614/101.062/101.832/0.506 ms
```

Рис. 10: Проверка

2. Интерактивные эксперименты



The image shows two terminal windows. The top window, titled "host: h1", displays network statistics: "6 packets transmitted, 6 received, 0% packet loss, time 5011ms" and "rtt min/avg/max/mdev = 100.614/101.062/101.832/0.506 ms". It then shows the command "root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem". The bottom window, titled "host: h2", shows the command "ot@mininet-vm:/home/mininet# sudo tc qdisc change dev h2-eth0 root netem delay 50ms", followed by "ot@mininet-vm:/home/mininet# sudo tc qdisc del dev h2-eth0 root netem", and finally "ot@mininet-vm:/home/mininet#" with a cursor. The top window also shows a partial command "t min/avg/max/mdev = 200.336/201.930/204.838/1.681 ms" at the top.

```
"host: h1"
6 packets transmitted, 6 received, 0% packet loss, time 5011ms
rtt min/avg/max/mdev = 100.614/101.062/101.832/0.506 ms
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem

"host: h2"
t min/avg/max/mdev = 200.336/201.930/204.838/1.681 ms
ot@mininet-vm:/home/mininet# sudo tc qdisc change dev h2-eth0 root netem delay 50ms
ot@mininet-vm:/home/mininet# sudo tc qdisc del dev h2-eth0 root netem
ot@mininet-vm:/home/mininet#
```

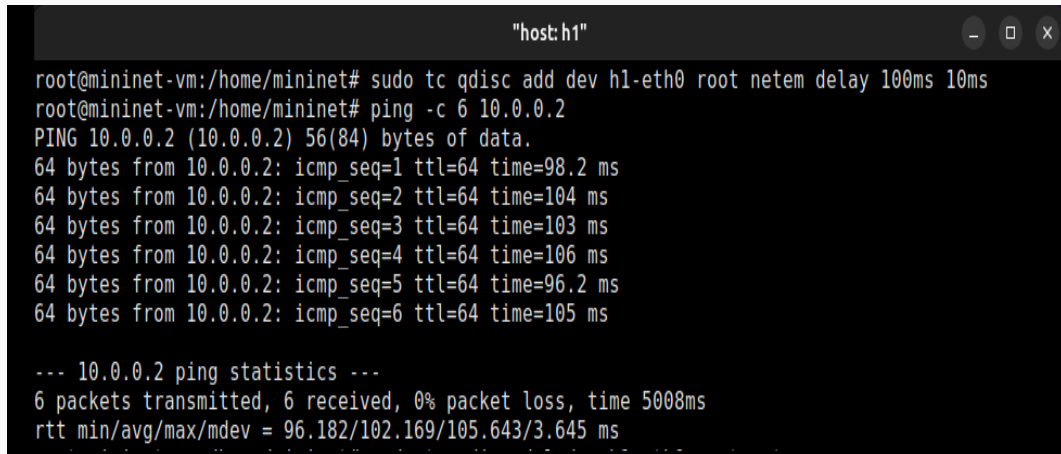
Рис. 11: Восстановление конфигураций по умолчанию

2. Интерактивные эксперименты



Рис. 12: Добавление задержки 100 мс со случайным отклонением на хост h1

2. Интерактивные эксперименты



A terminal window titled "host: h1" with standard window controls. It shows a user running a series of commands to configure network delay and perform a ping test. The commands are: `sudo tc qdisc add dev h1-eth0 root netem delay 100ms 10ms` and `ping -c 6 10.0.0.2`. The output shows six successful ping responses with varying round-trip times (96.2ms to 106ms). A summary line indicates 0% packet loss and provides detailed statistics: `rtt min/avg/max/mdev = 96.182/102.169/105.643/3.645 ms`.

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms 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=98.2 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=103 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=106 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=96.2 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=105 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 96.182/102.169/105.643/3.645 ms
```

Рис. 13: Проверка

2. Интерактивные эксперименты

```
--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 96.182/102.169/105.643/3.645 ms
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
```

Рис. 14: Восстановление конфигурацию по умолчанию

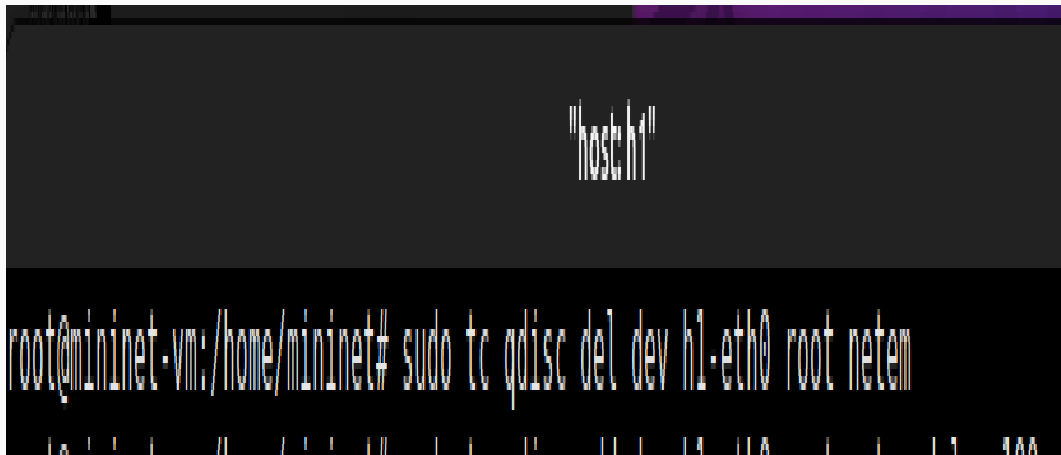
2. Интерактивные эксперименты

```
"host: h1"
rtt min/avg/max/mdev = 96.182/102.169/105.643/3.645 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 delay 100ms 10ms 25%
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=103 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=93.7 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=108 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=95.8 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=96.8 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=104 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=95.5 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=95.7 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=105 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=100 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=100 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=90.7 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=104 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=107 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=95.7 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=93.3 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=105 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=110 ms

--- 10.0.0.2 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 1904lms
rtt min/avg/max/mdev = 90.686/100.307/110.096/5.297 ms
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
```

Рис. 15: Проверка

2. Интерактивные эксперименты

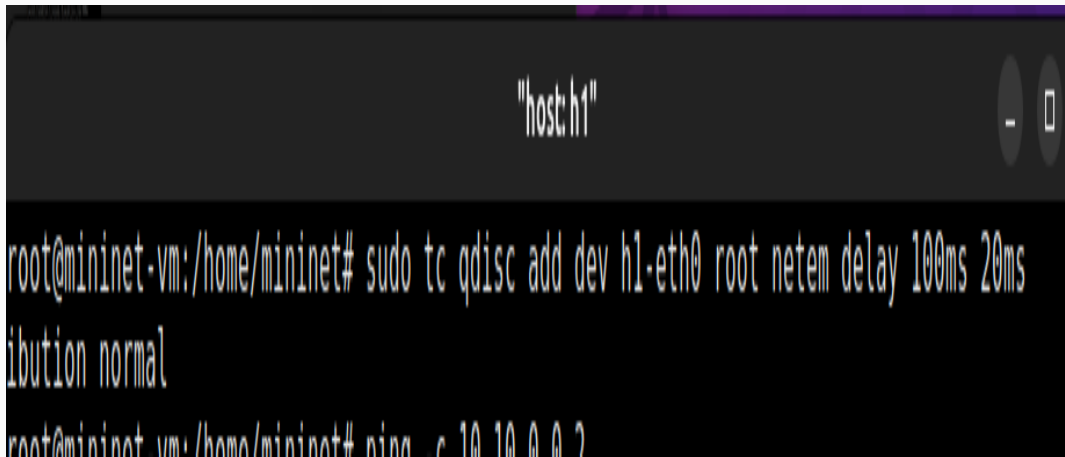


A terminal window with a dark background and a purple title bar. The prompt is `root@mininet-vm:/home/mininet#`. The command `sudo tc qdisc del dev h1-eth0 root netem` is entered. The output `"h1: h1"` is displayed in the center of the terminal.

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

Рис. 16: Восстановление конфигурацию по умолчанию

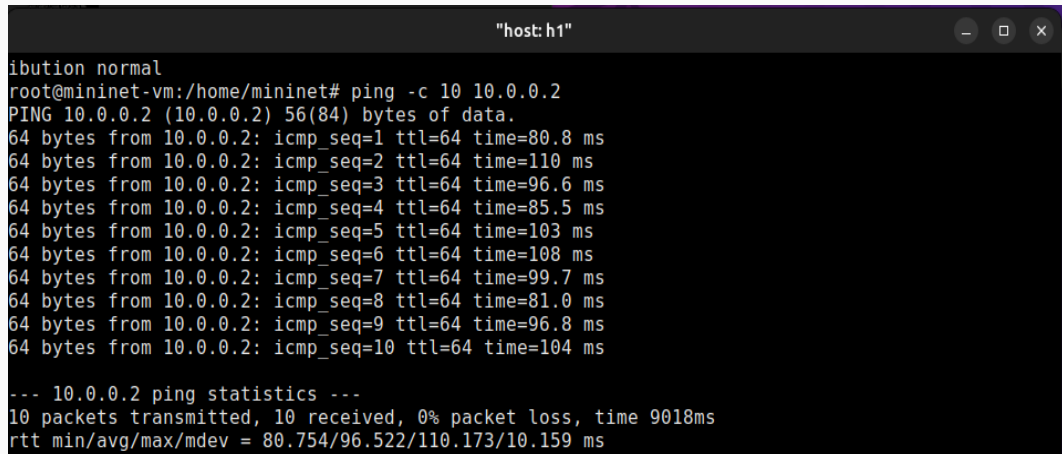
2. Интерактивные эксперименты

A screenshot of a terminal window with a dark background. The title bar at the top is dark purple and contains the text "host: h1" in white, along with standard window control buttons (minimize, maximize, close) on the right. The terminal text is in a light green monospaced font. It shows a user at the root of a mininet VM configuring traffic control on the h1-eth0 interface. The commands entered are: `sudo tc qdisc add dev h1-eth0 root netem delay 100ms 20ms`, `ibution normal`, and `ping -c 10 10.0.0.2`.

```
host: h1  
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms 20ms  
ibution normal  
root@mininet-vm:/home/mininet# ping -c 10 10.0.0.2
```

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

2. Интерактивные эксперименты

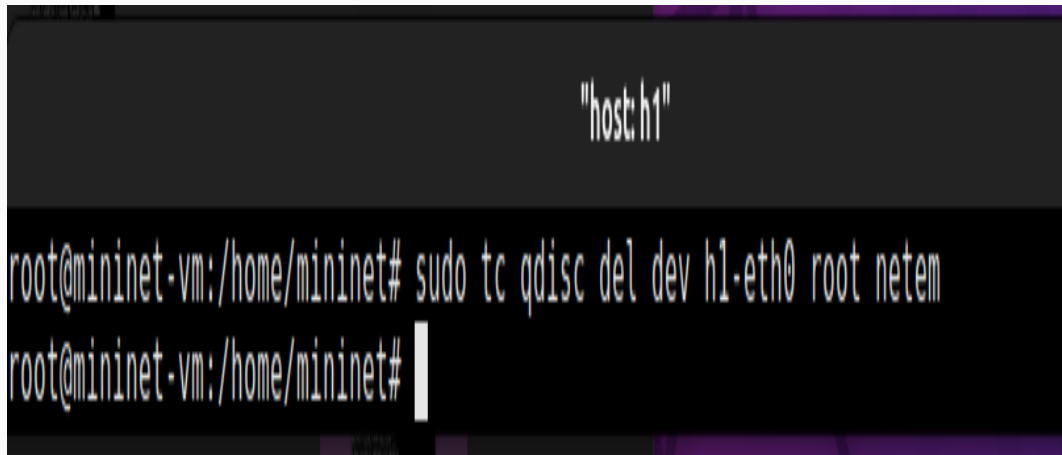
A terminal window titled "host: h1" with standard window controls. It displays the output of a ping command. The text is as follows:

```
ibution 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=80.8 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=110 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=96.6 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=85.5 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=103 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=108 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=99.7 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=81.0 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=96.8 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=104 ms

--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9018ms
rtt min/avg/max/mdev = 80.754/96.522/110.173/10.159 ms
```

Рис. 18: Проверка

2. Интерактивные эксперименты

A terminal window with a dark background and light-colored text. The text is as follows:

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

The prompt is followed by a white cursor bar.

Рис. 19: Восстановление конфигурацию по умолчанию

2. Интерактивные эксперименты

```
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 8 terms
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 1247.365 seconds
mininet@mininet-vm:~$
```

Рис. 20: Завершение работы mininet в интерактивном режиме

3. Воспроизведение экспериментов

```
mininet@mininet-vm:~$ sudo apt-get update
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [128 kB]
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/main i386 Packages [835 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [128 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [3,304 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [3,681 kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [484 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [1,016 kB]
Get:10 http://security.ubuntu.com/ubuntu focal-security/universe i386 Packages [683 kB]
Get:11 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [215 kB]
Get:12 http://us.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [1,056 kB]
Get:13 http://us.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [564 kB]
Get:14 http://us.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [3,397 kB]
Get:15 http://us.archive.ubuntu.com/ubuntu focal-updates/restricted i386 Packages [40.4 kB]
Get:16 http://us.archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [474 kB]
Get:17 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1,238 kB]
Get:18 http://us.archive.ubuntu.com/ubuntu focal-updates/universe i386 Packages [809 kB]
Get:19 http://us.archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [297 kB]
Get:20 http://us.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [27.9 kB]
Get:21 http://us.archive.ubuntu.com/ubuntu focal-updates/multiverse Translation-en [7,968 B]
Fetched 18.5 MB in 6s (2,880 kB/s)
Reading package lists... Done
```

Рис. 21: Обновление репозитория ПО на ВМ

3. Воспроизведение экспериментов

```
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-x11 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 gir1.2-accountsservice-1.0 gir1.2-atk-1.0 gir1.2-atspi-2.0
  gir1.2-freedesktop gir1.2-gck-1 gir1.2-gcr-3 gir1.2-gdesktopenums-3.0 gir1.2-gdkpixbuf-2.0
  gir1.2-gdm-1.0 gir1.2-geoclue-2.0 gir1.2-gnomebluetooth-1.0 gir1.2-gnomedesktop-3.0
  gir1.2-graphene-1.0 gir1.2-gtk-3.0 gir1.2-gweather-3.0 gir1.2-ibus-1.0 gir1.2-json-1.0
  gir1.2-mutter-6 gir1.2-nm-1.0 gir1.2-nma-1.0 gir1.2-notify-0.7 gir1.2-packagekitglib-1.0
  gir1.2-pango-1.0 gir1.2-polkit-1.0 gir1.2-rsvg-2.0 gir1.2-secret-1 gir1.2-soup-2.4
  gir1.2-upowerglib-1.0 gir1.2-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 libaa1 libaacs0 libaom0
  libappindicator3-1 libappstream4 libasound2-plugins libass9 libavahi-core7 libavahi-glib1
```

Рис. 22: Установка пакета geeqie

3. Воспроизведение экспериментов

```
Processing triggers for sgml-base (1.29.1) ...  
mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_i/expname  
mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_i/simple-dele
```

Рис. 23: Создание каталога

3. Воспроизведение экспериментов

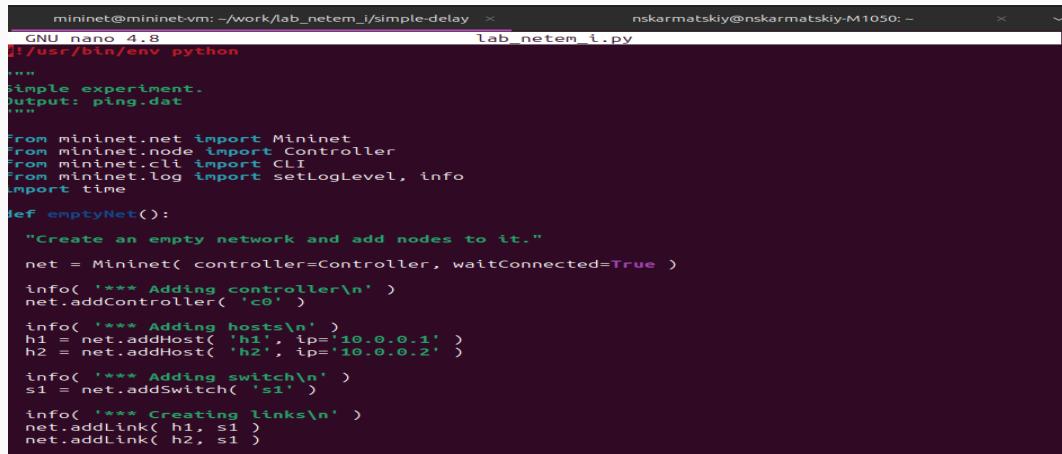
```
mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_i/simple-delay
mininet@mininet-vm:~$ cd ~/work/lab_netem_i/simple-delay
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ touch lab_netem_i.py
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nano lab_netem_i.py

Use "fg" to return to nano.

[1]+  Stopped                  nano lab_netem_i.py
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nano lab_netem_i.py
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nano lab_netem_i.py
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nano lab_netem_i.py
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ ls
lab_netem_i.py
```

Рис. 24: Создание каталога simple-delay

3. Воспроизведение экспериментов



The screenshot shows a terminal window with two tabs. The active tab is titled 'mininet@mininet-vm: ~/work/lab_netem_i/simple-delay'. The terminal displays the GNU nano 4.8 editor interface with the file 'lab_netem_i.py' open. 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 )
```

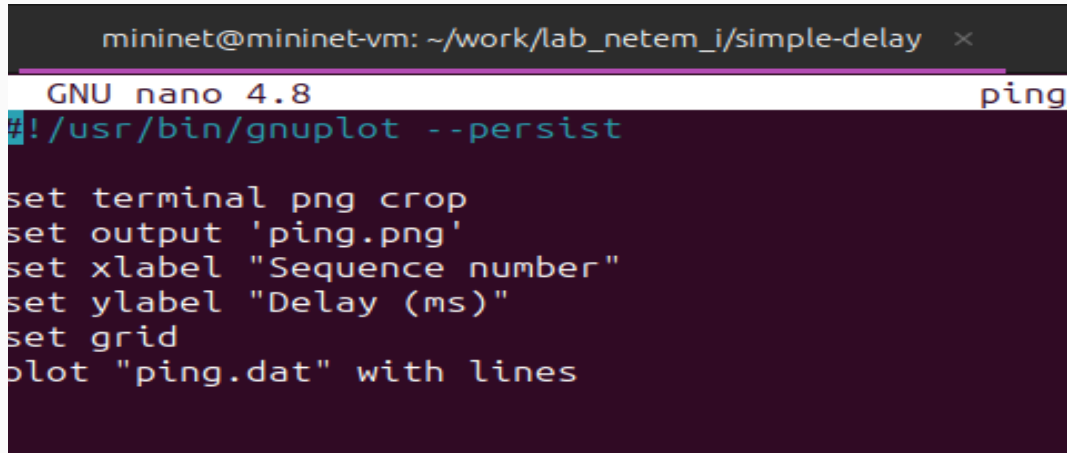
Рис. 25: Создание скрипта для эксперимента lab_netem_i.py

3. Воспроизведение экспериментов

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

Рис. 26: Создание файла ping_plot

3. Воспроизведение экспериментов



```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay ×
GNU nano 4.8 ping
#!/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

3. Воспроизведение экспериментов

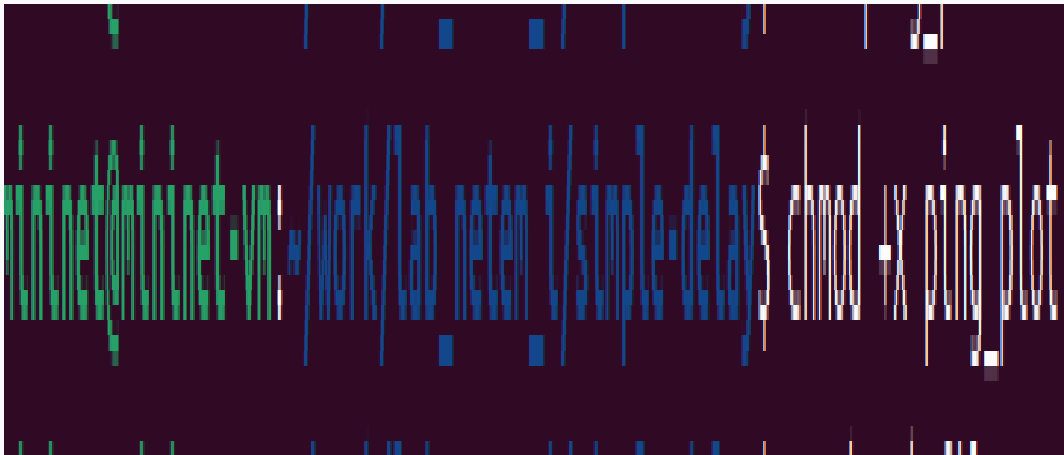
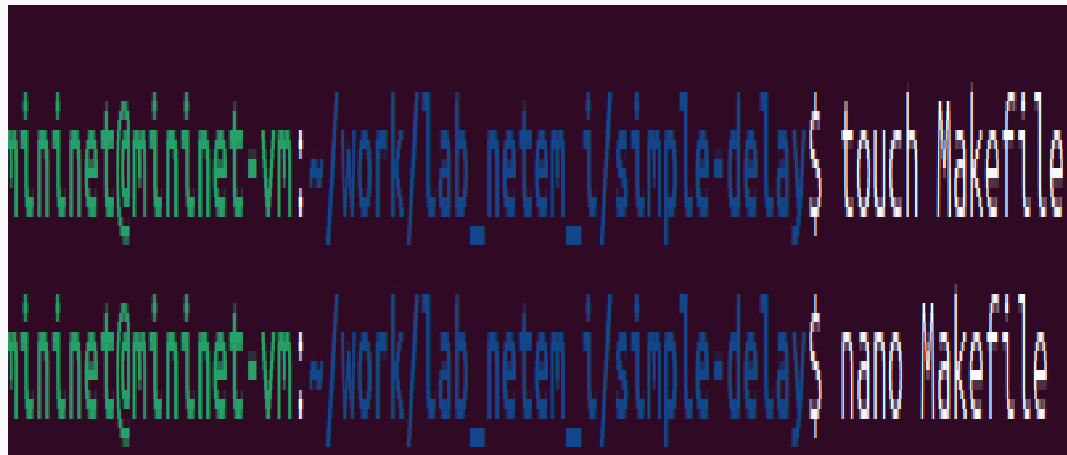


Рис. 28: Настройка прав доступа к файлу скрипта

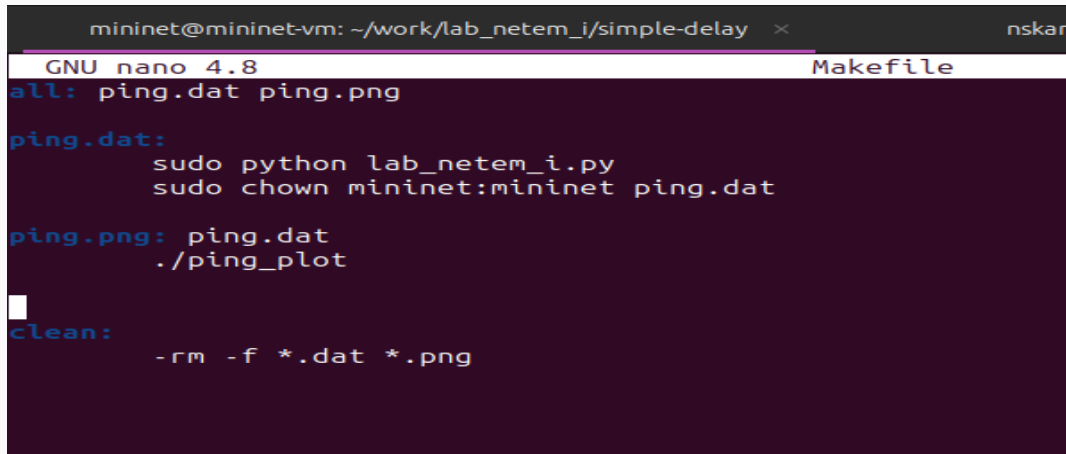
3. Воспроизведение экспериментов

A terminal window with a dark background. The prompt is 'mininet@mininet-vm: ~ /work/lab_netem_i/simple-delay\$'. The command 'touch Makefile' is entered and executed. The prompt is shown again, followed by the command 'nano Makefile' which is also entered and executed.

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

Рис. 29: Создание файла Makefile

3. Воспроизведение экспериментов



The image shows a terminal window with a dark background. At the top, the terminal title bar reads "mininet@mininet-vm: ~/work/lab_netem_i/simple-delay" with a close button on the right. Below the title bar, the text "GNU nano 4.8" is visible on the left and "Makefile" on the right. The main content of the terminal is a Makefile with the following rules:

```
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

3. Воспроизведение экспериментов

```
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, $7}\'' | sed -e \'/s/time=//g\' -e \'/s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ ls
```

Рис. 31: Выполнение эксперимента

3. Воспроизведение экспериментов

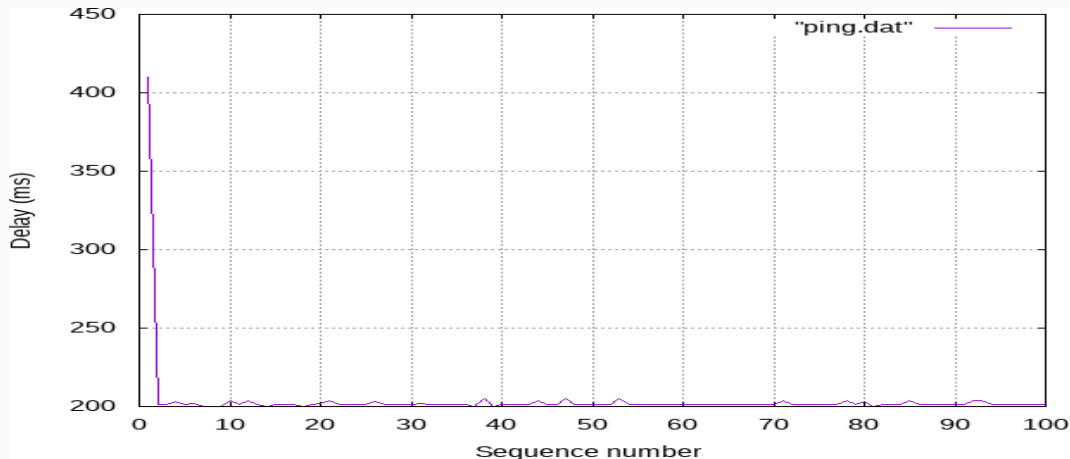
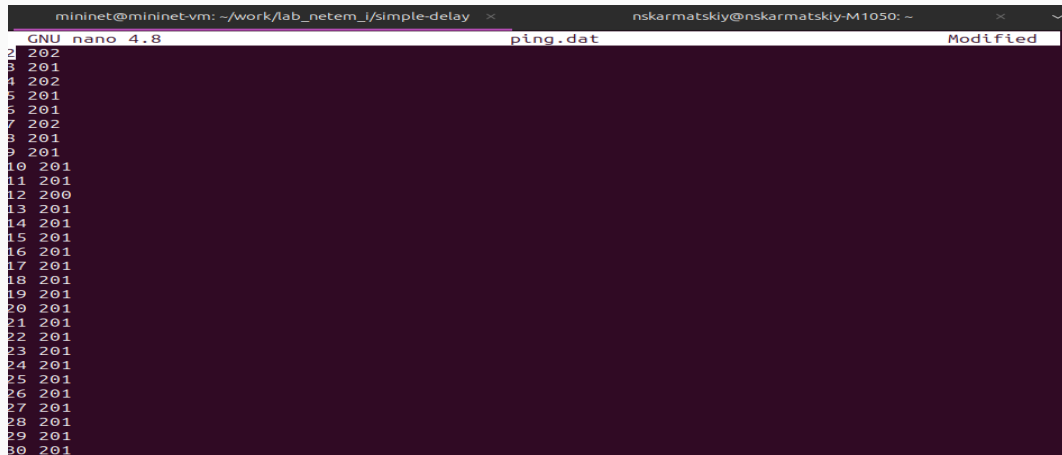


Рис. 32: Просмотр графика

3. Воспроизведение экспериментов



```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay x nskarmatskiy@nskarmatskiy-M1050: ~ x v
GNU nano 4.8 ping.dat Modified
2 202
3 201
4 202
5 201
6 201
7 202
8 201
9 201
10 201
11 201
12 200
13 201
14 201
15 201
16 201
17 201
18 201
19 201
20 201
21 201
22 201
23 201
24 201
25 201
26 201
27 201
28 201
29 201
30 201
```

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

3. Воспроизведение экспериментов

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make ping.png  
./ping_plot  
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ ls  
lab_netem_i.py  Makefile  ping.dat  ping_plot  ping.png
```

Рис. 34: Повторное построение графика

3. Воспроизведение экспериментов

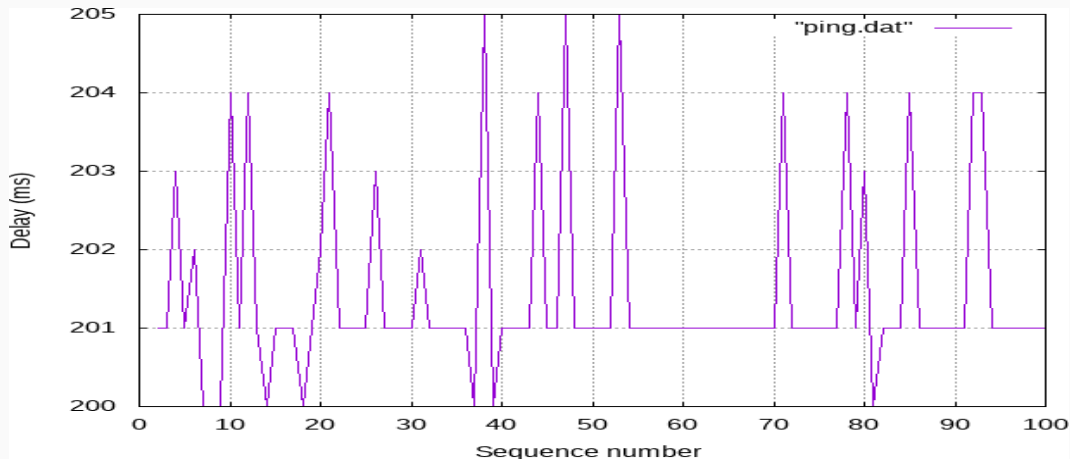
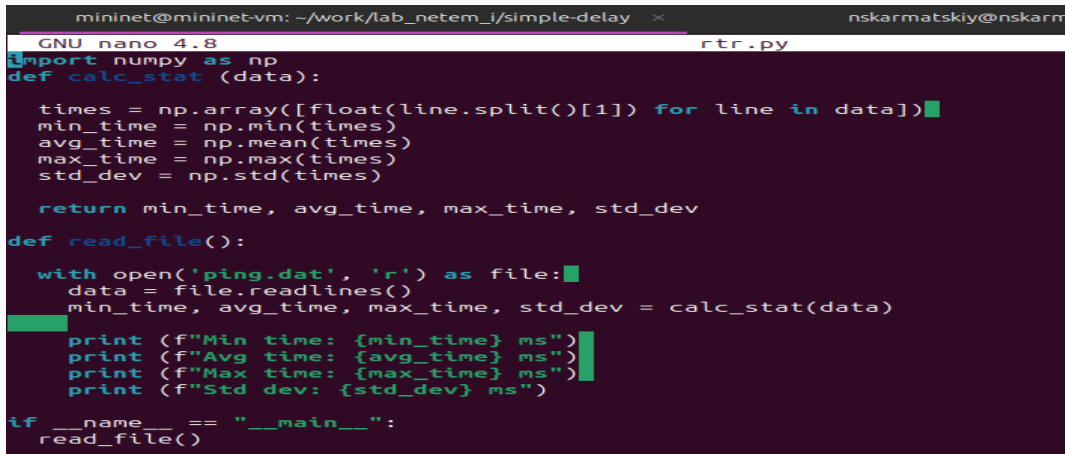


Рис. 35: Просмотр графика

3. Воспроизведение экспериментов



```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay x nskarmatskiy@nskarm
GNU nano 4.8 rtr.py
import numpy as np
def calc_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():

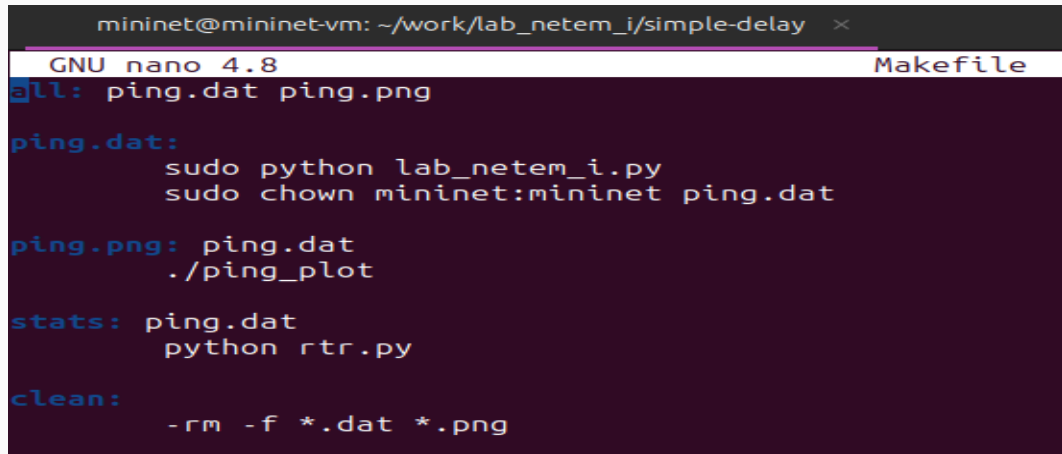
    with open('ping.dat', 'r') as file:
        data = file.readlines()
        min_time, avg_time, max_time, std_dev = calc_stat(data)

        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

3. Воспроизведение экспериментов

A screenshot of a terminal window with a dark background. The title bar at the top reads 'mininet@mininet-vm: ~/work/lab_netem_i/simple-delay' with a close button on the right. Below the title bar, a white bar indicates the editor is 'GNU nano 4.8' and the file is 'Makefile'. The main area shows the content of the Makefile in a monospaced font. The rules are: 'all' depends on 'ping.dat' and 'ping.png'; 'ping.dat' is created by running 'sudo python lab_netem_i.py' and 'sudo chown mininet:mininet ping.dat'; 'ping.png' depends on 'ping.dat' and is created by running './ping_plot'; 'stats' depends on 'ping.dat' and is created by running 'python rtr.py'; 'clean' removes all '.dat' and '.png' files.

```
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

stats: ping.dat
    python rtr.py

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

Рис. 37: Добавление правила запуска скрипта в Makefil

3. Воспроизведение экспериментов

```
Processing triggers for libc-bin (2.15-0ubuntu2) ...  
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make stats  
python rtr.py  
Min time: 200.0 ms  
Avg time: 201.40404040404042 ms  
Max time: 205.0 ms  
Std dev: 1.1713688092383212 ms
```

Рис. 38: Проверка

3. Воспроизведение экспериментов

```
GNU nano 4.8                                lab_netem_i.py                                Modified
"""
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' )
```

Рис. 39: Воспроизводимый эксперимент по изменению задержки

3. Воспроизведение экспериментов

```
mininet@mininet-vm:~/work/lab_netem_t/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, $7}\'' | sed -e \'/time
e=//g\' -e \'/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make stats
python rtr.py
Min time: 100.0 ms
Avg time: 102.17 ms
Max time: 202.0 ms
Std dev: 10.098569205585509 ms
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ ls
lab_netem_i.py  Makefile  ping.dat  ping_plot  ping.png  rtr.py
```

Рис. 40: Воспроизводимый эксперимент по изменению задержки

3. Воспроизведение экспериментов

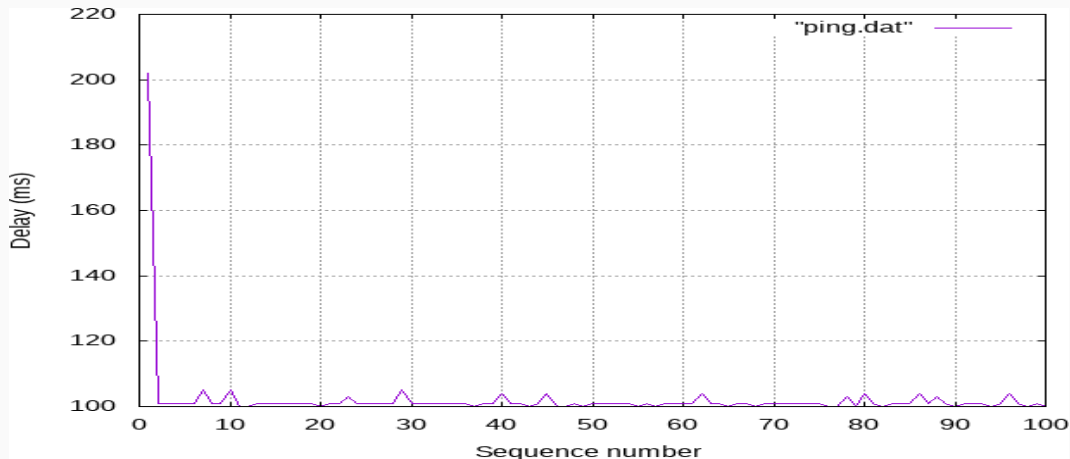
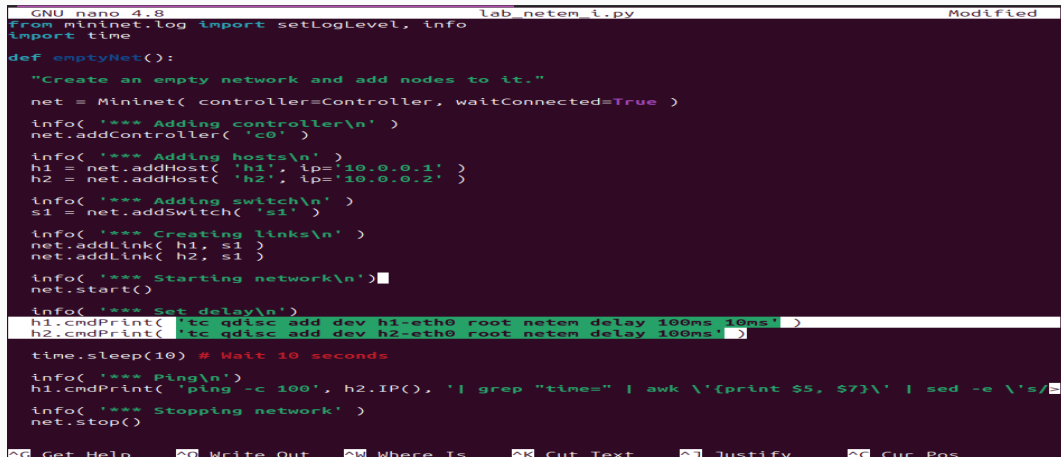


Рис. 41: Просмотр графика

3. Воспроизведение экспериментов



```
GNU nano 4.8 lab_netem_i.py Modified
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' )

    time.sleep(10) # Wait 10 seconds

    info( '*** Ping\n' )
    h1.cmdPrint( 'ping -c 100', h2.IP(), '| grep "time=" | awk \'{print $5, $7}\'' | sed -e 's/ /& /g' )

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

Рис. 42: Воспроизводимый эксперимент по изменению джиттера

3. Воспроизведение экспериментов

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make clean
rm -f *.dat *.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
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, $7}\\' | sed -e \\'s/tim
e=//g\\' -e \\'s/icmp_seq=//g\\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make stats
python rtr.py
Min time: 191.0 ms
Avg time: 202.9 ms
Max time: 409.0 ms
Std dev: 71.5 ms
```

Рис. 43: Воспроизводимый эксперимент по изменению джиттера

3. Воспроизведение экспериментов

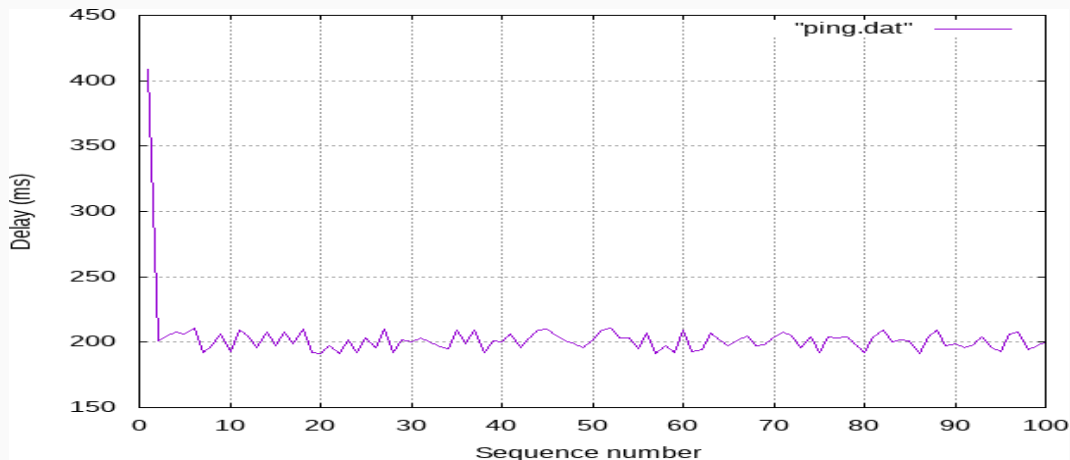
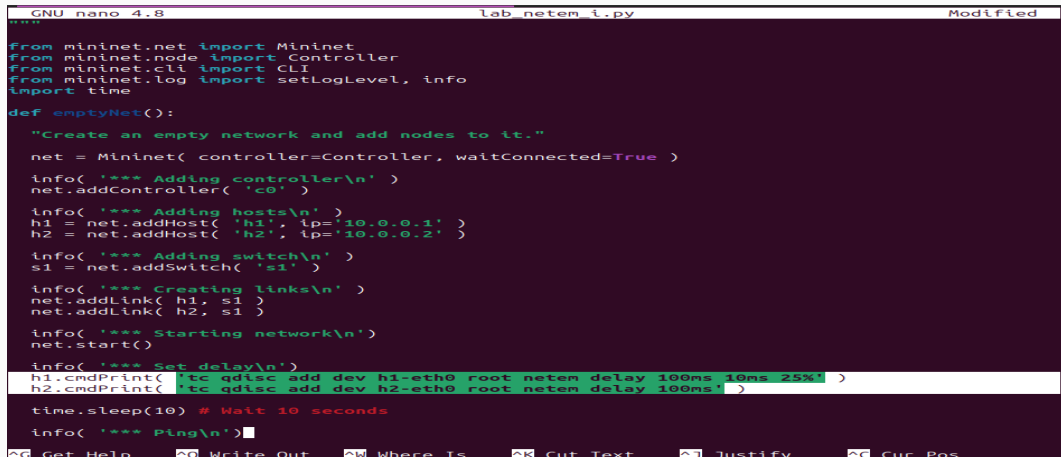


Рис. 44: Просмотр графика

3. Воспроизведение экспериментов



```
GNU nano 4.8 lab_netem_i.py Modified
***
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' )
```

Рис. 45: Воспроизводимый эксперимент по изменению значения корреляции для джиттера и задержки

3. Воспроизведение экспериментов

```
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, $7}\'' | sed -e \'/s/tim
e=//g\' -e \'/s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-del$ make stats
python rtr.py
Min time: 191.0 ms
Avg time: 203.46 ms
Max time: 406.0 ms
Std dev: 21.166681364824292 ms
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$
```

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

3. Воспроизведение экспериментов

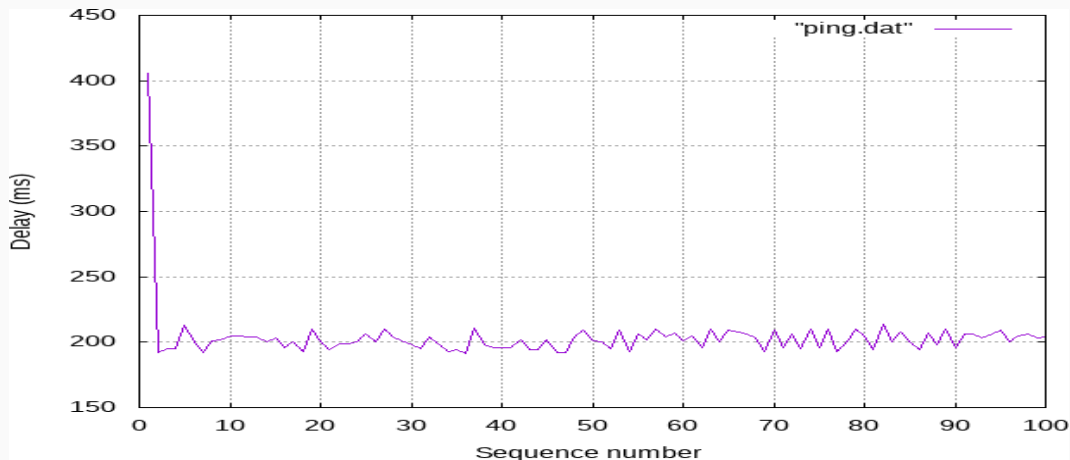


Рис. 47: Просмотр графика

3. Воспроизведение экспериментов



```
GNU nano 4.8 lab_netem_i.py
"""
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' )
    
```

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

3. Воспроизведение экспериментов

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nano lab_netem_i.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
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, $7}\'' | sed -e \\'s/time
g\' -e \\'s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make stats
python rtr.py
Min time: 179.0 ms
Avg time: 203.25 ms
Max time: 386.0 ms
Std dev: 21.171856319179952 ms
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$
```

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

3. Воспроизведение экспериментов

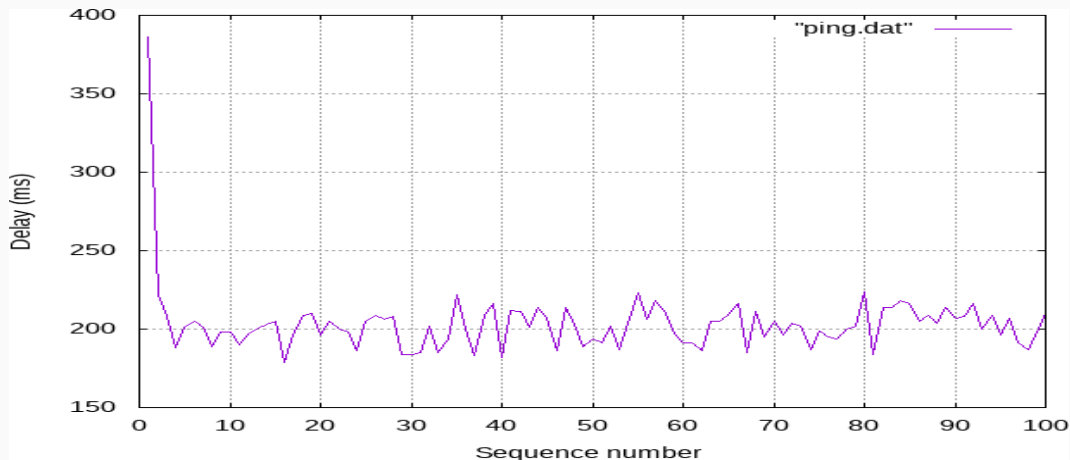


Рис. 50: Просмотр графика

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.

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