

Отчёт по лабораторной работе №4

Дисциплина: Моделирование сетей передачи данных

Боровиков Даниил Александрович НПИбд-01-22

Содержание

1	Цель работы	6
2	Задание	7
3	Выполнение лабораторной работы	8
4	Выводы	41
	Список литературы	42

Список иллюстраций

3.1	Исправление прав запуска X-соединения в виртуальной машине mininet	8
3.2	Создание простейшей топологии	9
3.3	Отображение информации их сетевых интерфейсов и IP-адресов	10
3.4	Проверка подключения между хостами h1 и h2	11
3.5	Добавление задержки в 100 мс к выходному интерфейсу на хосте h1	11
3.6	Проверка	12
3.7	Добавление задержки в 100 мс к выходному интерфейсу на хосте h2	12
3.8	Проверка	13
3.9	Изменение задержки со 100 мс до 50 мс	13
3.10	Проверка	14
3.11	Восстановление конфигураций по умолчанию	14
3.12	Добавление на узле h1 задержки в 100 мс со случайным отклонением 10 мс	14
3.13	Проверка	15
3.14	Восстановление конфигурации интерфейса по умолчанию	15
3.15	Проверка	16
3.16	Восстановление конфигурации интерфейса по умолчанию	16
3.17	Настройка нормального распределения задержки на узле h1 в эмулируемой сети	16
3.18	Проверка	17
3.19	Восстановление конфигурации интерфейса по умолчанию	17
3.20	Завершение работы mininet в интерактивном режиме	18
3.21	Обновление репозитория программного обеспечения на виртуальной машине	18
3.22	Установка пакета gnet	19
3.23	Создание нового каталога	19
3.24	Создание каталога simple-delay	19
3.25	Создание скрипта lab_netem_i.py для эксперимента	20
3.26	Создание файла ping_plot	20
3.27	Создание скрипта ping_plot для визуализации результатов эксперимента	21
3.28	Настройка прав доступа к файлу скрипта	21
3.29	Создание файла Makefile	21
3.30	Добавления скрипта в Makefile для управления процессом проведения эксперимента	22
3.31	Выполнение эксперимента	23

3.32	Просмотр графика	24
3.33	Удаление первой строки из файла ping.dat	25
3.34	Повторное построение графика	25
3.35	Просмотр графика	26
3.36	Разработка скрипта для вычисления на основе данных файла ping.dat минимального, среднего, максимального и стандартного отклонения времени приёма-передачи	27
3.37	Добавление правила запуска скрипта в Makefil	27
3.38	Проверка	28
3.39	Воспроизводимый эксперимент по изменению задержки	29
3.40	Воспроизводимый эксперимент по изменению задержки	30
3.41	Просмотр графика	31
3.42	Воспроизводимый эксперимент по изменению джиттера	32
3.43	Воспроизводимый эксперимент по изменению джиттера	33
3.44	Просмотр графика	34
3.45	Воспроизводимый эксперимент по изменению значения корреляции для джиттера и задержки	35
3.46	Воспроизводимый эксперимент по изменению значения корреляции для джиттера и задержки	36
3.47	Просмотр графика	37
3.48	Воспроизводимый эксперимент по изменению распределения вре- мени задержки в эмулируемой глобальной сети	38
3.49	Воспроизводимый эксперимент по изменению распределения вре- мени задержки в эмулируемой глобальной сети	39
3.50	Просмотр графика	40

Список таблиц

1 Цель работы

Основной целью работы является знакомство с NETEM [1]. — 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 Задание

1. Задайте простейшую топологию, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8.
2. Проведите интерактивные эксперименты по добавлению/изменению задержки, джиттера, значения корреляции для джиттера и задержки, распределения времени задержки в эмулируемой глобальной сети.
3. Реализуйте воспроизводимый эксперимент по заданию значения задержки в эмулируемой глобальной сети. Постройте график.
4. Самостоятельно реализуйте воспроизводимые эксперименты по изменению задержки, джиттера, значения корреляции для джиттера и задержки, распределения времени задержки в эмулируемой глобальной сети. Постройте графики.

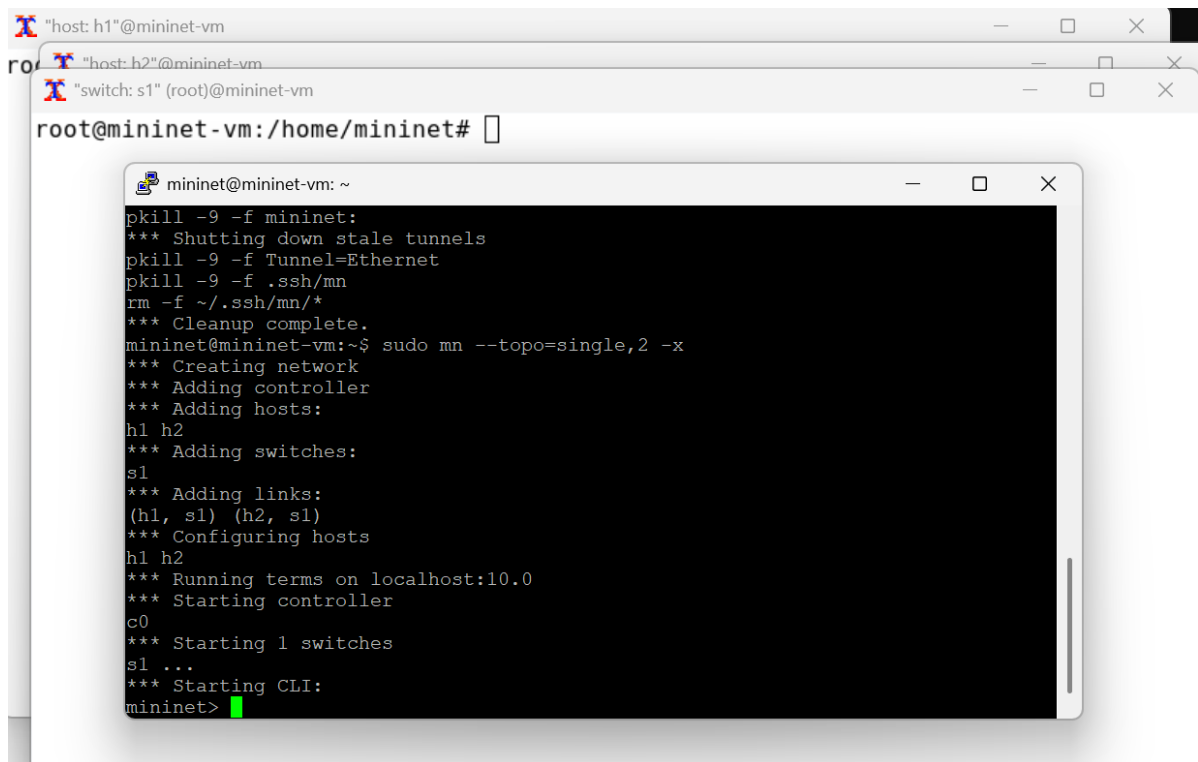
3 Выполнение лабораторной работы

В виртуальной машине mininet исправим права запуска X-соединения (рис. 3.1):

```
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  69fa6826576af937fcef92dc91d92ad
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth add mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  69fa6826576af937fc
efe92dc91d92ad
root@mininet-vm:~# xauth list $DISPLAY
mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  69fa6826576af937fcef92dc91d92ad
root@mininet-vm:~# logout
mininet@mininet-vm:~$
```

Рис. 3.1: 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. 3.2):



```
root@mininet-vm:/home/mininet#  
mininet@mininet-vm: ~  
pkill -9 -f mininet:  
*** Shutting down stale tunnels  
pkill -9 -f Tunnel=Ethernet  
pkill -9 -f .ssh/mn  
rm -f ~/.ssh/mn/*  
*** Cleanup complete.  
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. 3.2: Создание простейшей топологии

На хостах h1 и h2 введём команду `ifconfig`, чтобы отобразить информацию, относящуюся к их сетевым интерфейсам и назначенным им IP-адресам. В дальнейшем при работе с NETEM и командой `tc` будут использоваться интерфейсы h1-eth0 и h2-eth0 (рис. 3.3):

```
"host: h1"@mininet-vm
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 ca:e4:a0:5f:76:df 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 2004 bytes 337840 (337.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2004 bytes 337840 (337.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

"host: h2"@mininet-vm
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 66:c8:0c:4e:82:d3 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 1584 bytes 283212 (283.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1584 bytes 283212 (283.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet# █
```

Рис. 3.3: Отображение информации их сетевых интерфейсов и IP-адресов

Проверим подключение между хостами h1 и h2 с помощью команды ping с параметром -c 6 (рис. 3.4):

```
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=1.44 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.087 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.074 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=0.065 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.085 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5106ms
rtt min/avg/max/mdev = 0.036/0.298/1.443/0.512 ms
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=3.84 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.184 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.066 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.066 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.088 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.076 ms

--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5106ms
rtt min/avg/max/mdev = 0.066/0.720/3.842/1.396 ms
root@mininet-vm:/home/mininet#
```

Рис. 3.4: Проверка подключения между хостами h1 и h2

На хосте h1 добавим задержку в 100 мс к выходному интерфейсу (рис. 3.5):

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

Рис. 3.5: 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. 3.6):

```
"host: h1"@mininet-vm
6 packets transmitted, 6 received, 0% packet loss, time 5106ms
rtt min/avg/max/mdev = 0.036/0.298/1.443/0.512 ms
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms
root@mininet-vm:/home/mininet# ping -c 6 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=103 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 5012ms
rtt min/avg/max/mdev = 100.356/101.261/103.498/1.067 ms
root@mininet-vm:/home/mininet#
```

Рис. 3.6: Проверка

Для эмуляции глобальной сети с двунаправленной задержкой необходимо к соответствующему интерфейсу на хосте h2 также добавить задержку в 100 миллисекунд (рис. 3.7):

```
"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. 3.7: 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. 3.8):

```
"host: h1"@mininet-vm
--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5012ms
rtt min/avg/max/mdev = 100.356/101.261/103.498/1.067 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=202 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=203 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=202 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=203 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=202 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 5012ms
rtt min/avg/max/mdev = 200.830/202.079/202.631/0.607 ms
root@mininet-vm:/home/mininet#
```

Рис. 3.8: Проверка

Изменим задержку со 100 мс до 50 мс для отправителя h1 и для получателя h2 (рис. 3.9):

```
"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
00ms
root@mininet-vm:/home/mininet# sudo tc qdisc change dev h2-eth0 root netem delay 50ms
root@mininet-vm:/home/mininet#
```

Рис. 3.9: 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. 3.10):

```
"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=104 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=103 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 5010ms
rtt min/avg/max/mdev = 101.057/101.998/103.870/1.134 ms
root@mininet-vm:/home/mininet#
```

Рис. 3.10: Проверка

Восстановим конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса (рис. 3.11):

```
"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#
```

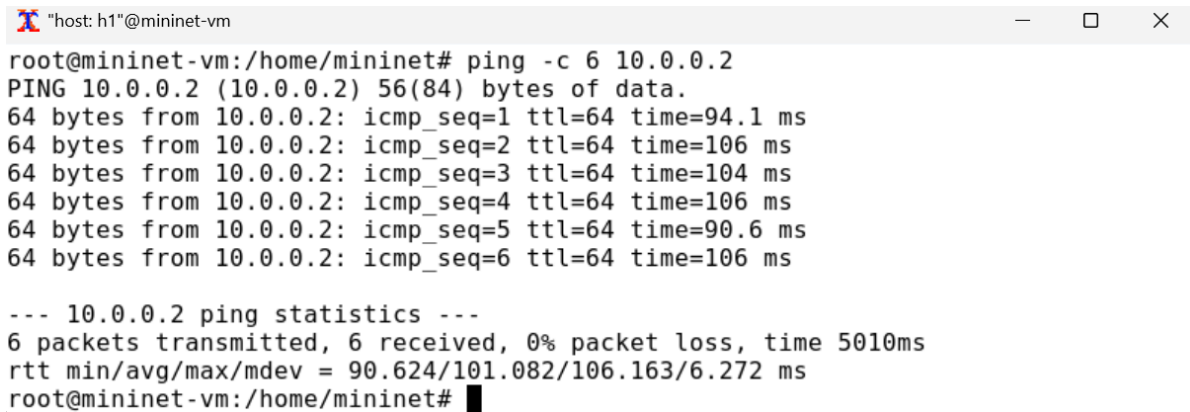
Рис. 3.11: Восстановление конфигураций по умолчанию

Добавим на узле h1 задержку в 100 мс со случайным отклонением 10 мс (рис. 3.12):

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

Рис. 3.12: 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. 3.13):

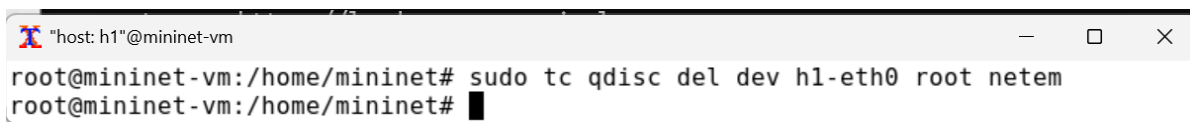
A terminal window titled "host: h1"@mininet-vm. The user runs the command "ping -c 6 10.0.0.2". The output shows six successful ping requests to 10.0.0.2, each with 64 bytes of data, TTL=64, and various response times between 90.6 ms and 106 ms. A summary line shows 6 packets transmitted, 6 received, 0% packet loss, and a total time of 5010ms. The round-trip time (rtt) statistics are also displayed: min/avg/max/mdev = 90.624/101.082/106.163/6.272 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=94.1 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=106 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=104 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=106 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=90.6 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=106 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5010ms
rtt min/avg/max/mdev = 90.624/101.082/106.163/6.272 ms
root@mininet-vm:/home/mininet#
```

Рис. 3.13: Проверка

Восстановим конфигурацию интерфейса по умолчанию на узле h1 (рис. 3.14):

A terminal window titled "host: h1"@mininet-vm. The user runs the command "sudo tc qdisc del dev h1-eth0 root netem". The prompt changes to root@mininet-vm, indicating the command was executed successfully.

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

Рис. 3.14: 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. 3.15):

```
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=100 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=95.2 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=92.0 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=107 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=105 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=92.9 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=106 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=103 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=110 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=110 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=93.9 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=98.3 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=98.6 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=111 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=100 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=104 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=107 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=103 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=108 ms

--- 10.0.0.2 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19040ms
rtt min/avg/max/mdev = 91.957/102.411/111.445/5.769 ms
root@mininet-vm:/home/mininet#
```

Рис. 3.15: Проверка

Восстановим конфигурацию интерфейса по умолчанию на узле h1 (рис. 3.16):

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

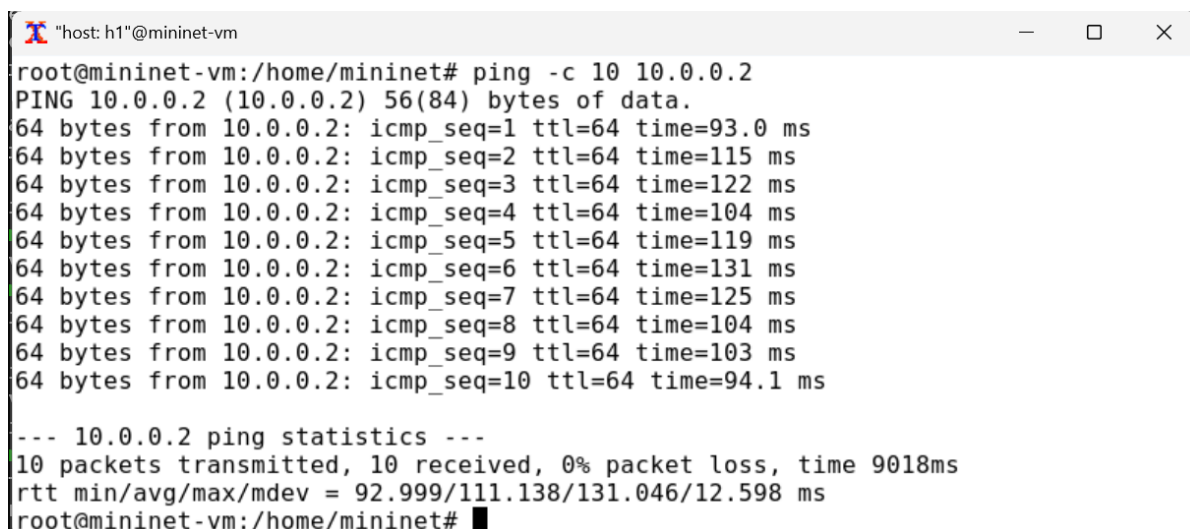
Рис. 3.16: Восстановление конфигурации интерфейса по умолчанию

Зададим нормальное распределение задержки на узле h1 в эмулируемой сети (рис. 3.17):

```
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. 3.17: 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. 3.18):

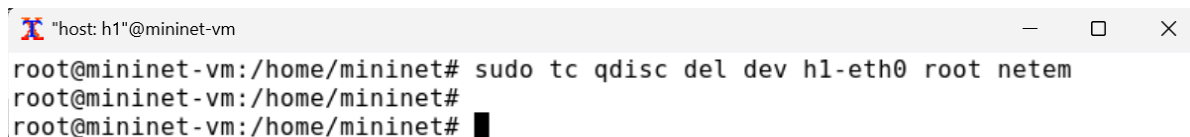


```
"host: h1"@mininet-vm
root@mininet-vm:/home/mininet# ping -c 10 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=93.0 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=115 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=122 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=104 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=119 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=131 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=125 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=104 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=103 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=94.1 ms

--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9018ms
rtt min/avg/max/mdev = 92.999/111.138/131.046/12.598 ms
root@mininet-vm:/home/mininet#
```

Рис. 3.18: Проверка

Восстановим конфигурацию интерфейса по умолчанию на узле h1 (рис. 3.19):



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

Рис. 3.19: Восстановление конфигурации интерфейса по умолчанию

Завершим работу mininet в интерактивном режиме (рис. 3.20):

```
mininet@mininet-vm: ~
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 1233.687 seconds
mininet@mininet-vm:~$
```

Рис. 3.20: Завершение работы mininet в интерактивном режиме

Обновим репозитории программного обеспечения на виртуальной машине (рис. 3.21):

```
mininet@mininet-vm: ~
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 1233.687 seconds
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://us.archive.ubuntu.com/ubuntu focal-backports InRelease [128 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [3,955 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [1,114 kB]
Fetched 5,452 kB in 4s (1,354 kB/s)
Reading package lists... Done
mininet@mininet-vm:~$
```

Рис. 3.21: Обновление репозитория программного обеспечения на виртуальной машине

Установим пакет geee для просмотра файлов png (рис. 3.22):

```
mininet@mininet-vm: ~  
invoke-rc.d: initscript gdm3, action "reload" failed.  
Setting up libclutter-gst-3.0-0:amd64 (3.0.27-1) ...  
Setting up ubuntu-docs (20.04.3) ...  
Setting up network-manager-gnome (1.8.24-1ubuntu3) ...  
Setting up gnome-user-docs (3.36.2+git20200704-0ubuntu0.1) ...  
Setting up gstreamer1.0-clutter-3.0:amd64 (3.0.27-1) ...  
Setting up libcheese8:amd64 (3.34.0-1ubuntu1) ...  
Setting up libcheese-gtk25:amd64 (3.34.0-1ubuntu1) ...  
Setting up aptdaemon (1.1.1+bzr982-0ubuntu32.3) ...  
Setting up python3-aptdaemon (1.1.1+bzr982-0ubuntu32.3) ...  
Setting up python3-aptdaemon.gtk3widgets (1.1.1+bzr982-0ubuntu32.3) ...  
Setting up language-selector-gnome (0.204.2) ...  
Setting up gnome-control-center (1:3.36.5-0ubuntu4.1) ...  
Processing triggers for mime-support (3.64ubuntu1) ...  
Processing triggers for hicolor-icon-theme (0.17-2) ...  
Processing triggers for libgtk-3-0:amd64 (3.24.20-0ubuntu1) ...  
Processing triggers for libc-bin (2.31-0ubuntu9) ...  
Processing triggers for systemd (245.4-4ubuntu3.4) ...  
Processing triggers for man-db (2.9.1-1) ...  
Processing triggers for shared-mime-info (1.15-1) ...  
Processing triggers for udev (245.4-4ubuntu3.4) ...  
Processing triggers for fontconfig (2.13.1-2ubuntu3) ...  
Processing triggers for sgml-base (1.29.1) ...  
Setting up sgml-data (2.0.11) ...  
Processing triggers for sgml-base (1.29.1) ...  
Setting up docbook-xml (4.5-9) ...  
Processing triggers for dbus (1.12.16-2ubuntu2.3) ...  
Processing triggers for rygel (0.38.3-1ubuntu1) ...  
Processing triggers for sgml-base (1.29.1) ...  
mininet@mininet-vm:~$
```

Рис. 3.22: Установка пакета geeqie

Для каждого воспроизводимого эксперимента expname создадим свой каталог, в котором будут размещаться файлы эксперимента (рис. 3.23):

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

Рис. 3.23: Создание нового каталога

В виртуальной среде mininet в своём рабочем каталоге с проектами создадим каталог simple-delay и перейдём в него (рис. 3.24):

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

Рис. 3.24: Создание каталога simple-delay

Создадим скрипт для эксперимента lab_netem_i.py (рис. 3.25):

```
mininet@mininet-vm: ~/work  GNU nano 4.8  lab_netem_i.py
#!/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' )
    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/time=//g\' -e \'/icmp_seq=//g\' > ping.dat' )

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

Рис. 3.25: Создание скрипта lab_netem_i.py для эксперимента

Создадим файл ping_plot (рис. 3.26):

```
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$ nano ping_plot |
```

Рис. 3.26: Создание файла ping_plot

Затем создадим скрипт для визуализации ping_plot результатов эксперимента (рис. 3.27):

```
GNU nano 4.8                                ping_plot
#!/usr/bin/gnuplot --persist

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

Рис. 3.27: Создание скрипта ping_plot для визуализации результатов эксперимента

Зададим права доступа к файлу скрипта (рис. 3.28):

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

Рис. 3.28: Настройка прав доступа к файлу скрипта

Создадим файла Makefile (рис. 3.29):

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

Рис. 3.29: Создание файла Makefile

Внутри файла Makefile поместим скрипт для управления процессом проведения эксперимента (рис. 3.30):

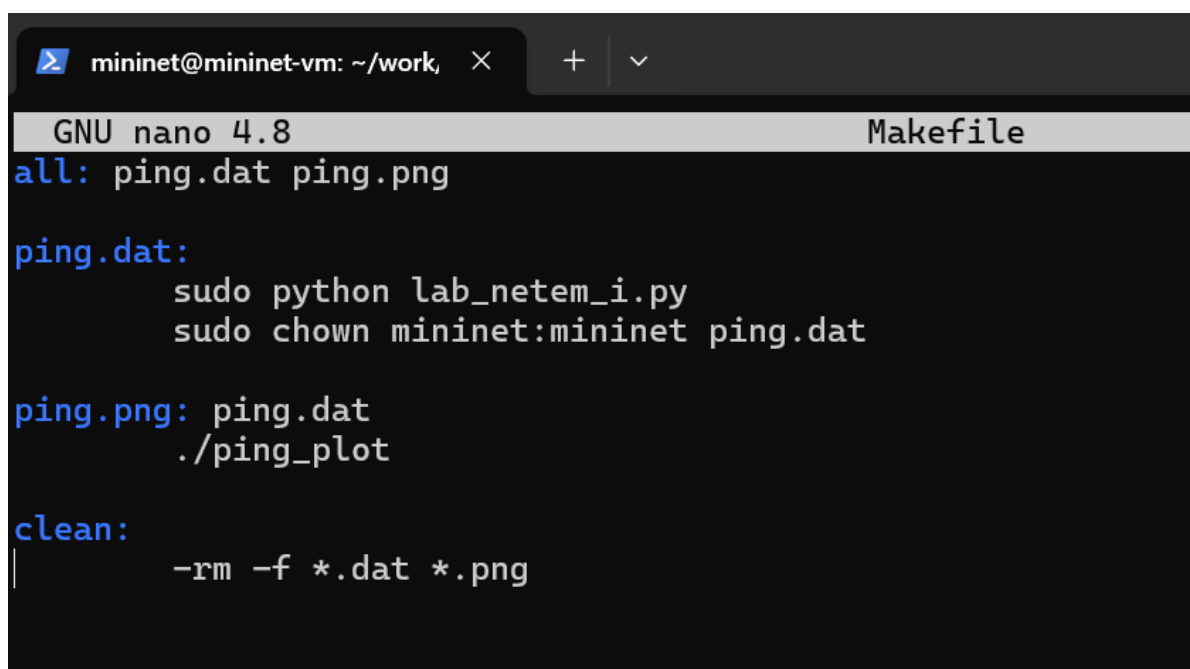
```

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. 3.30: Добавления скрипта в Makefile для управления процессом проведения эксперимента

Выполним эксперимент (рис. 3.31):

A screenshot of a terminal window with a dark background. The window title bar shows 'mininet@mininet-vm: ~/work' with a terminal icon, a close button, and window control buttons. The terminal content shows the GNU nano 4.8 editor editing a file named 'Makefile'. The Makefile contains 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. 3.31: Выполнение эксперимента

Посмотрим построенный в результате выполнения скриптов график (рис. 3.32):

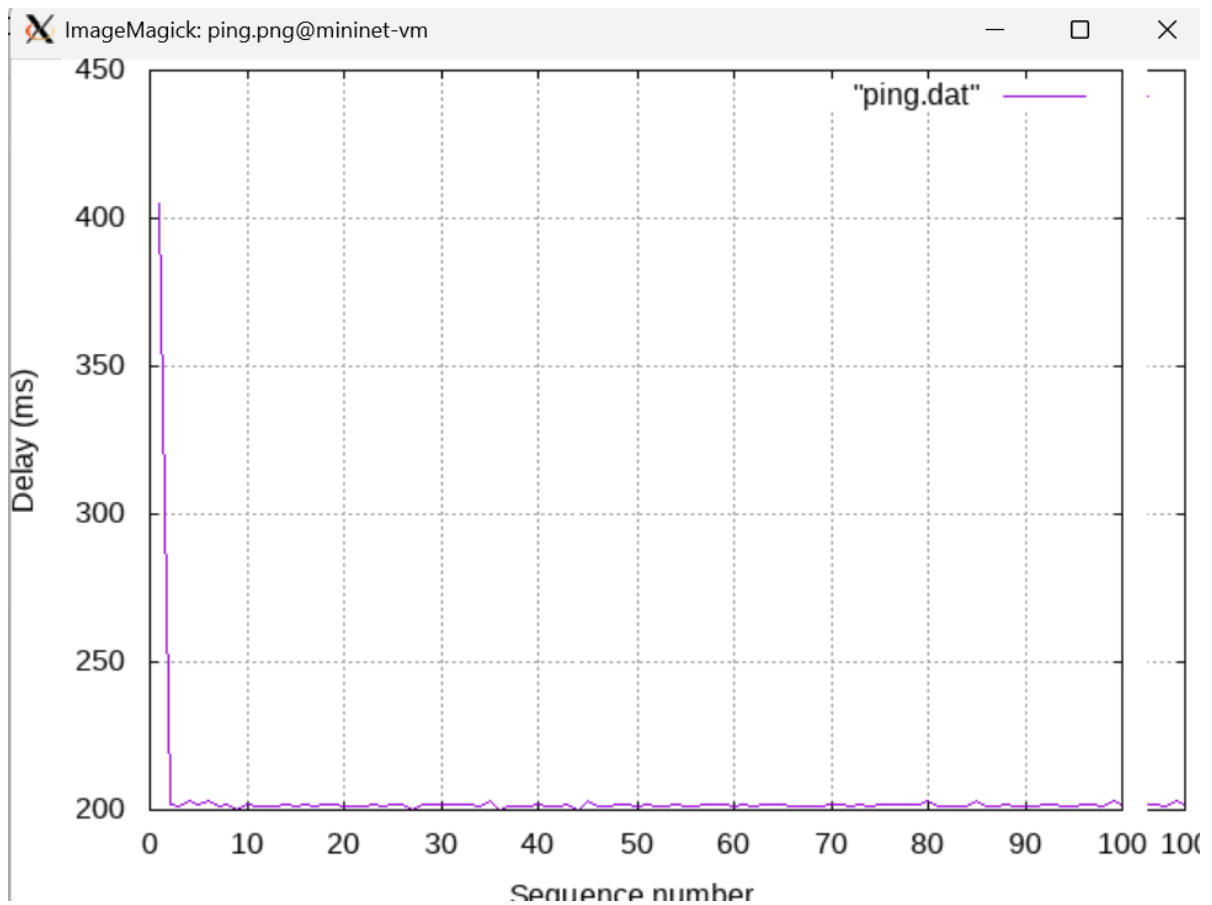


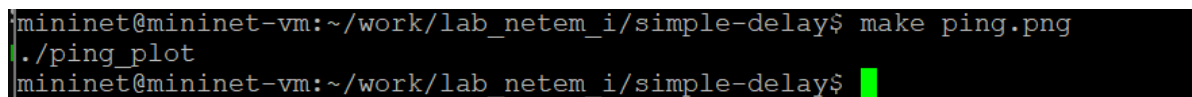
Рис. 3.32: Просмотр графика

Из файла ping.dat удалим первую строку и заново построим график (рис. 3.33 - рис. 3.34):



```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
GNU nano 4.8 ping.dat Modified
2 202
3 201
4 203
5 202
6 203
7 201
8 202
9 200
10 202
11 201
12 201
13 201
14 202
15 201
16 202
17 201
18 202
19 202
20 201
21 201
22 201
23 202
24 201
25 202
```

Рис. 3.33: Удаление первой строки 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. 3.34: Повторное построение графика

Посмотрим заново построенный график (рис. 3.35):

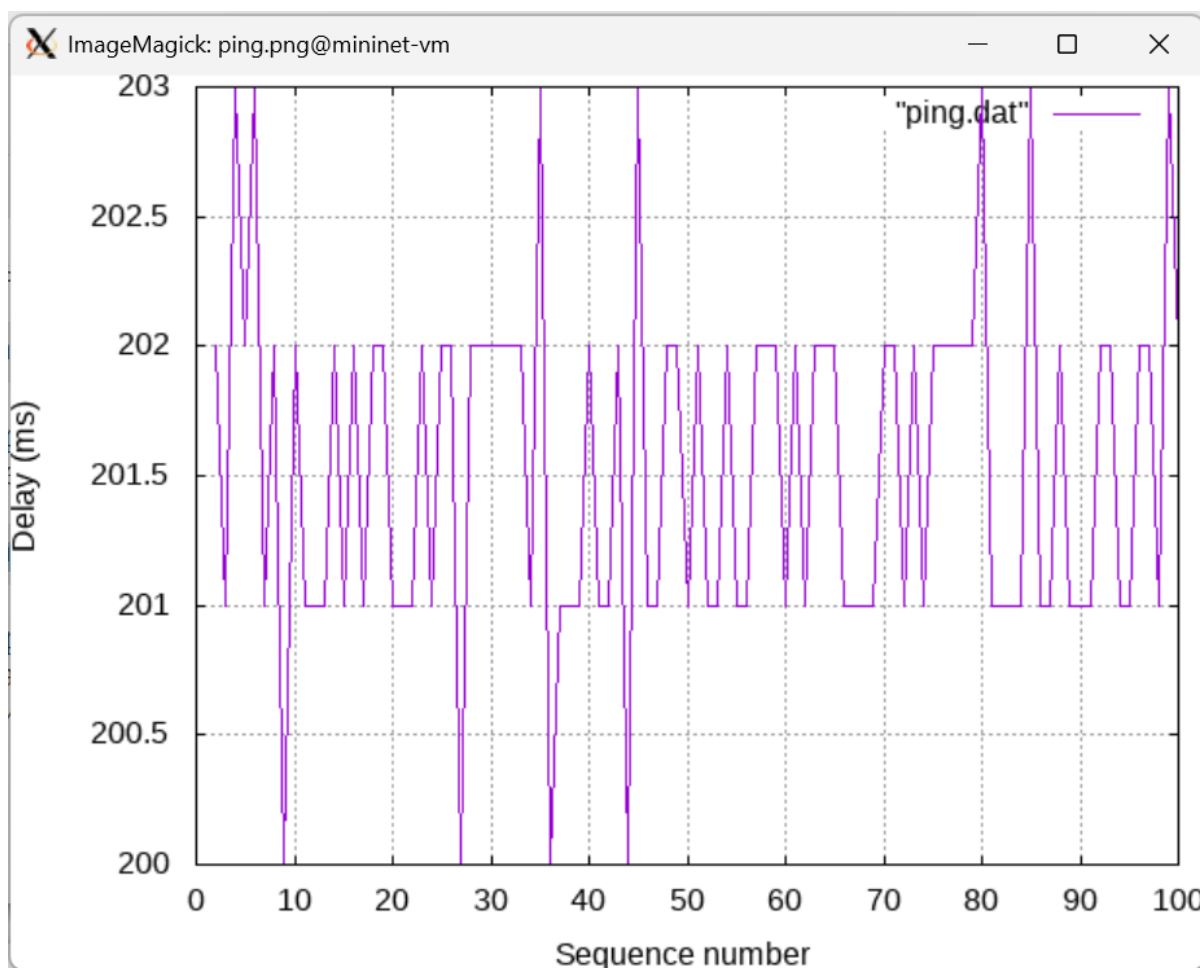
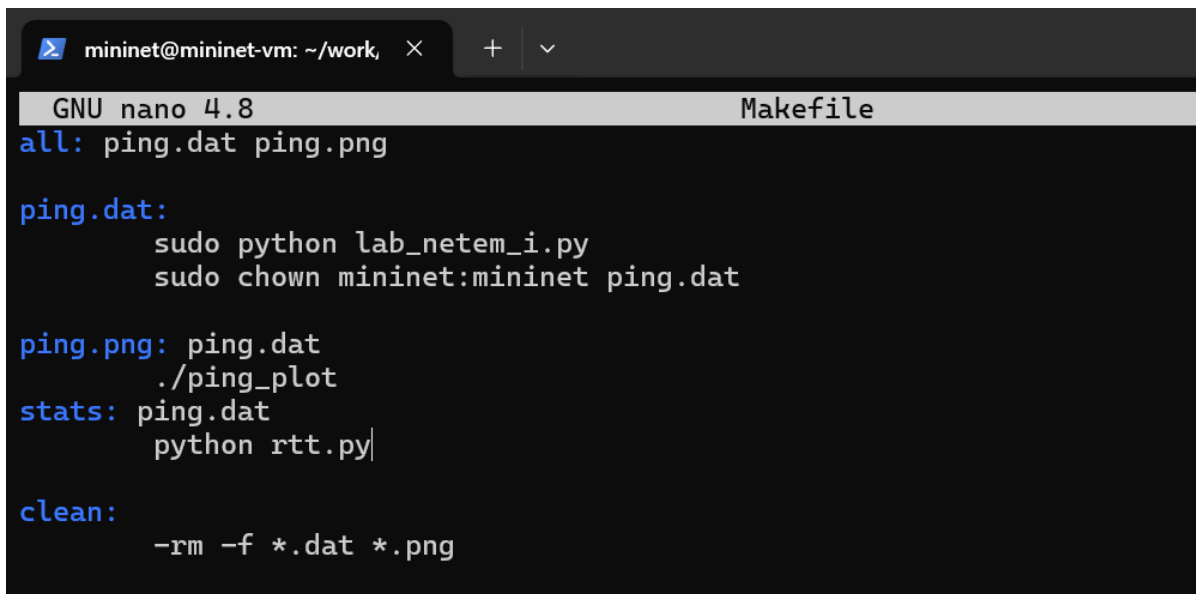


Рис. 3.35: Просмотр графика

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. 3.36 - рис. 3.38):



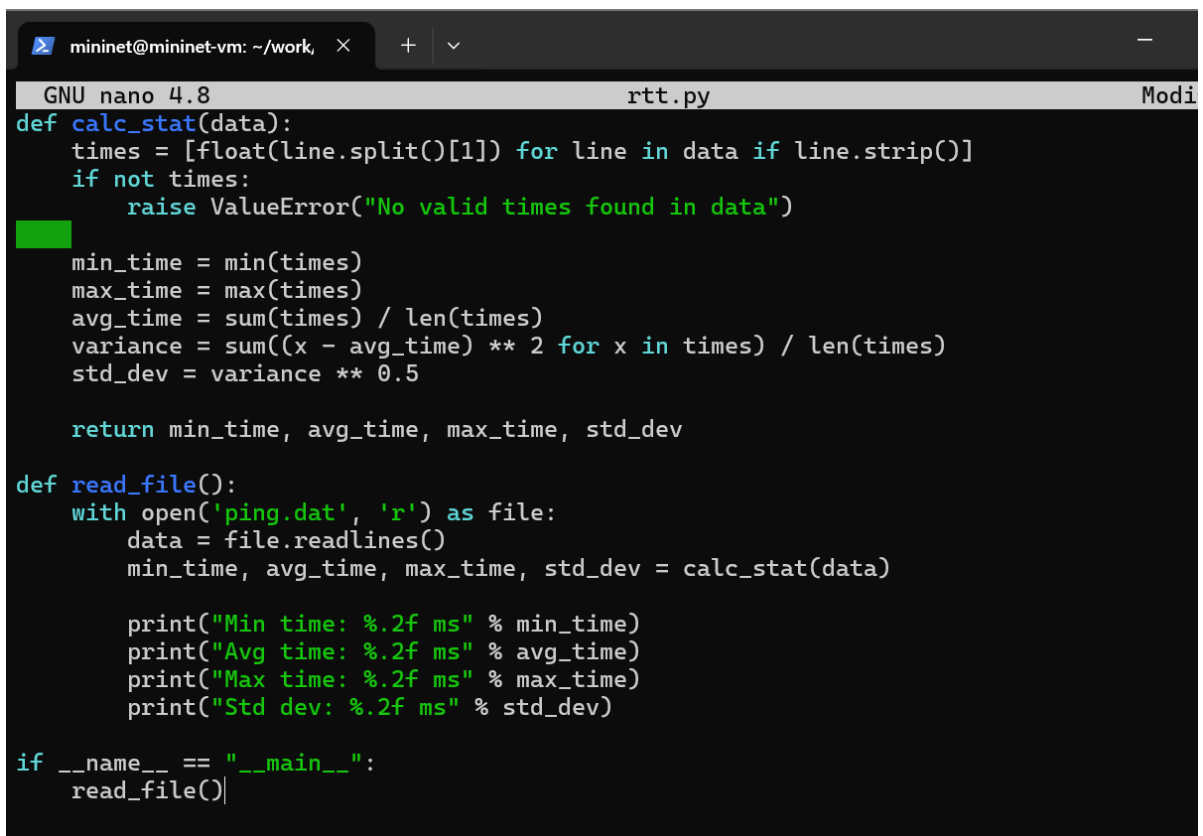
```
mininet@mininet-vm: ~/work, x + v
GNU nano 4.8 Makefile
all: ping.dat ping.png

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

ping.png: ping.dat
    ./ping_plot
stats: ping.dat
    python rtt.py

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

Рис. 3.36: Разработка скрипта для вычисления на основе данных файла ping.dat минимального, среднего, максимального и стандартного отклонения времени приёма-передачи



```
mininet@mininet-vm: ~/work, x + v
GNU nano 4.8 rtt.py Modi
def calc_stat(data):
    times = [float(line.split()[1]) for line in data if line.strip()]
    if not times:
        raise ValueError("No valid times found in data")

    min_time = min(times)
    max_time = max(times)
    avg_time = sum(times) / len(times)
    variance = sum((x - avg_time) ** 2 for x in times) / len(times)
    std_dev = variance ** 0.5

    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("Min time: %.2f ms" % min_time)
        print("Avg time: %.2f ms" % avg_time)
        print("Max time: %.2f ms" % max_time)
        print("Std dev: %.2f ms" % std_dev)

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

Рис. 3.37: Добавление правила запуска скрипта в Makefile

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make stats
python rtt.py
Min time: 200.00 ms
Avg time: 201.55 ms
Max time: 203.00 ms
Std dev: 0.69 ms
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ |
```

Рис. 3.38: 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. 3.39 - рис. 3.50):

```
mininet@mininet-vm: ~/work, × + ▾
GNU nano 4.8 lab_netem_i.py Modified
#!/usr/bin/env python

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}\'' | se

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

Рис. 3.39: Воспроизводимый эксперимент по изменению задержки

```

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: 100.00 ms
Avg time: 102.42 ms
Max time: 208.00 ms
Std dev: 10.63 ms
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ |

```

25 октября 2025 г.

Рис. 3.40: Воспроизводимый эксперимент по изменению задержки

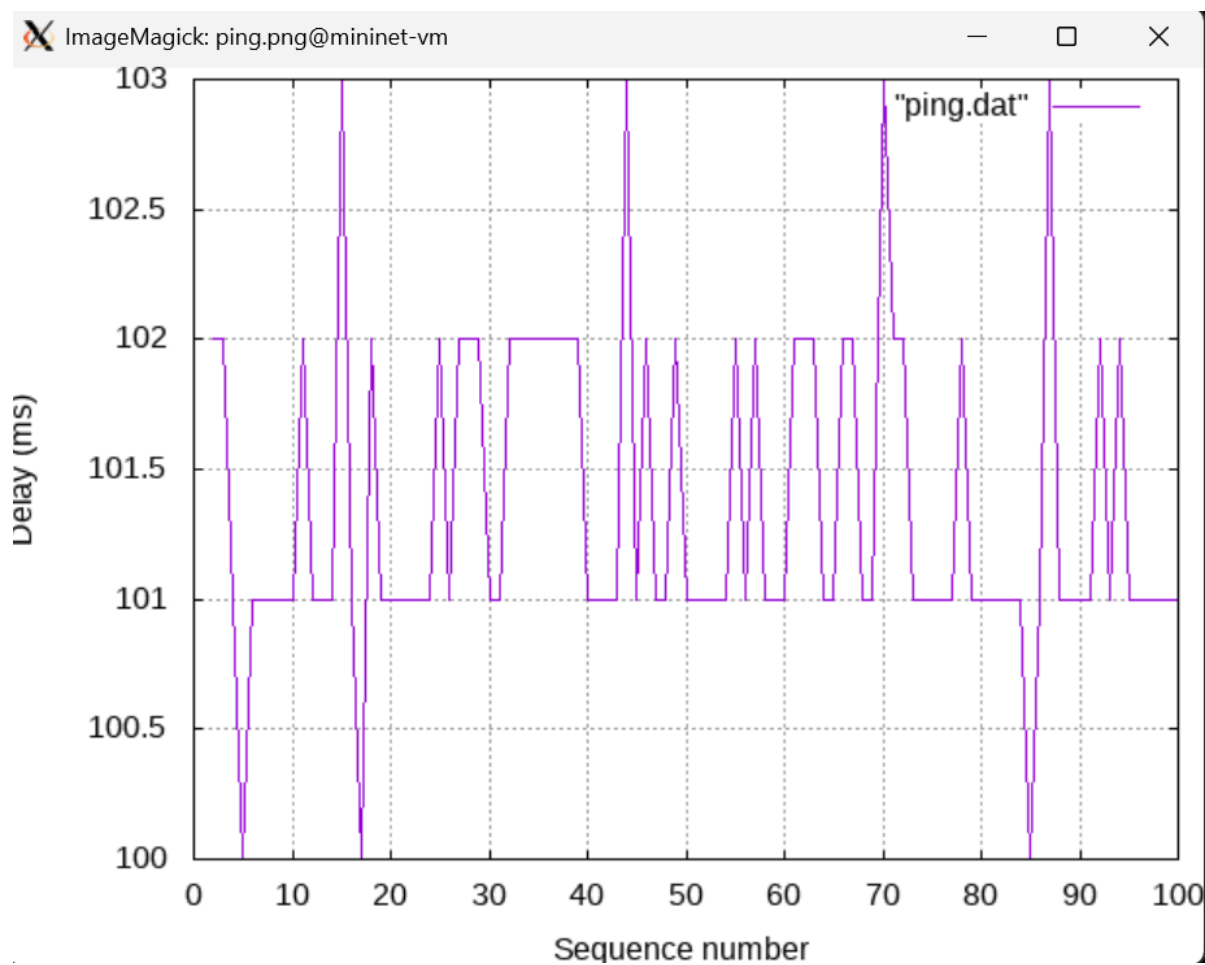


Рис. 3.41: Просмотр графика

```
mininet@mininet-vm: ~/work, × + ▾
GNU nano 4.8 lab_netem_i.py Modified
#!/usr/bin/env python

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}\'} | se

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

Рис. 3.42: Воспроизводимый эксперимент по изменению джиттера


```

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
python rtt.py
Min time: 192.00 ms
Avg time: 203.69 ms
Max time: 407.00 ms
Std dev: 21.23 ms
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$

```

25 октября 2025 г.

Сб 18:57 (Местное время)

Рис. 3.43: Воспроизводимый эксперимент по изменению джиттера

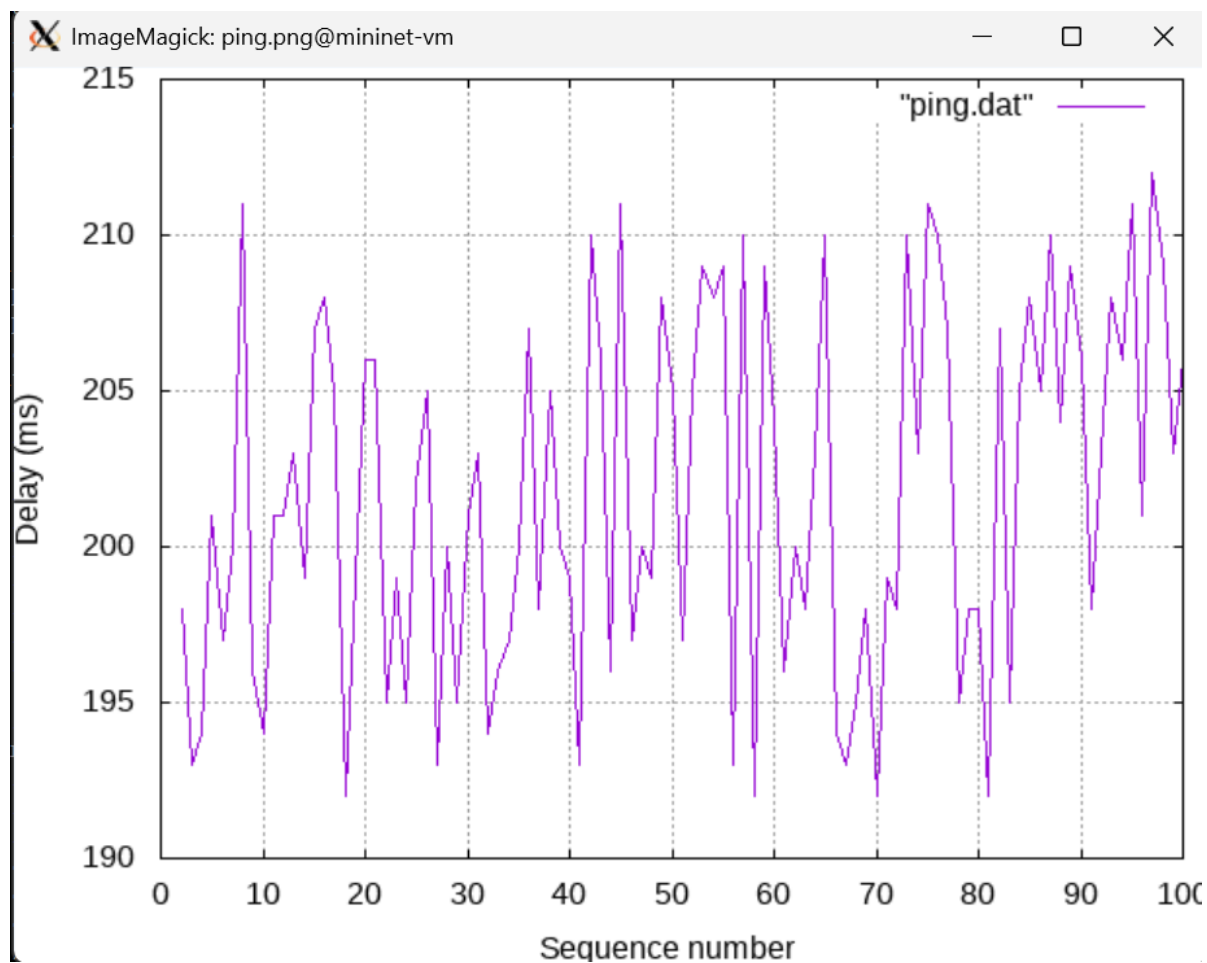


Рис. 3.44: Просмотр графика

```
mininet@mininet-vm: ~/work, × + ▾
GNU nano 4.8 lab_netem_i.py Modified
#!/usr/bin/env python

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}\'' | se>

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

Рис. 3.45: Воспроизводимый эксперимент по изменению значения корреляции для джиттера и задержки

```

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

mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make stats
python rtt.py
Traceback (most recent call last):
  File "rtt.py", line 25, in <module>
    read_file()
  File "rtt.py", line 17, in read_file
    min_time, avg_time, max_time, std_dev = calc_stat(data)
  File "rtt.py", line 2, in calc_stat
    times = [float(line.split()[1]) for line in data if line.strip()]
  File "rtt.py", line 2, in <listcomp>
    times = [float(line.split()[1]) for line in data if line.strip()]
IndexError: list index out of range
make: *** [Makefile:10: stats] Error 1
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nano ping.dat

Use "fg" to return to nano.

[1]+  Stopped                  nano ping.dat
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nano ping.dat
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nano ping.dat
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make stats
python rtt.py
Min time: 192.00 ms
Avg time: 201.54 ms
Max time: 212.00 ms
Std dev: 5.64 ms

```

Состояние батареи: осталось 95% заряда

Рис. 3.46: Воспроизводимый эксперимент по изменению значения корреляции для джиттера и задержки

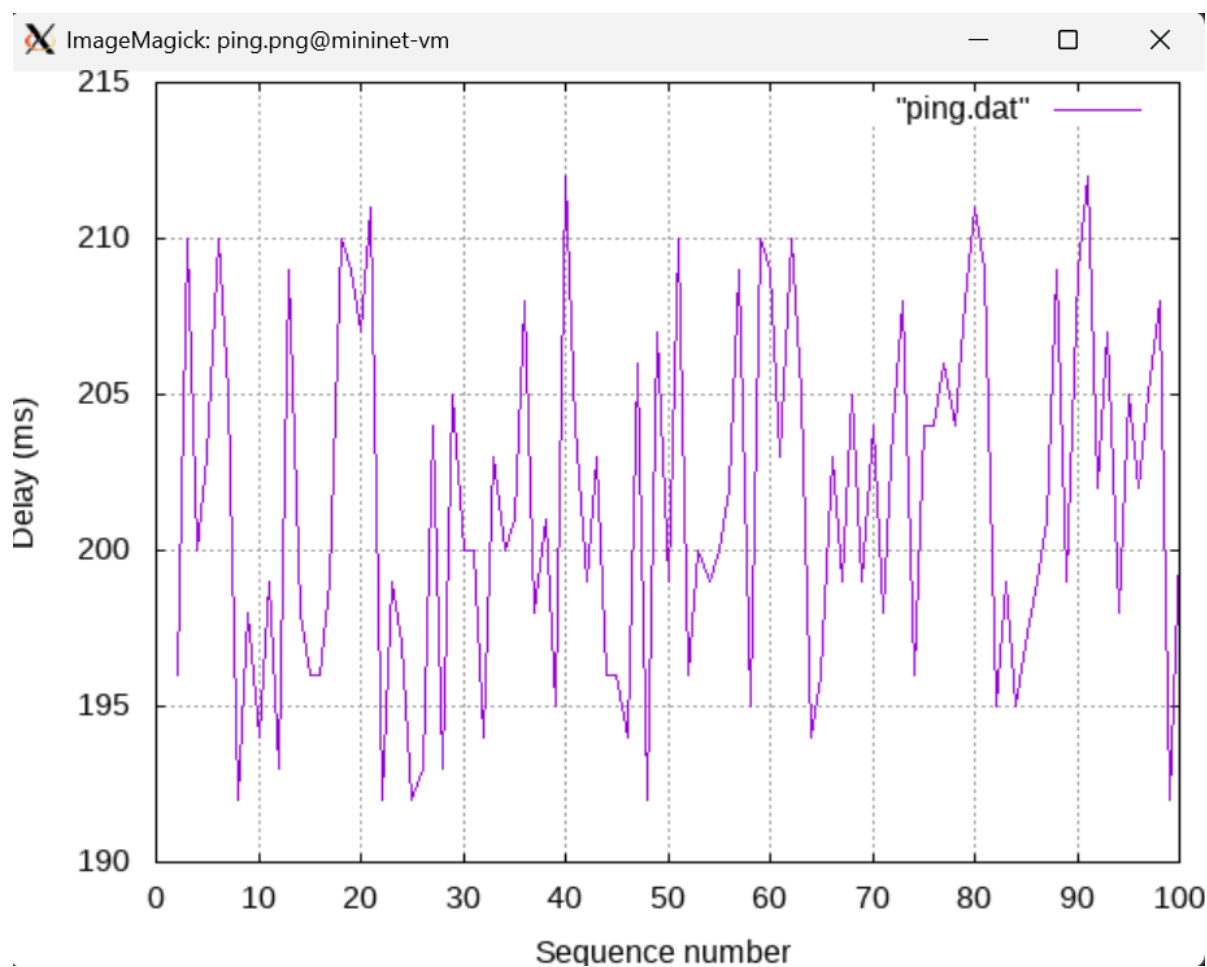


Рис. 3.47: Просмотр графика

```
#!/usr/bin/env python

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
```

Рис. 3.48: Воспроизводимый эксперимент по изменению распределения времени задержки в эмулируемой глобальной сети

```

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, $7}\'' | sed -e \'s/time=//g\' -e \'
s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nano ping.dat
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make stats
python rtt.py
Min time: 178.00 ms
Avg time: 201.71 ms
Max time: 225.00 ms
Std dev: 10.81 ms
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ |

```

Рис. 3.49: Воспроизводимый эксперимент по изменению распределения времени задержки в эмулируемой глобальной сети

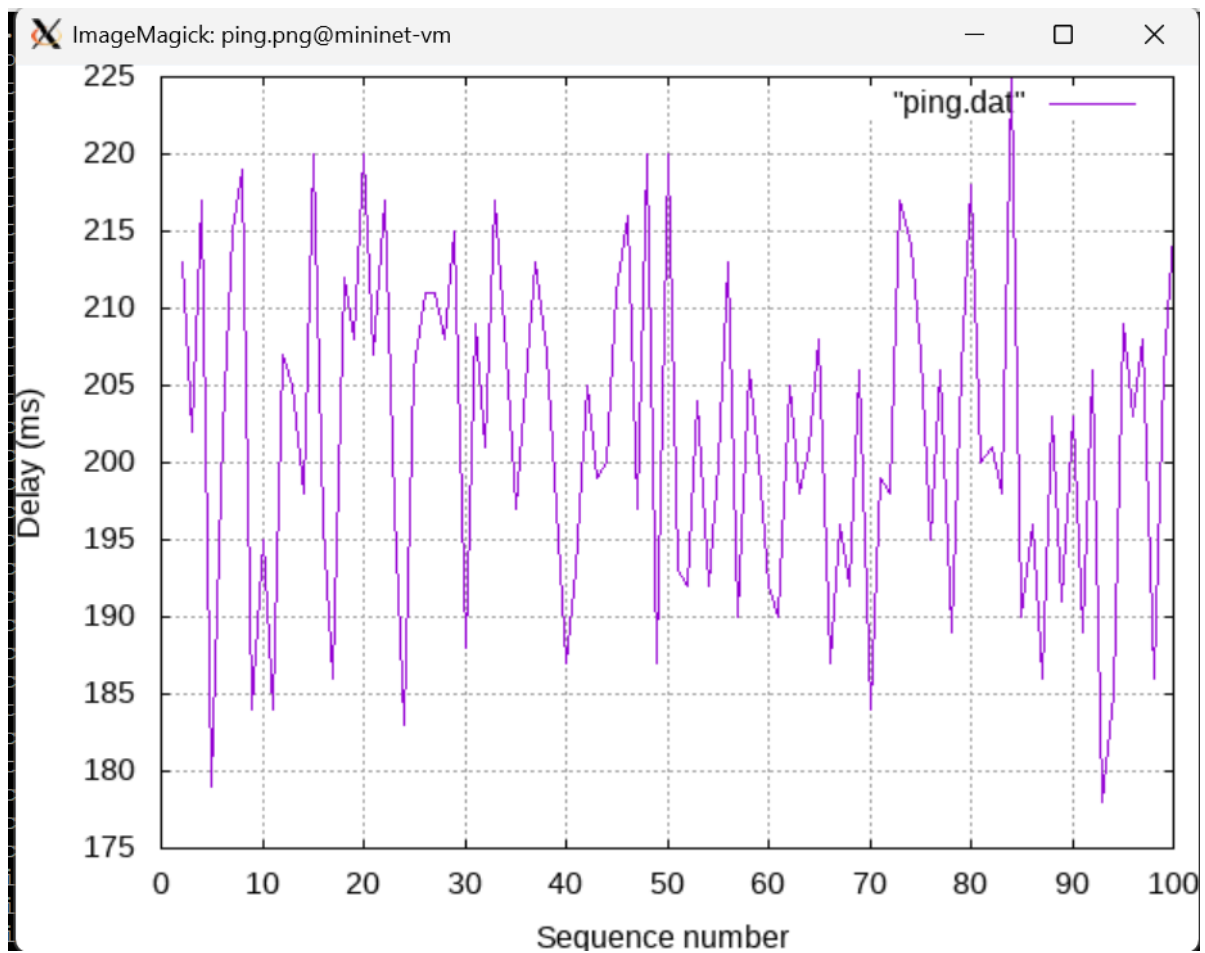


Рис. 3.50: Просмотр графика

4 Выводы

В ходе выполнения лабораторной работы мы познакомились 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.

Список литературы

1. NETEM [Электронный ресурс]. URL: <https://habr.com/ru/articles/949088/>.