

Отчёт по лабораторной работе №4

Моделирование сетей передачи данных

Эмуляция и измерение задержек в глобальных сетях

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

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Вывод	39
4	Список литературы. Библиография	40

Список иллюстраций

2.1	Исправление прав запуска X-соединения в виртуальной машине mininet	6
2.2	Создание простейшей топологии	7
2.3	Отображение информации их сетевых интерфейсов и IP-адресов	8
2.4	Проверка подключения между хостами h1 и h2	9
2.5	Добавление задержки в 100 мс к выходному интерфейсу на хосте h1	9
2.6	Проверка	10
2.7	Добавление задержки в 100 мс к выходному интерфейсу на хосте h2	10
2.8	Проверка	10
2.9	Изменение задержки со 100 мс до 50 мс	11
2.10	Проверка	11
2.11	Восстановление конфигураций по умолчанию	11
2.12	Добавление на узле h1 задержки в 100 мс со случайным отклонением 10 мс	12
2.13	Проверка	12
2.14	Восстановление конфигурации интерфейса по умолчанию	12
2.15	Проверка	13
2.16	Восстановление конфигурации интерфейса по умолчанию	13
2.17	Настройка нормального распределения задержки на узле h1 в эмулируемой сети	14
2.18	Проверка	14
2.19	Восстановление конфигурации интерфейса по умолчанию	14
2.20	Завершение работы mininet в интерактивном режиме	15
2.21	Обновление репозитория программного обеспечения на виртуальной машине	16
2.22	Установка пакета gnet	17
2.23	Создание нового каталога	17
2.24	Создание каталога simple-delay	18
2.25	Создание скрипта lab_netem_i.py для эксперимента	18
2.26	Создание файла ping_plot	19
2.27	Создание скрипта ping_plot для визуализации результатов эксперимента	19
2.28	Настройка прав доступа к файлу скрипта	19
2.29	Создание файла Makefile	19
2.30	Добавления скрипта в Makefile для управления процессом проведения эксперимента	20
2.31	Выполнение эксперимента	21

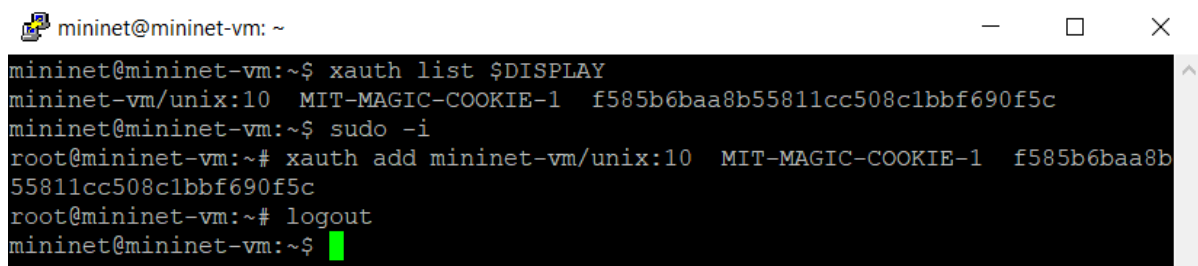
2.32	Просмотр графика	22
2.33	Удаление первой строчки из файла ping.dat	23
2.34	Повторное построение графика	23
2.35	Просмотр графика	24
2.36	Разработка скрипта для вычисления на основе данных файла ping.dat минимального, среднего, максимального и стандартного отклонения времени приёма-передачи	25
2.37	Добавление правила запуска скрипта в Makefil	26
2.38	Проверка	26
2.39	Воспроизводимый эксперимент по изменению задержки	27
2.40	Воспроизводимый эксперимент по изменению задержки	28
2.41	Просмотр графика	29
2.42	Воспроизводимый эксперимент по изменению джиттера	30
2.43	Воспроизводимый эксперимент по изменению джиттера	31
2.44	Просмотр графика	32
2.45	Воспроизводимый эксперимент по изменению значения корреля- ции для джиттера и задержки	33
2.46	Воспроизводимый эксперимент по изменению значения корреля- ции для джиттера и задержки	34
2.47	Просмотр графика	35
2.48	Воспроизводимый эксперимент по изменению распределения вре- мени задержки в эмулируемой глобальной сети	36
2.49	Воспроизводимый эксперимент по изменению распределения вре- мени задержки в эмулируемой глобальной сети	37
2.50	Просмотр графика	38

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

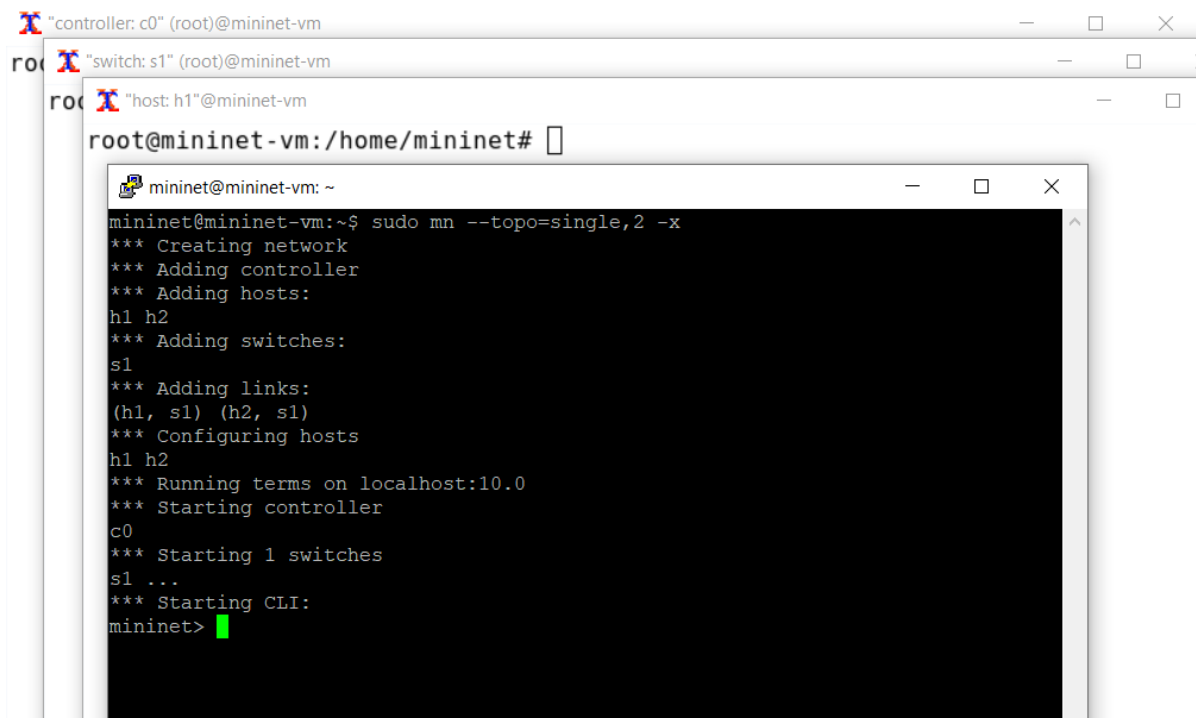
В виртуальной машине mininet исправим права запуска X-соединения (рис. 2.1):



```
mininet@mininet-vm: ~  
mininet@mininet-vm:~$ xauth list $DISPLAY  
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 f585b6baa8b55811cc508c1bbf690f5c  
mininet@mininet-vm:~$ sudo -i  
root@mininet-vm:~# xauth add mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 f585b6baa8b55811cc508c1bbf690f5c  
root@mininet-vm:~# logout  
mininet@mininet-vm:~$
```

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

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

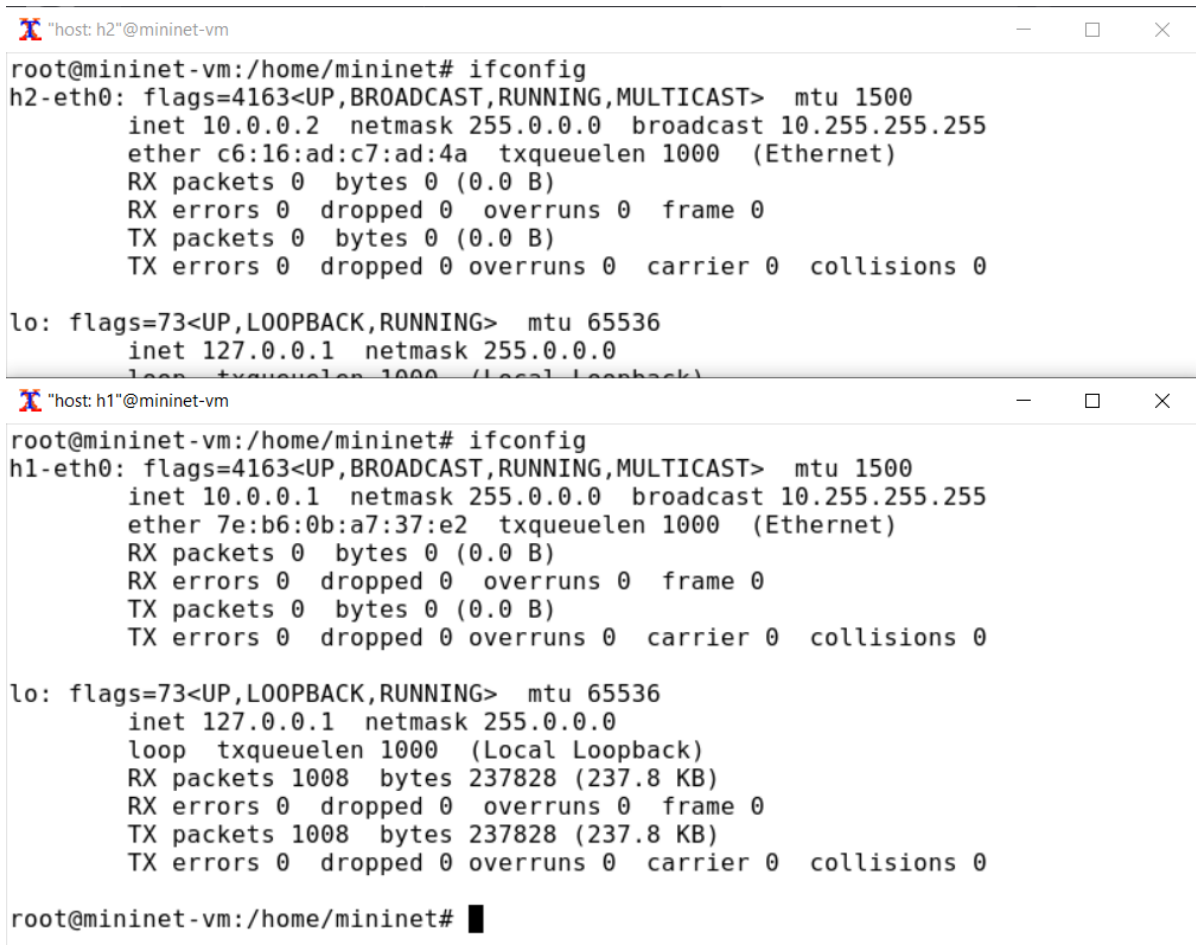


The image shows a series of overlapping terminal windows from a Mininet environment. The top window is titled "controller: c0" (root)@mininet-vm. Below it is "switch: s1" (root)@mininet-vm. The third window is titled "host: h1" (root)@mininet-vm. The main window in the foreground is titled "root@mininet-vm:/home/mininet#" and shows the execution of the command `sudo mn --topo=single,2 -x`. The output of this command is as follows:

```
mininet@mininet-vm:~$ sudo mn --topo=single,2 -x
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Running terms on localhost:10.0
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

Рис. 2.2: Создание простейшей топологии

На хостах h1 и h2 введём команду `ifconfig`, чтобы отобразить информацию, относящуюся к их сетевым интерфейсам и назначенным им IP-адресам. В дальнейшем при работе с NETEM и командой `tc` будут использоваться интерфейсы h1-eth0 и h2-eth0 (рис. 2.3):



The image displays two terminal windows from a Mininet VM. The top window is titled "host: h2" and shows the output of the 'ifconfig' command for host h2. It lists the h2-eth0 interface with IP 10.0.0.2 and the loopback interface lo with IP 127.0.0.1. The bottom window is titled "host: h1" and shows the output of 'ifconfig' for host h1. It lists the h1-eth0 interface with IP 10.0.0.1 and the loopback interface lo with IP 127.0.0.1. Both windows show standard network statistics like RX/TX packets, errors, and collisions.

```
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 c6:16:ad:c7:ad:4a txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

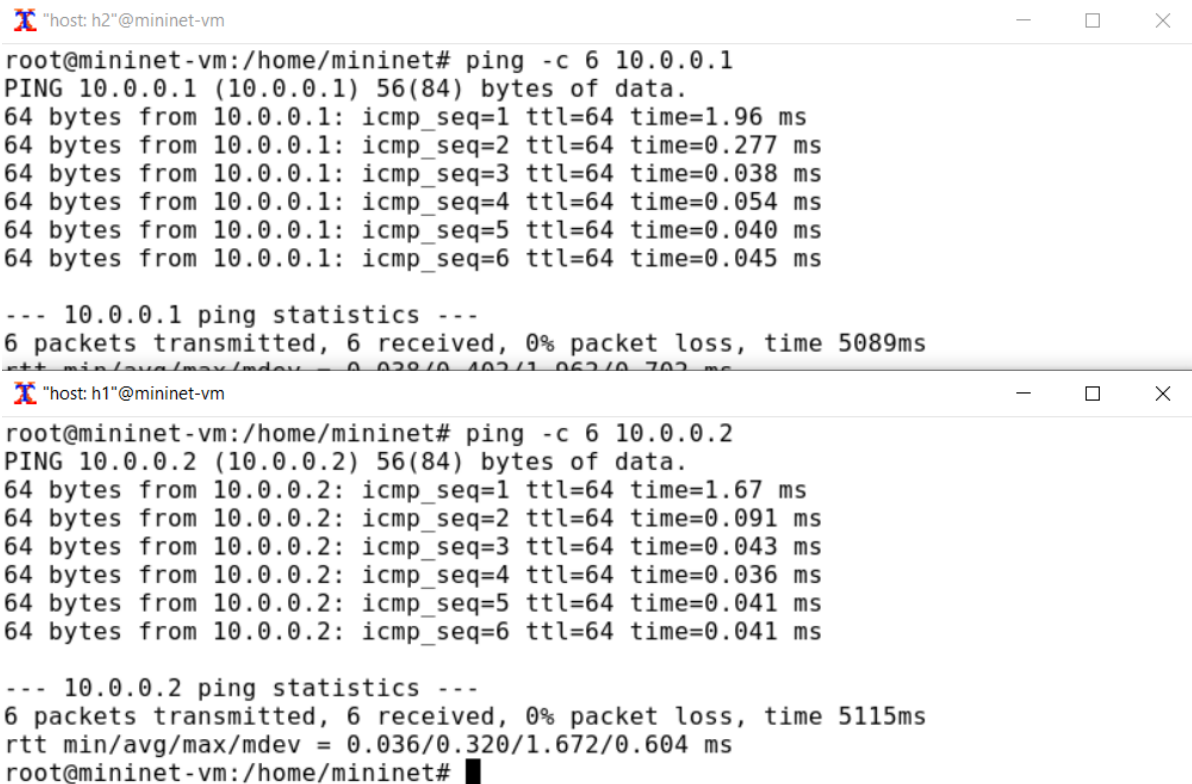
```
root@mininet-vm:/home/mininet# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.1 netmask 255.0.0.0 broadcast 10.255.255.255
    ether 7e:b6:0b:a7:37:e2 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    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 1008 bytes 237828 (237.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1008 bytes 237828 (237.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

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

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



The image shows two terminal windows from a Mininet VM. The top window, titled "host: h2" @mininet-vm, shows a ping command from h2 to 10.0.0.1. It displays six successful ping responses with varying times (1.96 ms to 0.045 ms) and a summary: 6 packets transmitted, 6 received, 0% packet loss, time 5089ms. The bottom window, titled "host: h1" @mininet-vm, shows a ping command from h1 to 10.0.0.2. It also displays six successful ping responses with times (1.67 ms to 0.041 ms) and a summary: 6 packets transmitted, 6 received, 0% packet loss, time 5115ms. Both windows show the command prompt as root@mininet-vm:/home/mininet#.

```
root@mininet-vm:/home/mininet# ping -c 6 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=1.96 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.277 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.038 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.054 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.040 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.045 ms


--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5089ms
rtt min/avg/max/mdev = 0.038/0.402/1.062/0.702 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=1.67 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.091 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.036 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.041 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.041 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5115ms
rtt min/avg/max/mdev = 0.036/0.320/1.672/0.604 ms
root@mininet-vm:/home/mininet#
```

Рис. 2.4: Проверка подключения между хостами h1 и h2

На хосте h1 добавим задержку в 100 мс к выходному интерфейсу (рис. 2.5):



The image shows a terminal window titled "host: h1" @mininet-vm. The user enters the command 'sudo tc qdisc add dev h1-eth0 root netem delay 100ms'. The command is executed successfully, and the prompt returns to root@mininet-vm:/home/mininet#.

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms
root@mininet-vm:/home/mininet#
```

Рис. 2.5: 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. 2.6):

```
"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=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=101 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=100 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=100 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 100.076/100.768/101.502/0.529 ms
root@mininet-vm:/home/mininet#
```

Рис. 2.6: Проверка

Для эмуляции глобальной сети с двунаправленной задержкой необходимо к соответствующему интерфейсу на хосте h2 также добавить задержку в 100 миллисекунд (рис. 2.7):

```
"host: h2"@mininet-vm
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h2-eth0 root netem delay 100ms
root@mininet-vm:/home/mininet#
```

Рис. 2.7: Добавление задержки в 100 мс к выходному интерфейсу на хосте h2

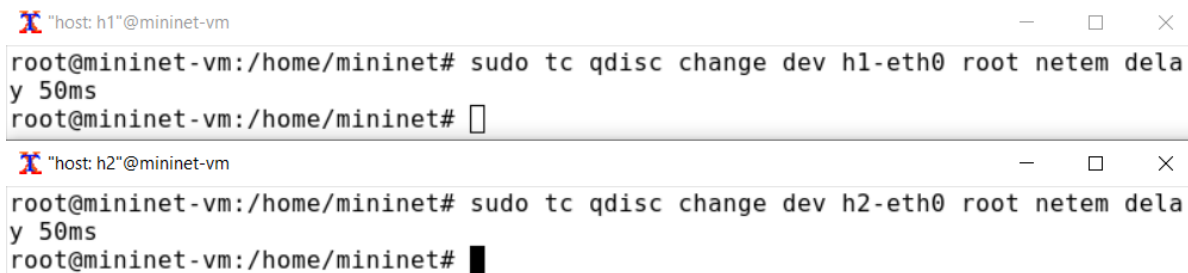
Проверим, что соединение между хостом h1 и хостом h2 имеет RTT в 200 мс (100 мс от хоста h1 к хосту h2 и 100 мс от хоста h2 к хосту h1), повторив команду ping с параметром -c 6 на терминале хоста h1 (рис. 2.8):

```
"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=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=201 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=200 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 5010ms
rtt min/avg/max/mdev = 200.318/200.925/201.423/0.362 ms
root@mininet-vm:/home/mininet#
```

Рис. 2.8: Проверка

Изменим задержку со 100 мс до 50 мс для отправителя h1 и для получателя h2 (рис. 2.9):

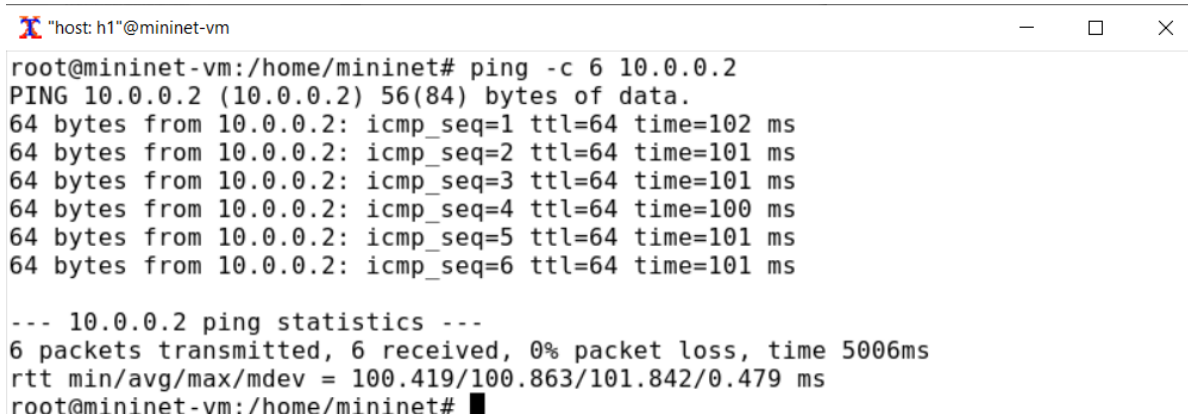


```
"host: h1"@mininet-vm
root@mininet-vm:/home/mininet# sudo tc qdisc change dev h1-eth0 root netem delay 50ms
root@mininet-vm:/home/mininet#

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

Рис. 2.9: 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. 2.10):

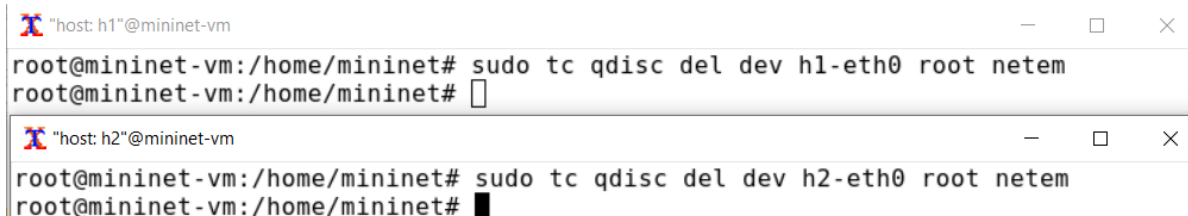


```
"host: h1"@mininet-vm
root@mininet-vm:/home/mininet# ping -c 6 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=102 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=100 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 5006ms
rtt min/avg/max/mdev = 100.419/100.863/101.842/0.479 ms
root@mininet-vm:/home/mininet#
```

Рис. 2.10: Проверка

Восстановим конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса (рис. 2.11):

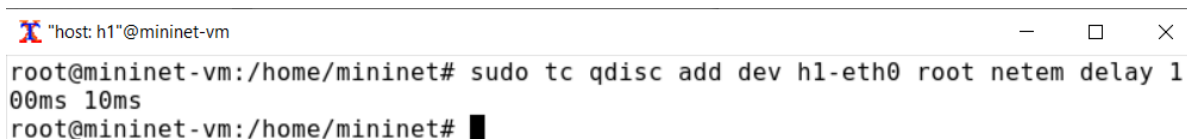


```
"host: h1"@mininet-vm
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
root@mininet-vm:/home/mininet#

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

Рис. 2.11: Восстановление конфигураций по умолчанию

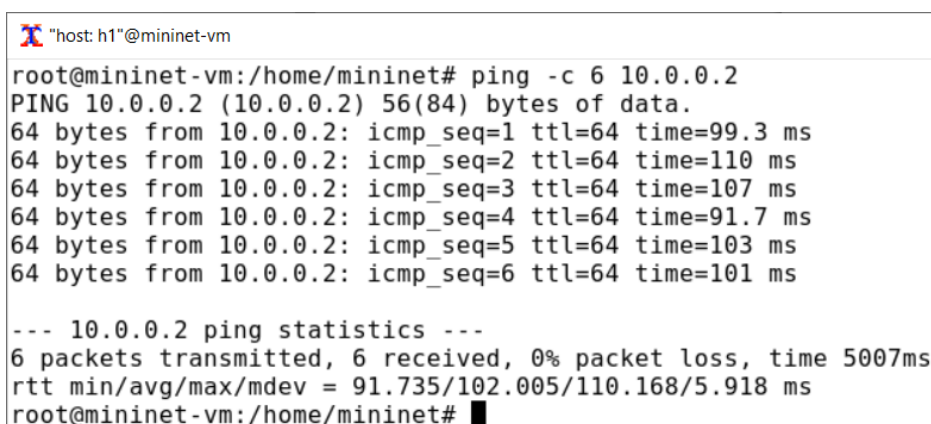
Добавим на узле h1 задержку в 100 мс со случайным отклонением 10 мс (рис. 2.12):



```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms 10ms
root@mininet-vm:/home/mininet#
```

Рис. 2.12: 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. 2.13):

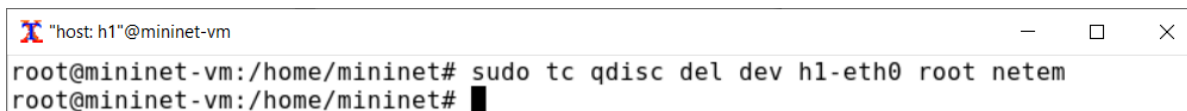


```
root@mininet-vm:/home/mininet# ping -c 6 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=99.3 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=110 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=107 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=91.7 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=101 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5007ms
rtt min/avg/max/mdev = 91.735/102.005/110.168/5.918 ms
root@mininet-vm:/home/mininet#
```

Рис. 2.13: Проверка

Восстановим конфигурацию интерфейса по умолчанию на узле h1 (рис. 2.14):



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

Рис. 2.14: 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. 2.15):

```
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#
```

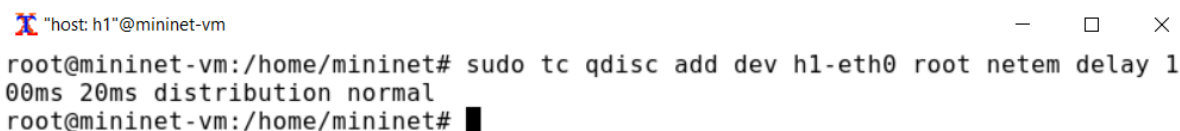
Рис. 2.15: Проверка

Восстановим конфигурацию интерфейса по умолчанию на узле h1 (рис. 2.16):

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

Рис. 2.16: Восстановление конфигурации интерфейса по умолчанию

Зададим нормальное распределение задержки на узле h1 в эмулируемой сети (рис. 2.17):



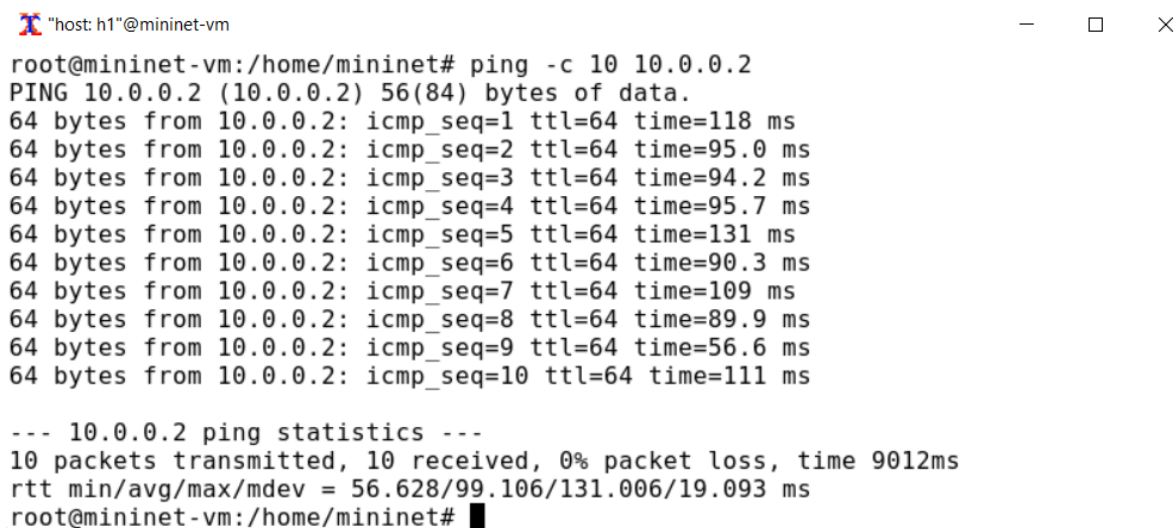
```

"host: h1"@mininet-vm
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. 2.17: 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 \pm 20 мс. Используем для этого команду ping на терминале хоста h1 с параметром -c 10 (рис. 2.18):



```

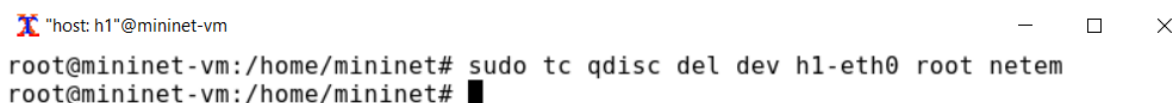
"host: h1"@mininet-vm
root@mininet-vm:/home/mininet# ping -c 10 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=118 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=95.0 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=94.2 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=95.7 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=131 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=90.3 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=109 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=89.9 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=56.6 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=111 ms

--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9012ms
rtt min/avg/max/mdev = 56.628/99.106/131.006/19.093 ms
root@mininet-vm:/home/mininet#

```

Рис. 2.18: Проверка

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



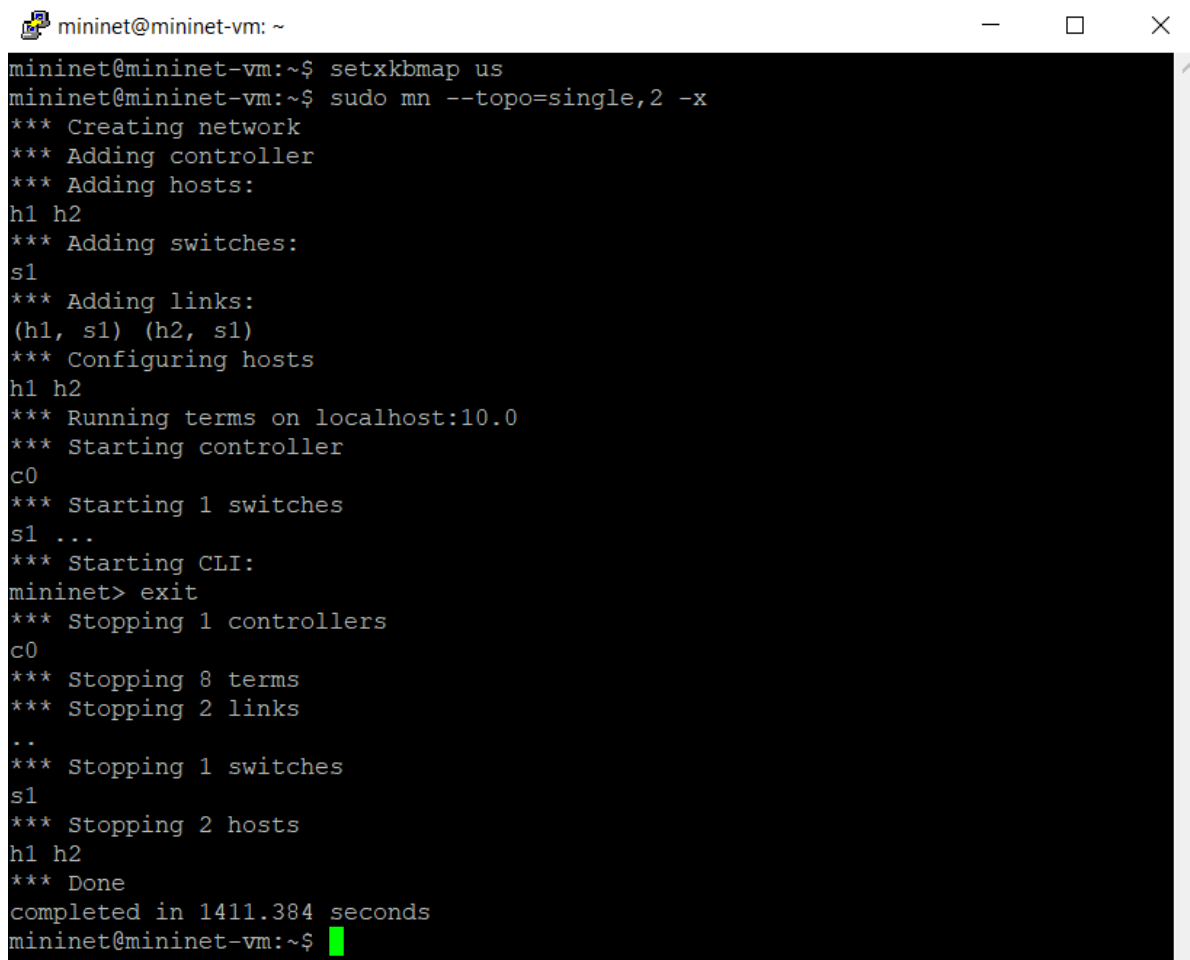
```

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

```

Рис. 2.19: Восстановление конфигурации интерфейса по умолчанию

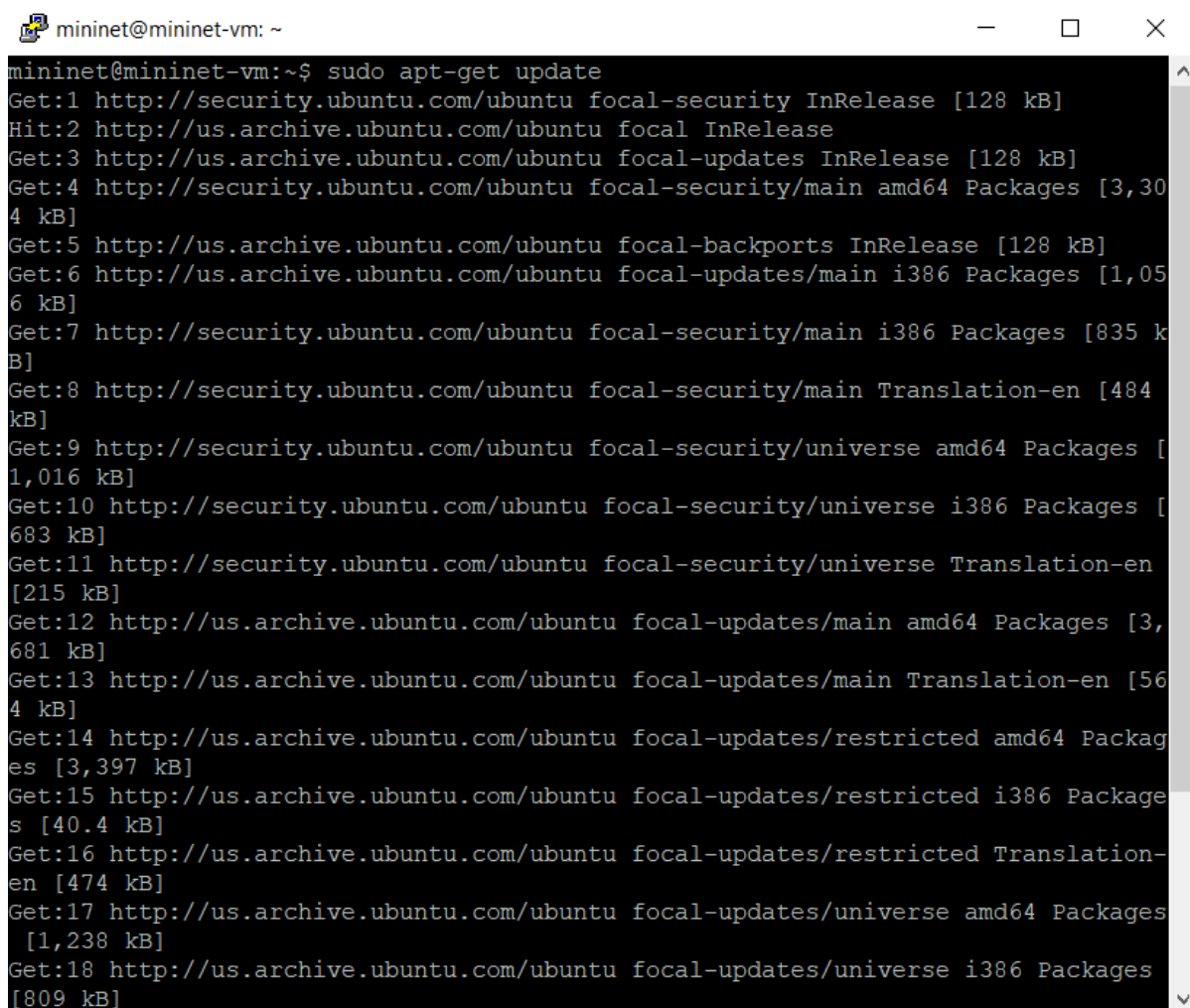
Завершим работу mininet в интерактивном режиме (рис. 2.20):

A terminal window titled 'mininet@mininet-vm: ~' with standard window controls. The terminal shows the execution of 'sudo mn --topo=single,2 -x', which sets up a network topology. It lists the creation of a network, adding a controller (c0), two hosts (h1, h2), and a switch (s1), and connecting them. It then starts the controller, hosts, and switch, and finally stops them all. The process completes in 1411.384 seconds.

```
mininet@mininet-vm:~$ setxkbmap us
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> exit
*** Stopping 1 controllers
c0
*** Stopping 8 terms
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 1411.384 seconds
mininet@mininet-vm:~$
```

Рис. 2.20: Завершение работы mininet в интерактивном режиме

Обновим репозитории программного обеспечения на виртуальной машине (рис. 2.21):



```
mininet@mininet-vm: ~  
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 amd64 Packages [3,304 kB]  
Get:5 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [128 kB]  
Get:6 http://us.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [1,056 kB]  
Get:7 http://security.ubuntu.com/ubuntu focal-security/main i386 Packages [835 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 amd64 Packages [3,681 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]
```

Рис. 2.21: Обновление репозитория программного обеспечения на виртуальной машине

Установим пакет geeeie для просмотра файлов png (рис. 2.22):


```
mininet@mininet-vm: ~  
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 libavc1394-0 libavcodec58 libavfilter7 libavformat58  
  libavutil56 libbdplus0 libbluetooth3 libbluray2 libboost-thread1.71.0  
  libbs2b0 libcac0 libcamel-1.2-62 libcanberra-gtk3-0 libcanberra-gtk3-module  
  libcanberra-pulse libcheese-gtk25 libcheese8 libchromaprint1  
  libclutter-1.0-0 libclutter-1.0-common libclutter-gst-3.0-0
```

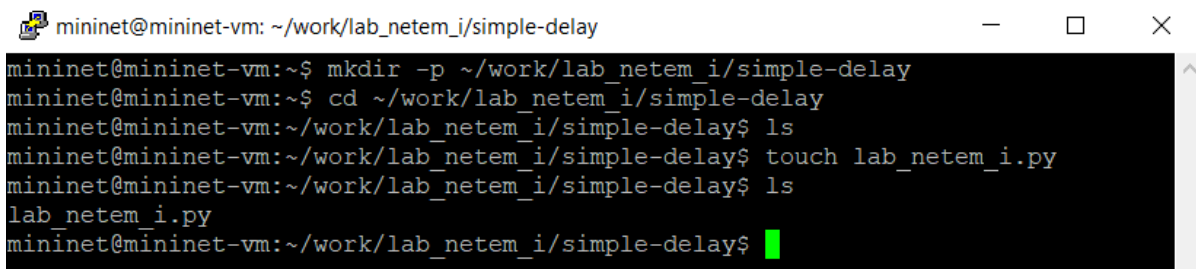
Рис. 2.22: Установка пакета geeqie

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

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

Рис. 2.23: Создание нового каталога

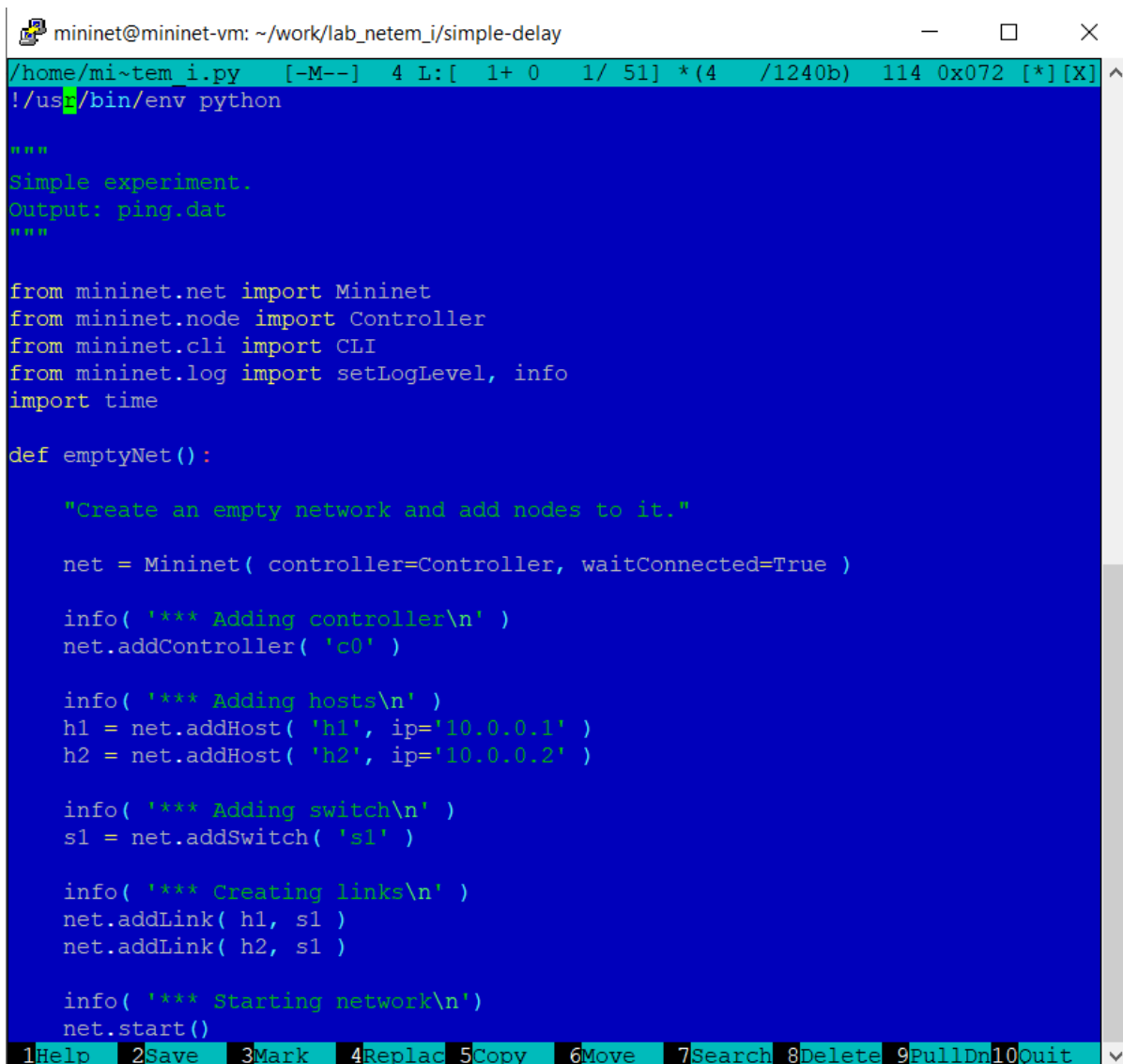
В виртуальной среде mininet в своём рабочем каталоге с проектами создадим каталог simple-delay и перейдём в него (рис. 2.24):



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

Рис. 2.24: Создание каталога simple-delay

Создадим скрипт для эксперимента lab_netem_i.py (рис. 2.25):



```
/home/mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
#!/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()

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

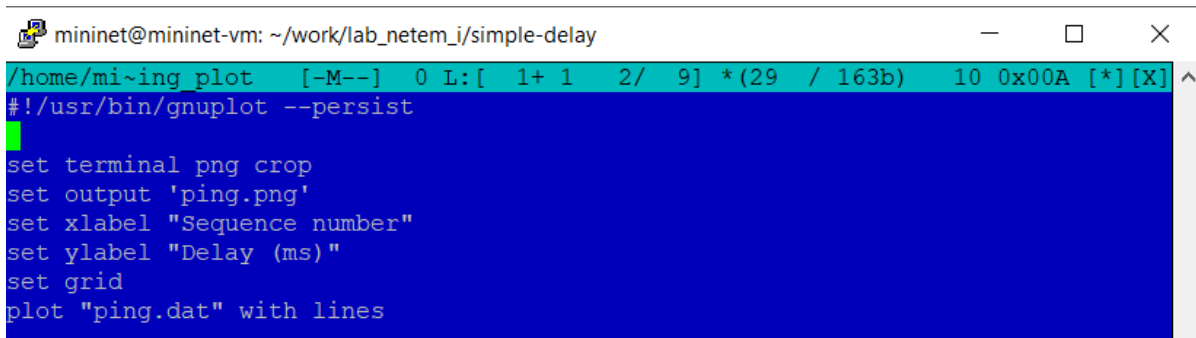
Рис. 2.25: Создание скрипта lab_netem_i.py для эксперимента

Создадим файл `ping_plot` (рис. 2.26):

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

Рис. 2.26: Создание файла `ping_plot`

Затем создадим скрипт для визуализации `ping_plot` результатов эксперимента (рис. 2.27):



```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
/home/mininet@mininet-vm: ~/work/lab_netem_i/simple-delay [~M--] 0 L:[ 1+ 1 2/ 9] *(29 / 163b) 10 0x00A [*] [X] ^
#!/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. 2.27: Создание скрипта `ping_plot` для визуализации результатов эксперимента

Зададим права доступа к файлу скрипта (рис. 2.28):

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

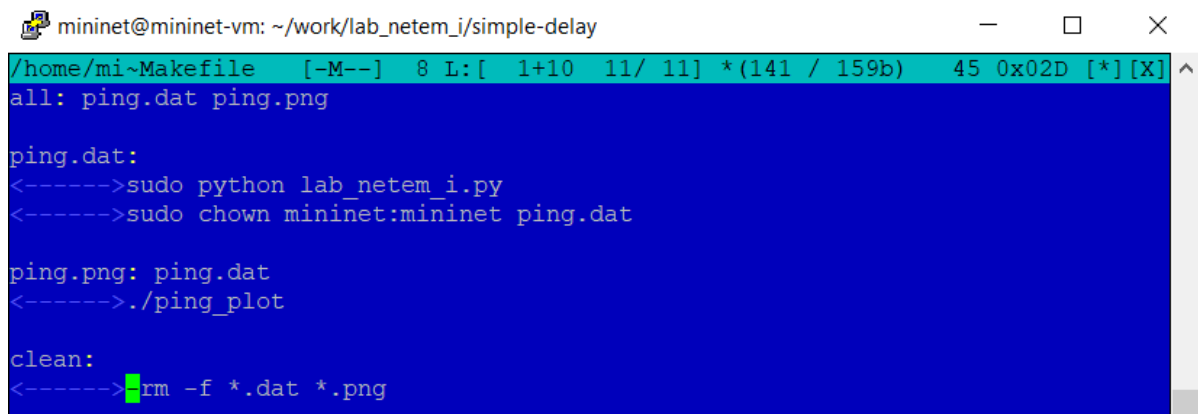
Рис. 2.28: Настройка прав доступа к файлу скрипта

Создадим файла `Makefile` (рис. 2.29):

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

Рис. 2.29: Создание файла `Makefile`

Внутри файла `Makefile` поместим скрипт для управления процессом проведения эксперимента (рис. 2.30):



The screenshot shows a terminal window with a title bar indicating the user is 'mininet' on a 'mininet-vm' machine, located at '~/work/lab_netem_i/simple-delay'. The terminal displays the contents of a Makefile, which defines targets for 'ping.dat' and 'ping.png', and a 'clean' target. The 'ping.dat' target runs 'sudo python lab_netem_i.py' and 'sudo chown mininet:mininet ping.dat'. The 'ping.png' target runs 'ping.dat' and './ping_plot'. The 'clean' target runs 'rm -f *.dat *.png'. The terminal output shows the 'all' target being executed, which builds 'ping.dat' and 'ping.png'.

```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
/home/mi~Makefile  [-M--]  8 L:[ 1+10 11/ 11] *(141 / 159b) 45 0x02D [*][X] ^
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. 2.30: Добавления скрипта в Makefile для управления процессом проведения эксперимента

Выполним эксперимент (рис. 2.31):

```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
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. 2.31: Выполнение эксперимента

Посмотрим построенный в результате выполнения скриптов график (рис. 2.32):

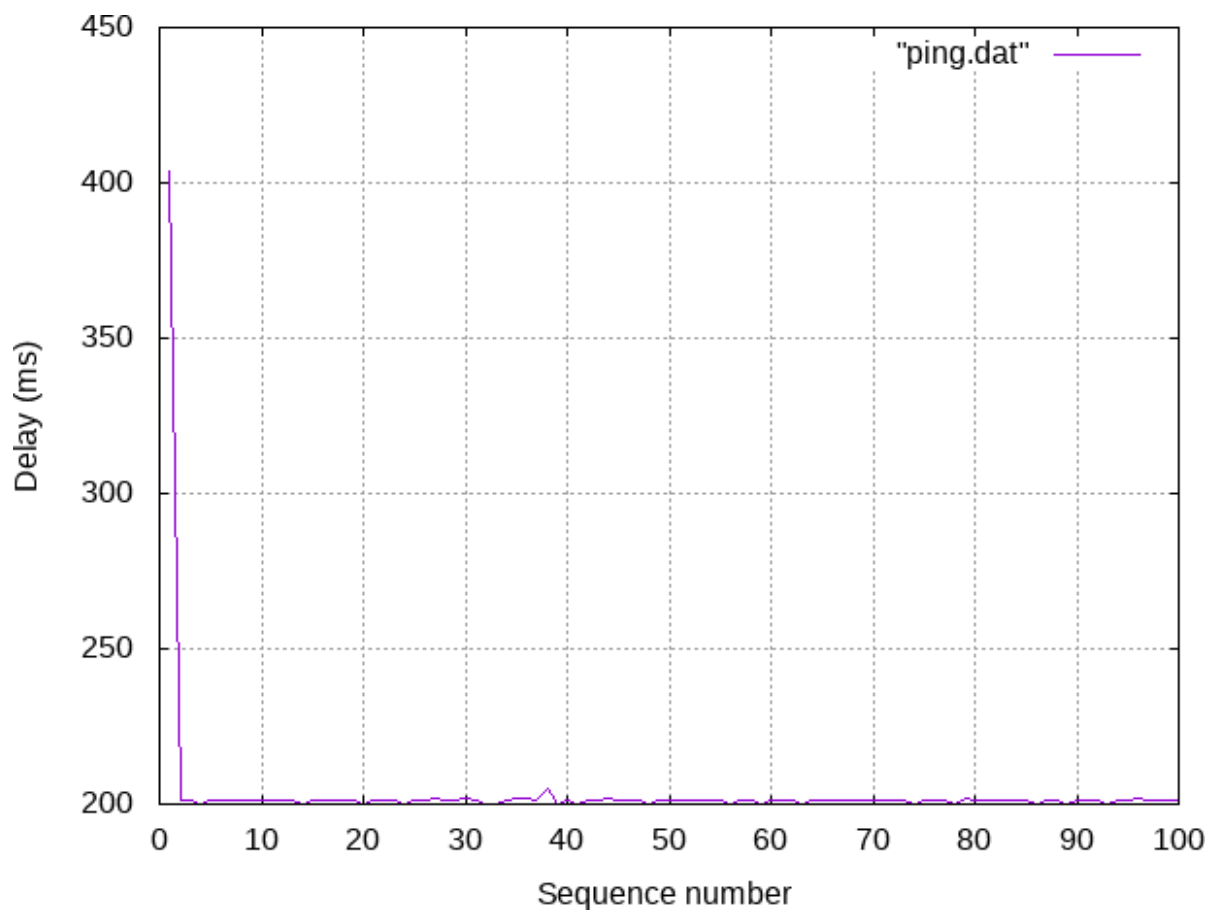


Рис. 2.32: Просмотр графика

Из файла ping.dat удалим первую строку и заново построим график (рис. 2.33 - рис. 2.34):

```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
/home/mi~ping.dat  [----]  0 L:[ 1+ 0 1/100] *(0 / 686b) 50 0x032 [*][X]
2 201
3 201
4 200
5 201
6 201
7 201
8 201
9 201
10 201
11 201
12 201
13 201
14 200
15 201
16 201
17 201
18 201
19 201
20 200
21 201
22 201
23 201
24 200
25 201
26 201
27 202
28 201
29 201
30 202
31 201
32 200
33 200
34 201
35 202
36 202
1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn10Quit
```

Рис. 2.33: 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$
```

Рис. 2.34: Повторное построение графика

Посмотрим заново построенный график (рис. 2.35):

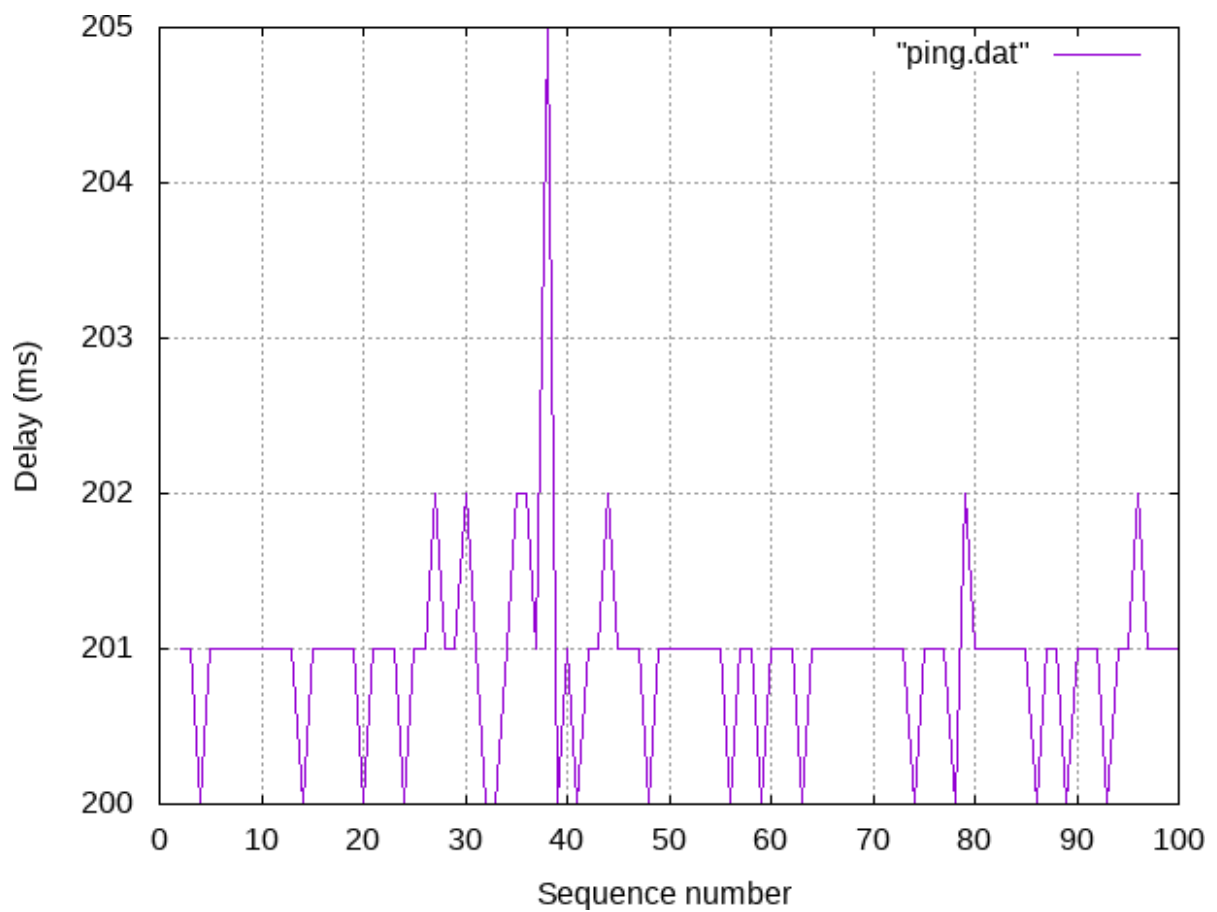
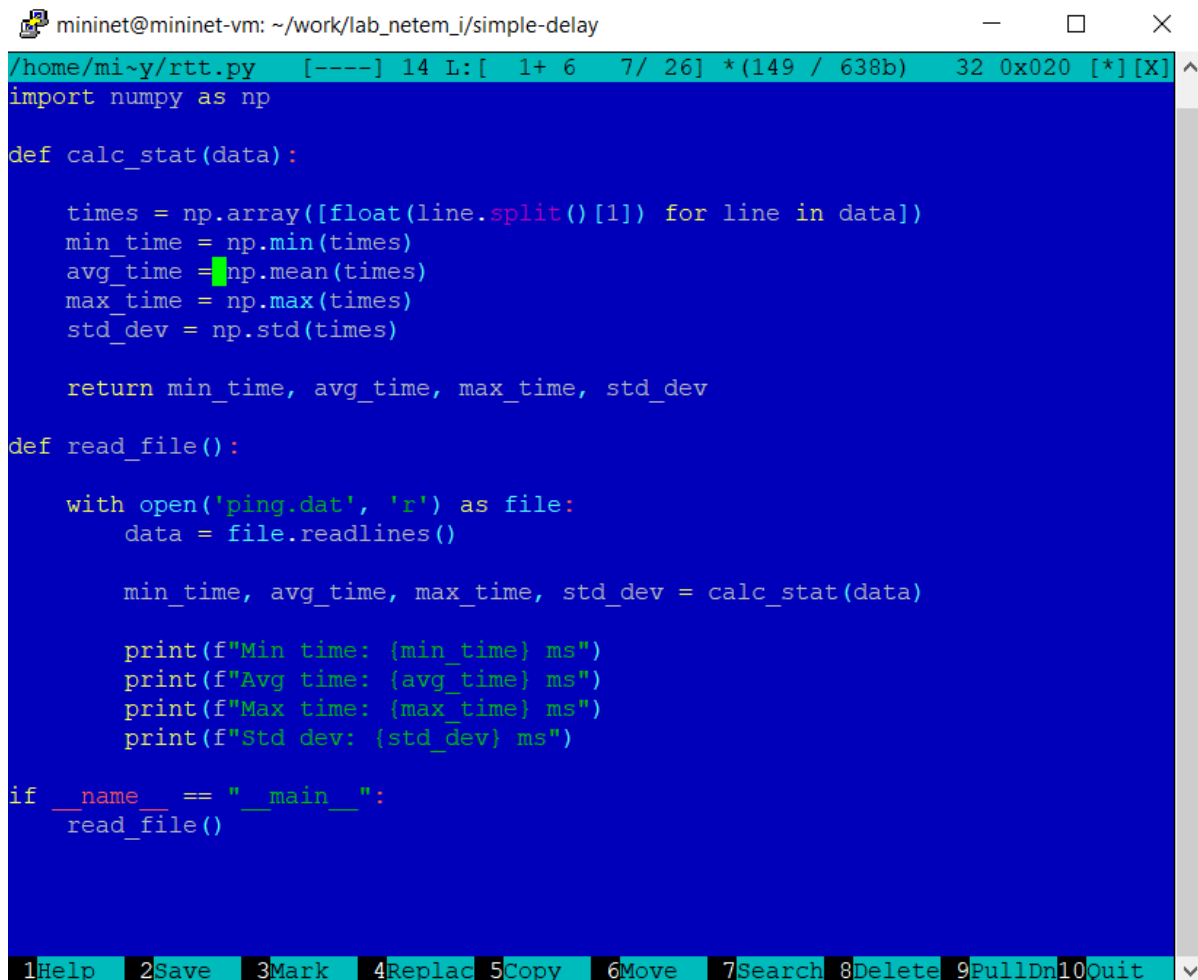


Рис. 2.35: Просмотр графика

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. 2.36 - рис. 2.38):



```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
/home/mininet-y/rtt.py [-----] 14 L: [ 1+ 6 7/ 26] *(149 / 638b) 32 0x020 [*][X]
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()

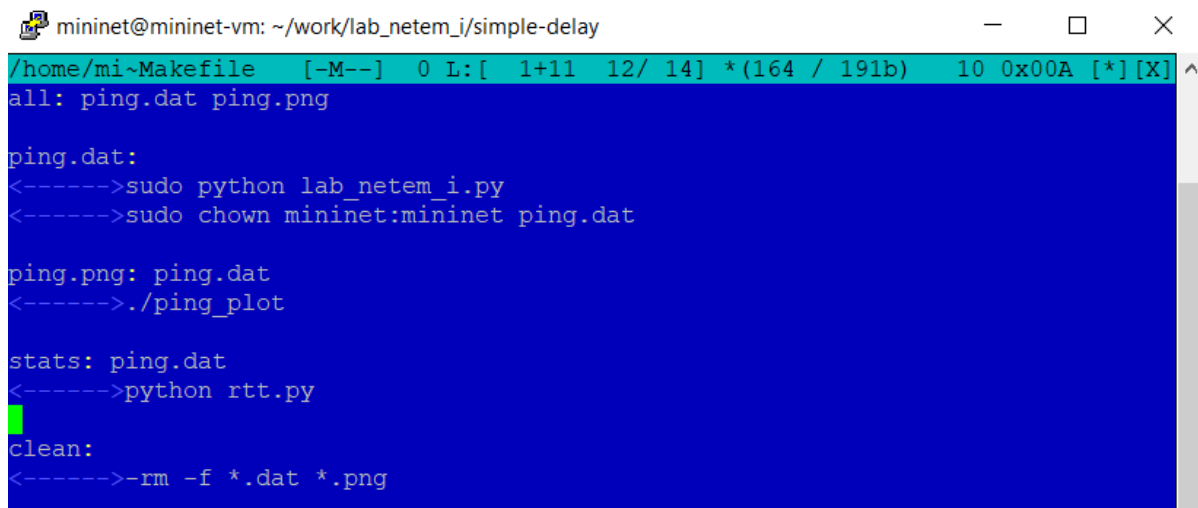
    min_time, avg_time, max_time, std_dev = calc_stat(data)

    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()
```

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

Рис. 2.36: Разработка скрипта для вычисления на основе данных файла ping.dat минимального, среднего, максимального и стандартного отклонения времени приёма-передачи



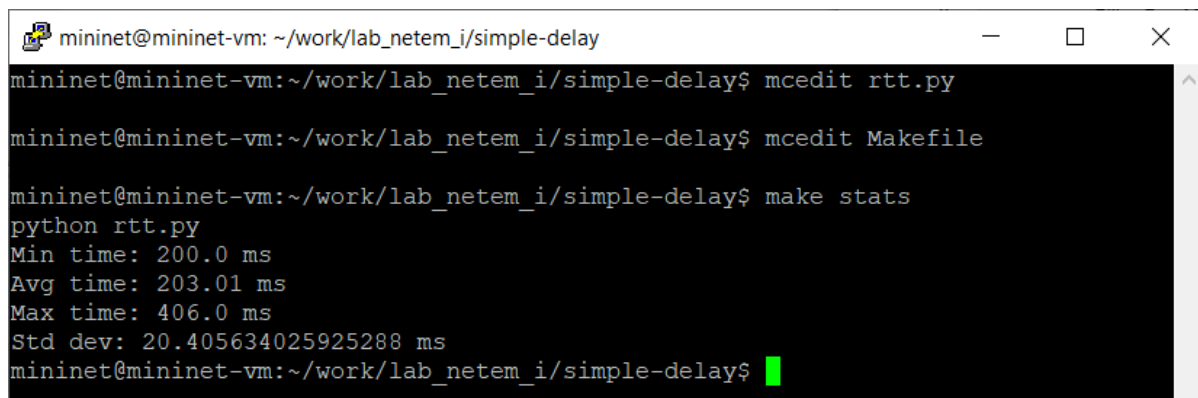
```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
/home/mi~Makefile  [-M--]  0 L:[ 1+11 12/ 14] *(164 / 191b) 10 0x00A [*][X]
all: ping.dat ping.png

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. 2.37: Добавление правила запуска скрипта в Makefil



```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ mcedit rtt.py
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ mcedit Makefile
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make stats
python rtt.py
Min time: 200.0 ms
Avg time: 203.01 ms
Max time: 406.0 ms
Std dev: 20.405634025925288 ms
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$
```

Рис. 2.38: 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. 2.39 - рис. 2.50):

```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
/home/mininet/.mininet/venv/bin/python [BM--] 67 L: [ 1+38 39/ 51] *(914 /1239b) 10 0x00A [*][X]
#!/usr/bin/env python

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

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

def emptyNet():
    "Create an empty network and add nodes to it."

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

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

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

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

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

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

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

    time.sleep(10) # Wait 10 seconds

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

    info( '*** Stopping network\n' )

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

Рис. 2.39: Воспроизводимый эксперимент по изменению задержки

```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ mcedit 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, $7}\'' |
sed -e \'/s/time=//g\' -e \'/s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make stats
python rtt.py
Min time: 100.0 ms
Avg time: 102.13 ms
Max time: 203.0 ms
Std dev: 10.15249230484811 ms
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$
```

Рис. 2.40: Воспроизводимый эксперимент по изменению задержки

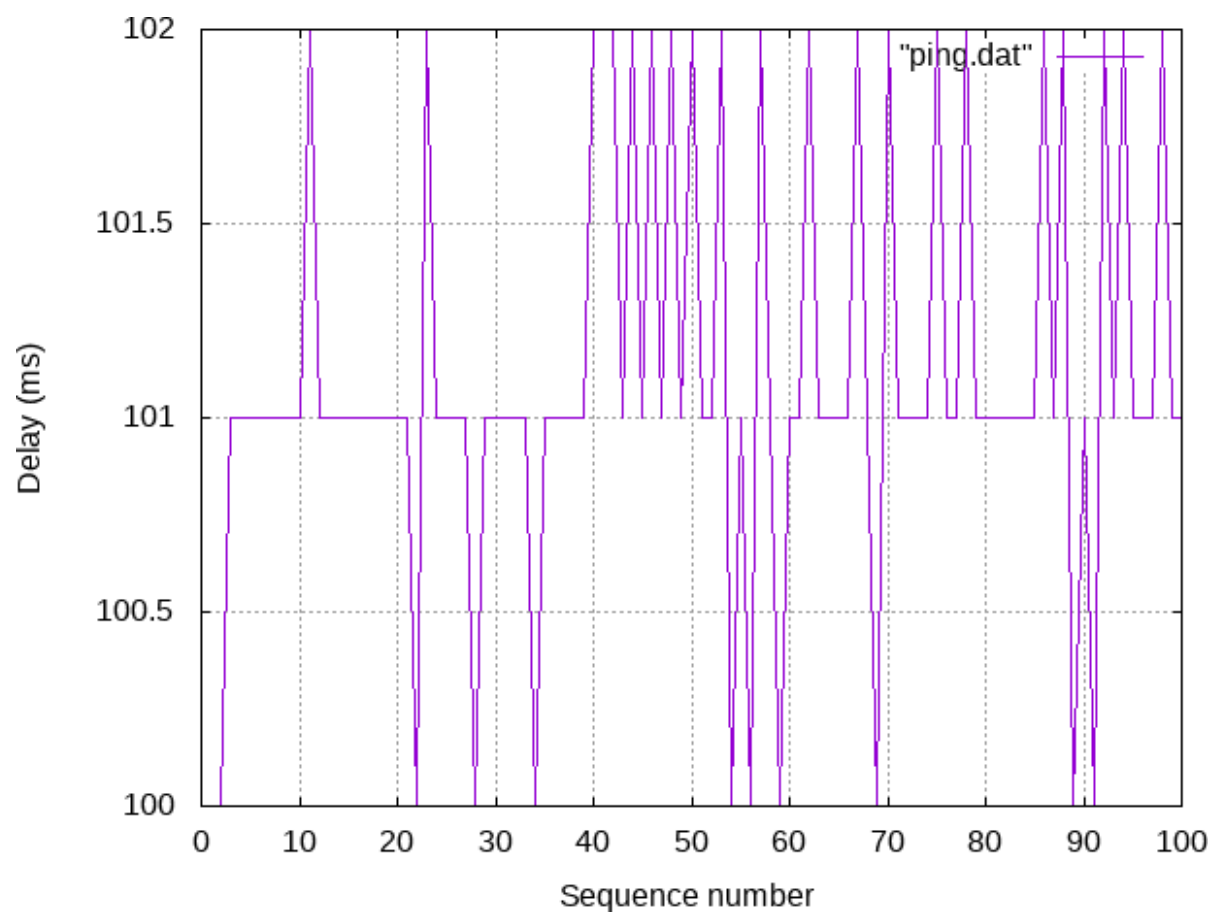


Рис. 2.41: Просмотр графика

```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
/home/mininet@mininet-vm:~/work/lab_netem_i/simple-delay [B---] 0 L:[ 1+37 38/ 51] *(779 /1246b) 32 0x020 [*][X] ^
#!/usr/bin/env python

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

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

def emptyNet():

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

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

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

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

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

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

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

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

    info( '*** Stopping network' )

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

Рис. 2.42: Воспроизводимый эксперимент по изменению джиттера

```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ mcedit 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, $7}\\' |
sed -e \'s/time=//g\' -e \'s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make stats
bash: make: command not found
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make stats
python rtt.py
Min time: 191.0 ms
Avg time: 202.09 ms
Max time: 404.0 ms
Std dev: 21.051885901267852 ms
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$
```

Рис. 2.43: Воспроизводимый эксперимент по изменению джиттера

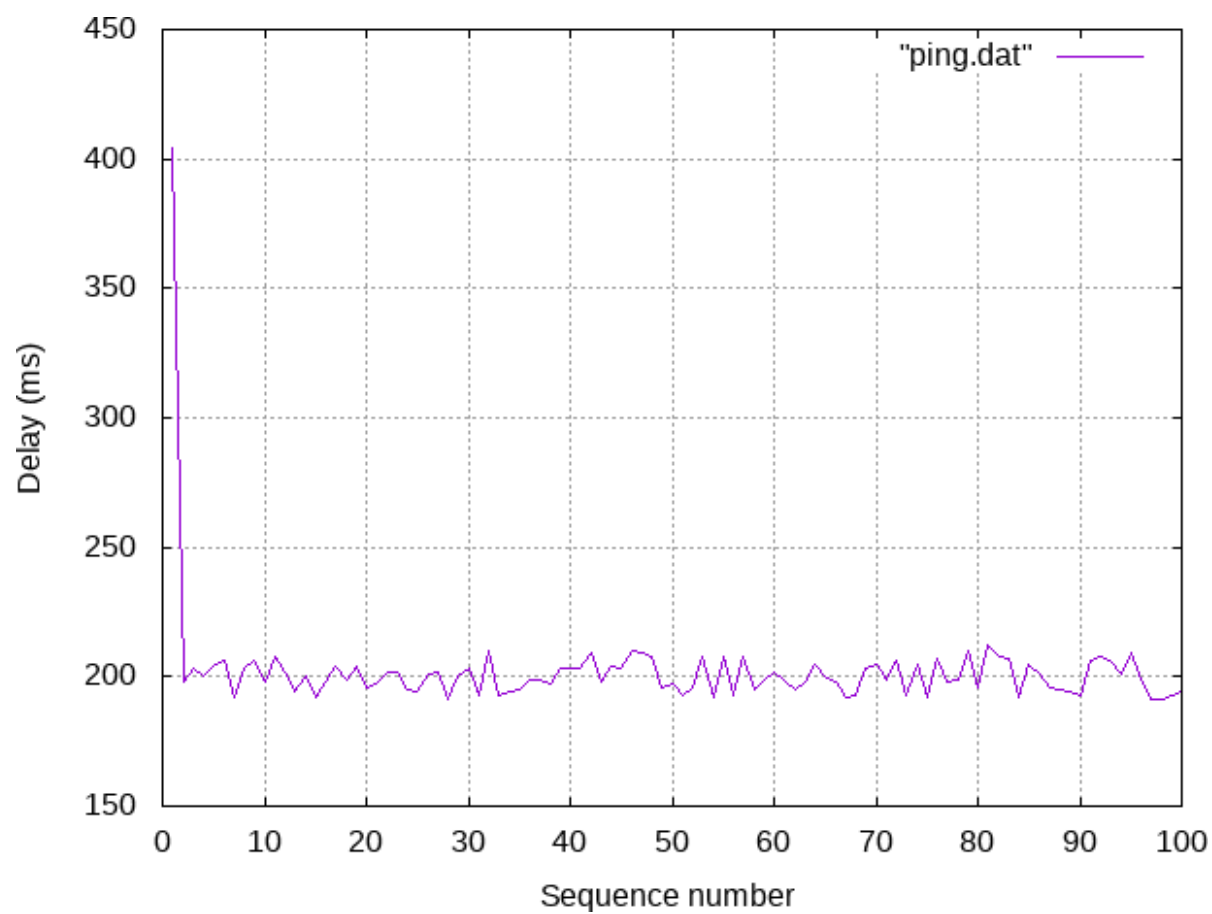
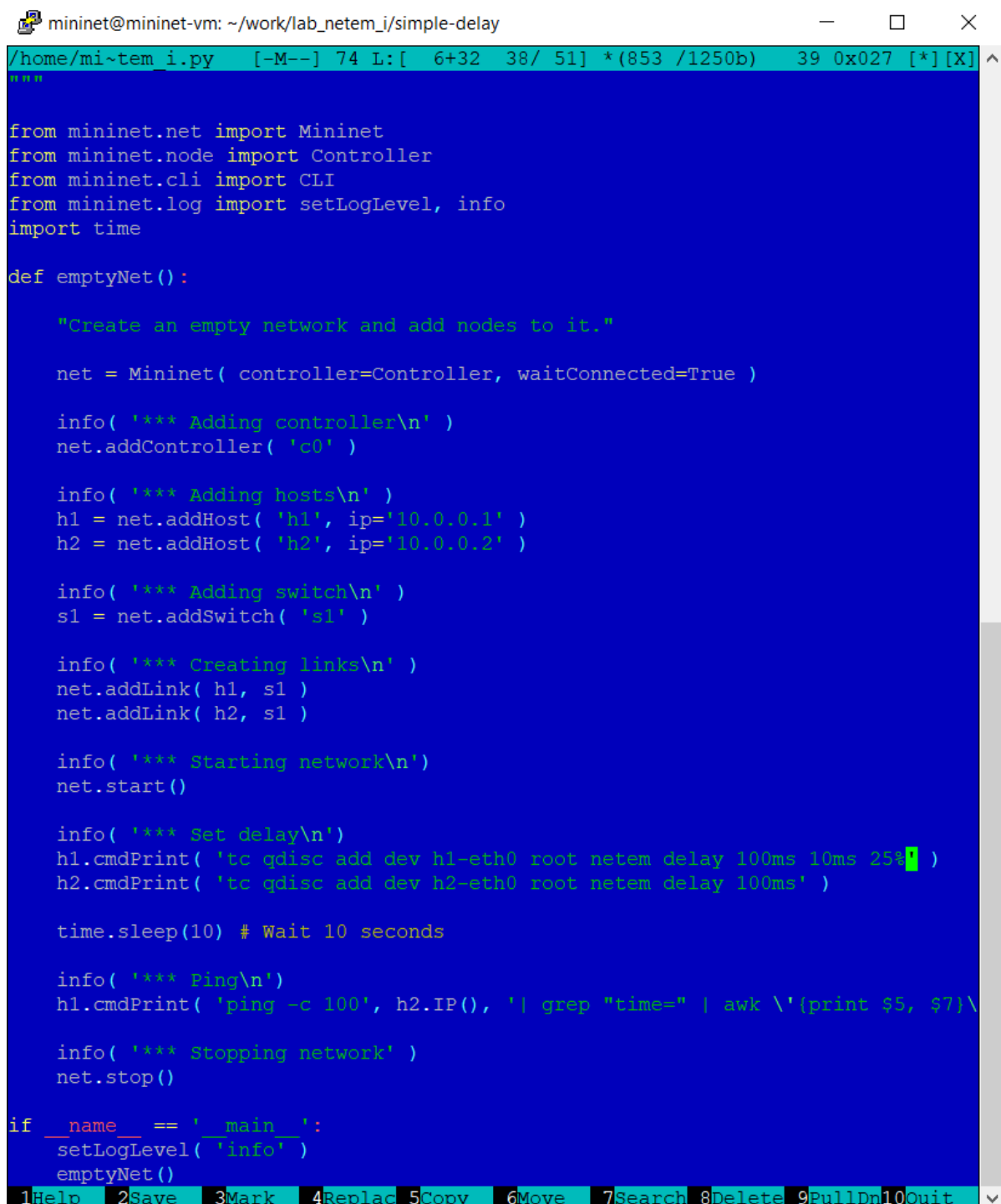


Рис. 2.44: Просмотр графика



```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
/home/mininet@mininet-vm:~/work/lab_netem_i/simple-delay
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' )
    h1.cmdPrint( 'ping -c 100', h2.IP(), '| grep "time=" | awk \'{print $5, $7}\'' )

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

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

Рис. 2.45: Воспроизводимый эксперимент по изменению значения корреляции для джиттера и задержки

```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
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, $7}\'' |
sed -e \'/s/time=//g\' -e \'/s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make stats
python rtt.py
Min time: 190.0 ms
Avg time: 202.43 ms
Max time: 408.0 ms
Std dev: 21.35989466266161 ms
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$
```

Рис. 2.46: Воспроизводимый эксперимент по изменению значения корреляции для джиттера и задержки

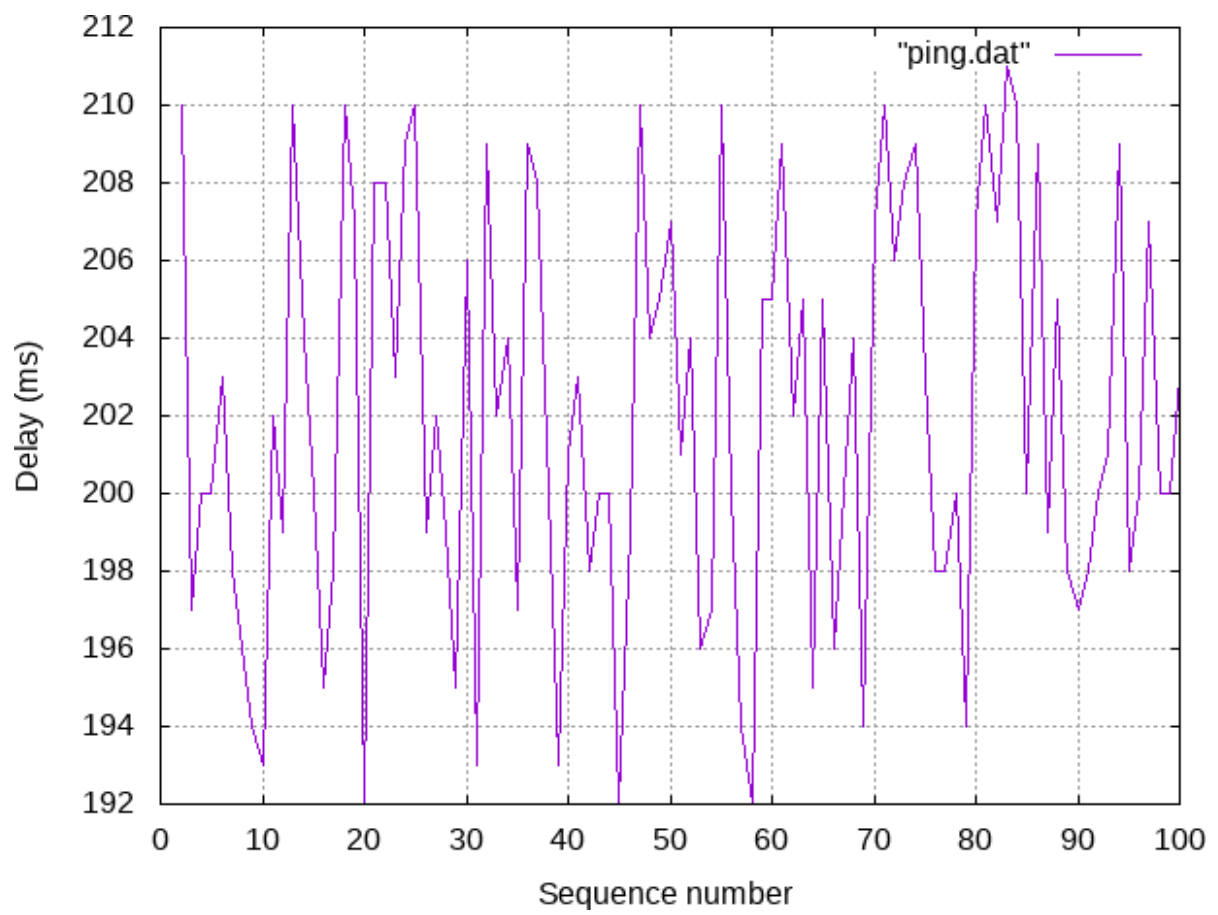


Рис. 2.47: Просмотр графика

```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
/home/mininet/work-ay/lab_netem_i.py [BM--] 68 L:[ 6+33 39/ 51] *(945 /1270b) 10 0x00A [*][X] ^
"""
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' )
    h1.cmdPrint( 'ping -c 100', h2.IP(), '| grep "time=" | awk \'{print $5, $7}\'} | sed -e \'/time

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

if __name__ == '__main__':
    setLogLevel( 'info' )
    emptyNet()
1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit
```

Рис. 2.48: Воспроизводимый эксперимент по изменению распределения времени задержки в эмулируемой глобальной сети

```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ mcedit 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, $7}\'' | sed -e \'/s/time=//g\' -e \'/s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make stats
python rtt.py
Min time: 180.0 ms
Avg time: 203.65 ms
Max time: 413.0 ms
Std dev: 23.55732370198279 ms
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$
```

Рис. 2.49: Воспроизводимый эксперимент по изменению распределения времени задержки в эмулируемой глобальной сети

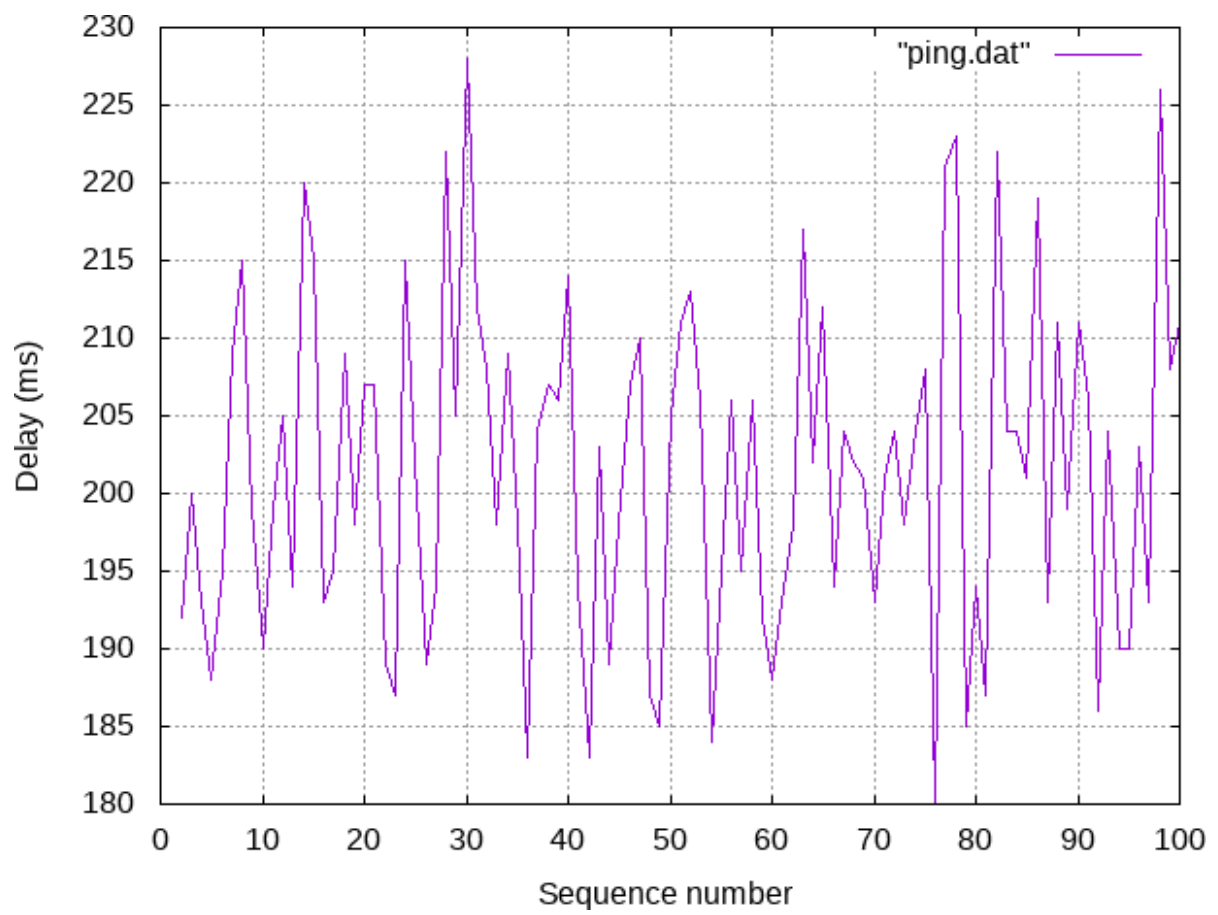


Рис. 2.50: Просмотр графика

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/>