

# **Лабораторная работа №4**

**Моделирование сетей передачи данных**

Еюбоглу Тимур

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>6</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>3</b>	<b>Вывод</b>	<b>37</b>
<b>4</b>	<b>Список литературы. Библиография</b>	<b>38</b>

# Список иллюстраций

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

2.32	Просмотр графика . . . . .	20
2.33	Удаление первой строчки из файла ping.dat . . . . .	21
2.34	Повторное построение графика . . . . .	22
2.35	Просмотр графика . . . . .	23
2.36	Разработка скрипта для вычисления на основе данных файла ping.dat минимального, среднего, максимального и стандартного отклонения времени приёма-передачи . . . . .	23
2.37	Добавление правила запуска скрипта в Makefil . . . . .	24
2.38	Проверка . . . . .	25
2.39	Воспроизводимый эксперимент по изменению задержки . . . . .	26
2.40	Воспроизводимый эксперимент по изменению задержки . . . . .	26
2.41	Просмотр графика . . . . .	27
2.42	Воспроизводимый эксперимент по изменению джиттера . . . . .	28
2.43	Воспроизводимый эксперимент по изменению джиттера . . . . .	29
2.44	Просмотр графика . . . . .	30
2.45	Воспроизводимый эксперимент по изменению значения корреляции для джиттера и задержки . . . . .	31
2.46	Воспроизводимый эксперимент по изменению значения корреляции для джиттера и задержки . . . . .	32
2.47	Просмотр графика . . . . .	33
2.48	Воспроизводимый эксперимент по изменению распределения времени задержки в эмулируемой глобальной сети . . . . .	34
2.49	Воспроизводимый эксперимент по изменению распределения времени задержки в эмулируемой глобальной сети . . . . .	35
2.50	Просмотр графика . . . . .	36

## **Список таблиц**

# 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.

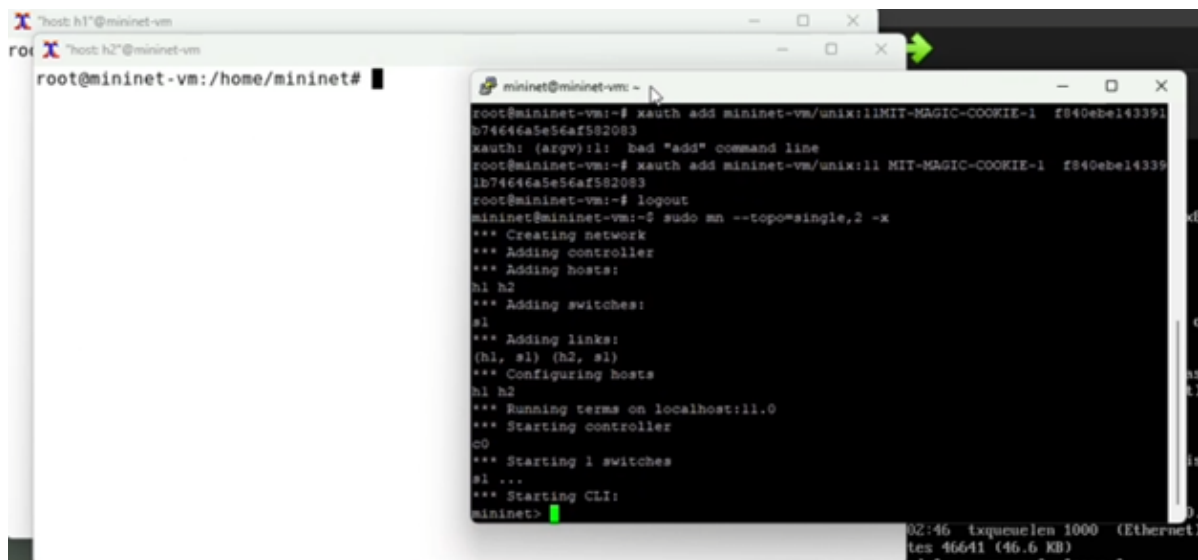
## 2 Выполнение лабораторной работы

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

```
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:11 MIT-MAGIC-COOKIE-1 f840ebel43391b74646a5e56af582083
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth add mininet-vm/unix:11MIT-MAGIC-COOKIE-1 f840ebel43391b74646a5e56af582083
xauth: (argv):1: bad "add" command line
root@mininet-vm:~# xauth add mininet-vm/unix:11 MIT-MAGIC-COOKIE-1 f840ebel43391b74646a5e56af582083
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):



```
root@mininet-vm:/home/mininet#
root@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:11.0
*** 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):

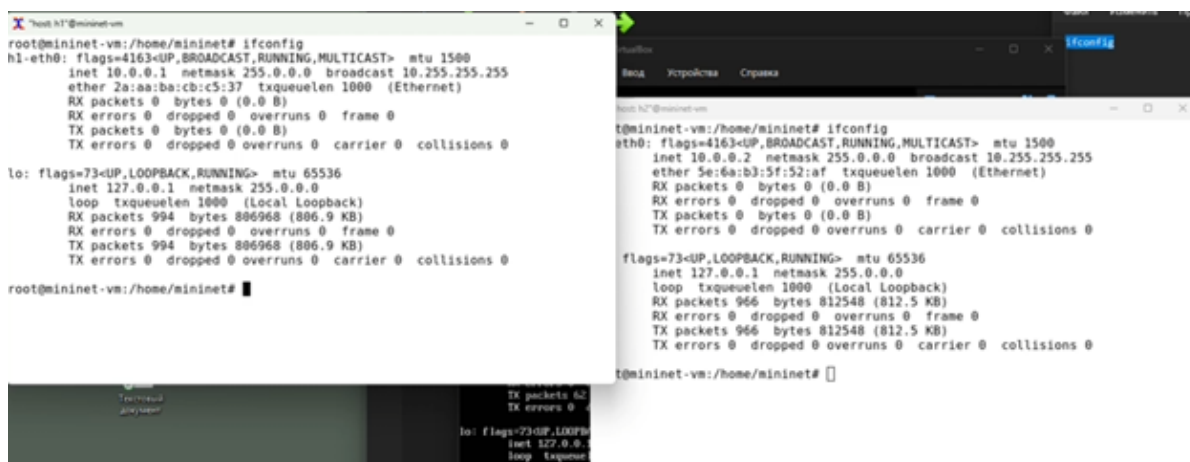


Рис. 2.3: Отображение информации их сетевых интерфейсов и IP-адресов

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



Рис. 2.4: Проверка подключения между хостами h1 и h2

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

```
--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5091ms
rtt min/avg/max/mdev = 0.050/0.327/1.398/0.488 ms
root@mininet-vn:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms
root@mininet-vn:/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):

```
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=101 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=100 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 500ms
rtt min/avg/max/mdev = 100.071/100.507/101.183/0.450 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
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.056 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.054 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.073 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.080 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.060 ms

--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 510ms
rtt min/avg/max/mdev = 0.054/0.203/0.895/0.309 ms
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h2-eth0 root netem delay 100ms
```

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):

```

root@mininet-vm:/home/mininet# sudo tc qdisc add dev h2-eth0 root netem delay 100ms
root@mininet-vm:/home/mininet# ping -c 6 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data:
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=201 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=202 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=201 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=202 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=201 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=201 ms

--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 200.724/201.278/201.968/0.411 ms

```

Рис. 2.8: Проверка

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

```

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 100.071/100.507/101.183/0.450 ms
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 50ms
Error: Exclusivity flag on, cannot modify.
root@mininet-vm:/home/mininet# sudo tc qdisc change dev h1-eth0 root netem delay 50ms
root@mininet-vm:/home/mininet#

```

```

64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=202 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=201 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=202 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=201 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=201 ms
--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 200.724/201.278/201.968/0.411 ms
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h2-eth0 root netem delay 50ms
Error: Exclusivity flag on, cannot modify.
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):

```

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=101 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=102 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=100 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=101 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5010ms
rtt min/avg/max/mdev = 100.259/101.092/102.104/0.581 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):

```

rtt min/avg/max/mdev = 100.259/101.092/102.104/0.581 ms
root@mininet-virtual-machine:~# sudo tc qdisc del dev h1-eth0 root netem
root@mininet-virtual-machine:~#
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=101 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=101 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=100 ms

--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5007ms
rtt min/avg/max/mdev = 100.137/101.030/101.907/0.649 ms
root@mininet-virtual-machine:~# sudo tc qdisc del dev h2-eth0 root netem
root@mininet-virtual-machine:~#

```

Рис. 2.11: Восстановление конфигураций по умолчанию

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

```

6 packets transmitted, 6 received, 0% packet loss, time 5115ms
rtt min/avg/max/mdev = 0.045/0.218/0.576/0.189 ms
root@mininet-virtual-machine:~# sudo tc qdisc add dev h1-eth0 root netem delay 100ms 10ms
root@mininet-virtual-machine:~#

```

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  $\pm 10$  мс, используя в терминале хоста h1 команду ping с параметром -c 6 (рис. 2.13):

```

00ms 10ms
root@mininet-virtual-machine:~# 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=106 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=98.4 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=94.2 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=107 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=94.1 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 94.073/100.060/106.534/4.989 ms
root@mininet-virtual-machine:~#

```

Рис. 2.13: Проверка

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

```

rtt min/avg/max/mdev = 94.073/100.060/106.534/4.989 ms
root@mininet-virtual-machine:~# sudo tc qdisc del dev h1-eth0 root netem
root@mininet-virtual-machine:~#

```

Рис. 2.14: Восстановление конфигурации интерфейса по умолчанию

Добавим на интерфейсе хоста h1 задержку в 100 мс с вариацией  $\pm 10$  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  $\pm 10$  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):

```

host: h1@mininet-vm
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=91.2 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=99.5 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=94.0 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=96.0 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=103 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=100 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=105 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=96.8 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=92.5 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=97.8 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=91.0 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=104 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=96.7 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=107 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=99.4 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=97.7 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=92.5 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=104 ms

--- 10.0.0.2 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19033ms
rtt min/avg/max/mdev = 90.995/98.435/107.431/4.703 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):

```

20 packets transmitted, 20 received, 0% packet loss, time 19033ms
rtt min/avg/max/mdev = 90.995/98.435/107.431/4.703 ms
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 1
00ms 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):

```

root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 1
00ms 20ms distribution normal
root@mininet-vm:/home/mininet# ping -c 10 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=102 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=49.4 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=126 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=84.0 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=137 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=96.2 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=90.5 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=94.3 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=122 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=105 ms

--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9014ms
rtt min/avg/max/mdev = 49.384/100.638/136.604/23.466 ms
root@mininet-vm:/home/mininet# █

```

Рис. 2.18: Проверка

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

```

10 packets transmitted, 10 received, 0% packet loss, time 9014ms
rtt min/avg/max/mdev = 49.384/100.638/136.604/23.466 ms
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):

```
mininet@mininet-vm: ~  
s1  
*** Adding links:  
(h1, s1) (h2, s1)  
*** Configuring hosts  
h1 h2  
*** Running terms on localhost:11.0  
*** 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 838.149 seconds  
mininet@mininet-vm:~$
```

Рис. 2.20: Завершение работы mininet в интерактивном режиме

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

```
*** Stopping 2 hosts  
h1 h2  
*** Done  
completed in 838.149 seconds  
mininet@mininet-vm:~$ sudo apt-get update  
Hit:1 http://security.ubuntu.com/ubuntu focal-security InRelease  
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease  
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease  
Hit:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease  
Reading package lists... Done  
mininet@mininet-vm:~$  
  
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255  
    ether 08:00:27:06:02:46 txqueuelen 1000 (Ethernet)  
    RX packets 60 bytes 6995 (6.9 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0
```

Рис. 2.21: Обновление репозитиев программного обеспечения на виртуальной машине

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



```
mininet@mininet-vm: ~  
python3-talloc python3-tz python3-wadllib rtkit rygel samba-libs sane-utils  
session-migration sgml-base sgml-data switcheroo-control  
system-config-printer system-config-printer-common  
system-config-printer-udev ubuntu-docs ubuntu-session ubuntu-wallpapers  
ubuntu-wallpapers-focal update-inetd upower usb-modeswitch  
usb-modeswitch-data usbmuxd va-driver-all vdpau-driver-all  
whoopsie-preferences wpasupplicant xdg-dbus-proxy xfonts-base  
xfonts-encodings xfonts-utils xml-core xserver-common xserver-xephyr  
xserver-xorg xserver-xorg-core xserver-xorg-input-all  
xserver-xorg-input-libinput xserver-xorg-input-wacom xserver-xorg-legacy  
xserver-xorg-video-all xserver-xorg-video-amdgpu xserver-xorg-video-ati  
xserver-xorg-video-fbdev xserver-xorg-video-intel xserver-xorg-video-nouveau  
xserver-xorg-video-qxl xserver-xorg-video-radeon xserver-xorg-video-vesa  
xserver-xorg-video-vmware xwayland yaru-theme-gnome-shell yelp yelp-xsl  
zenity zenity-common  
The following packages will be upgraded:  
dbus language-selector-common libcups2 libdbus-1-3 libdrm-amdgpu libegl1  
libgl1 libglvnd0 libglx0 libgstreamer-plugins-basel0-0  
libgstreamer1.0-0 libpl1-kit0 libpolkit-gobject-1-0 libpulse0 libtdb1  
libwebp-mux3  
17 upgraded, 410 newly installed, 0 to remove and 377 not upgraded.  
Need to get 181 MB of archives.  
After this operation, 744 MB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

Рис. 2.22: Установка пакета geeqie

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

```
Processing triggers for rygel (0.38.3-lubuntu) ...  
Processing triggers for sgml-base (1.29.1) ...  
mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_i/expname  
mininet@mininet-vm:~$
```

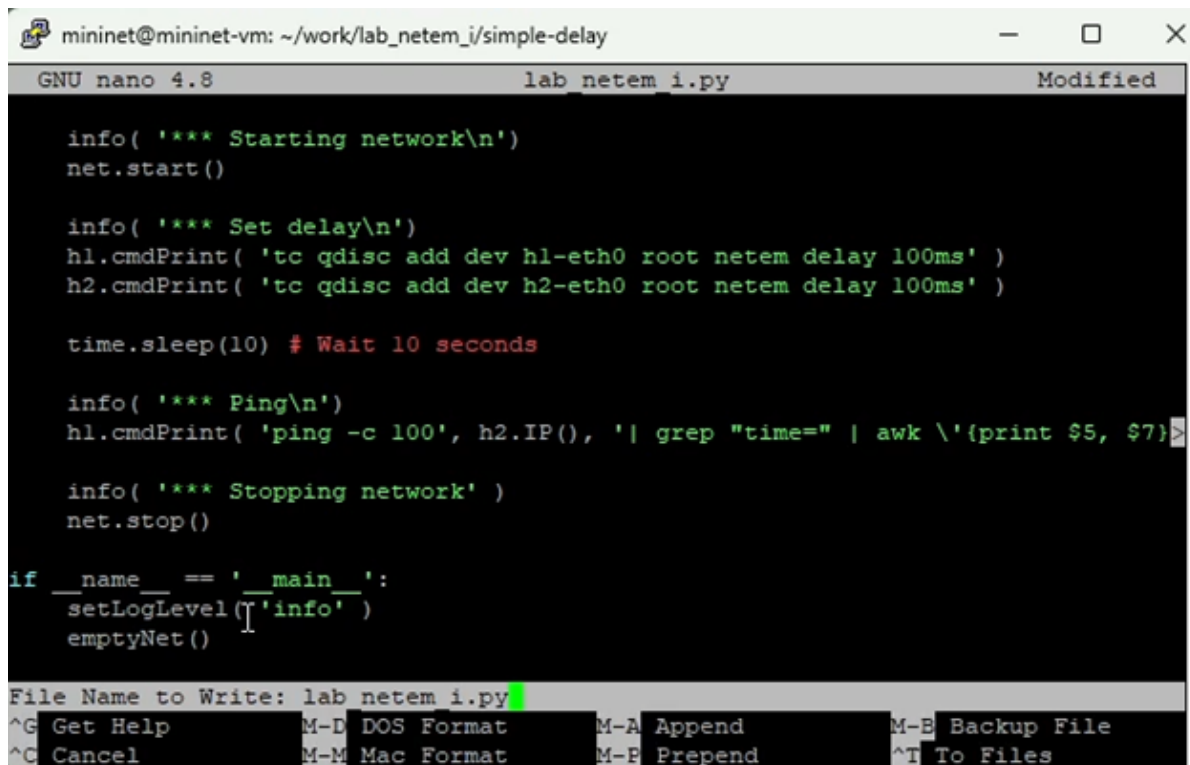
Рис. 2.23: Создание нового каталога

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

```
mininet@mininet-vm:~$ 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):



```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
GNU nano 4.8 lab_netem_i.py Modified

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}\'' )

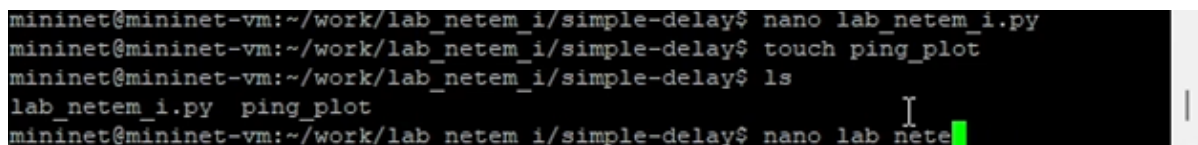
info( '*** Stopping network' )
net.stop()

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

File Name to Write: lab_netem_i.py
^G Get Help      M-D DOS Format  M-A Append      M-B Backup File
^C Cancel        M-M Mac Format  M-P Prepend     ^T To Files
```

Рис. 2.25: Создание скрипта lab\_netem\_i.py для эксперимента

Создадим файл ping\_plot (рис. 2.26):

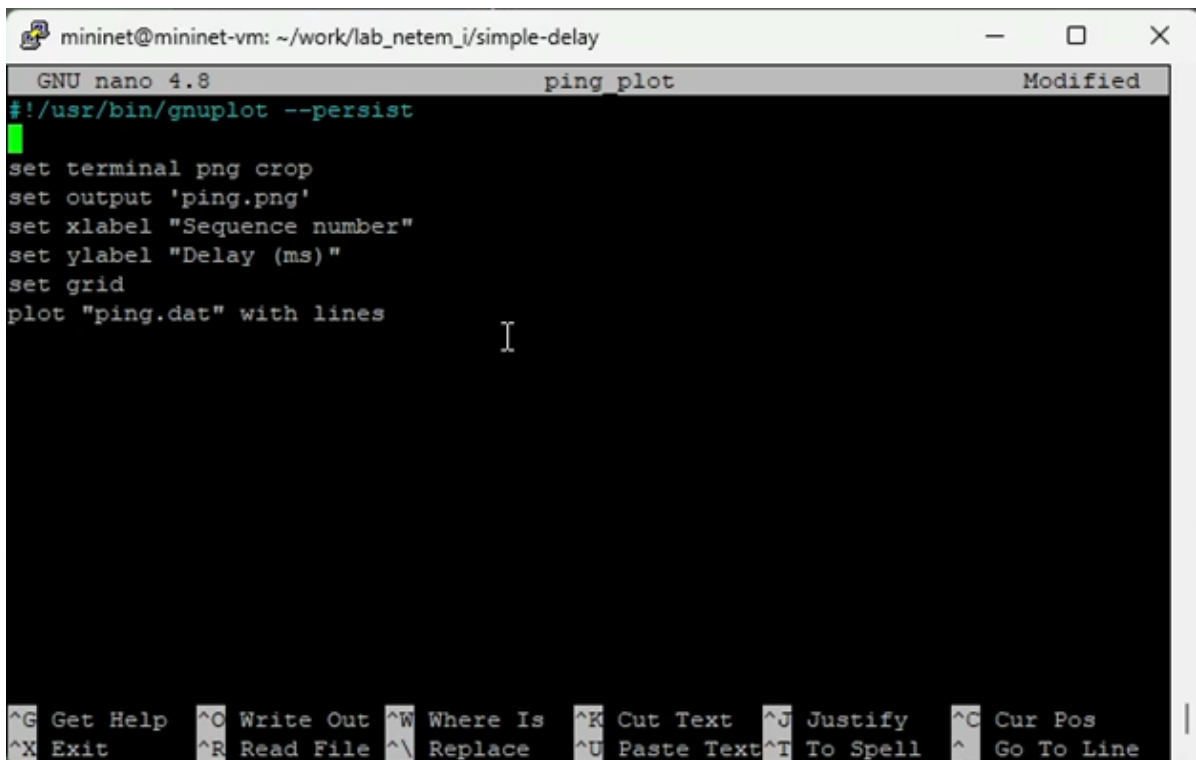


```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nano lab_netem_i.py
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ touch ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ ls
lab_netem_i.py  ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nano lab_netem_i.py
```

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
GNU nano 4.8 ping_plot Modified
#!/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
```

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
```

Рис. 2.29: Создание файла Makefile

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

```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
/home/mi~Makefile  [-M--]  0 L:[ 1+10 11/ 11] *(140 / 159b) 0009 0x009 [*][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

1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit
```

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

Рис. 2.31: Выполнение эксперимента

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

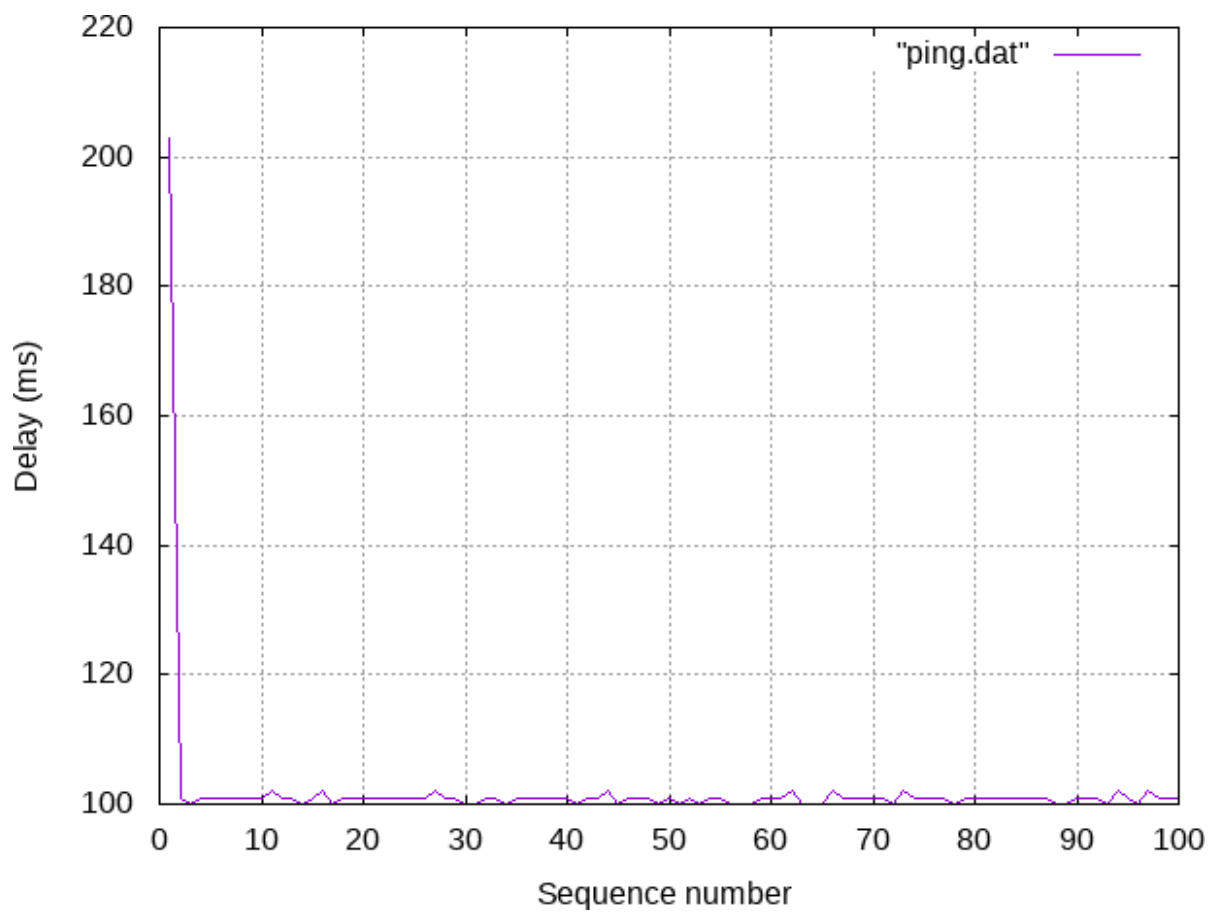


Рис. 2.32: Просмотр графика

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

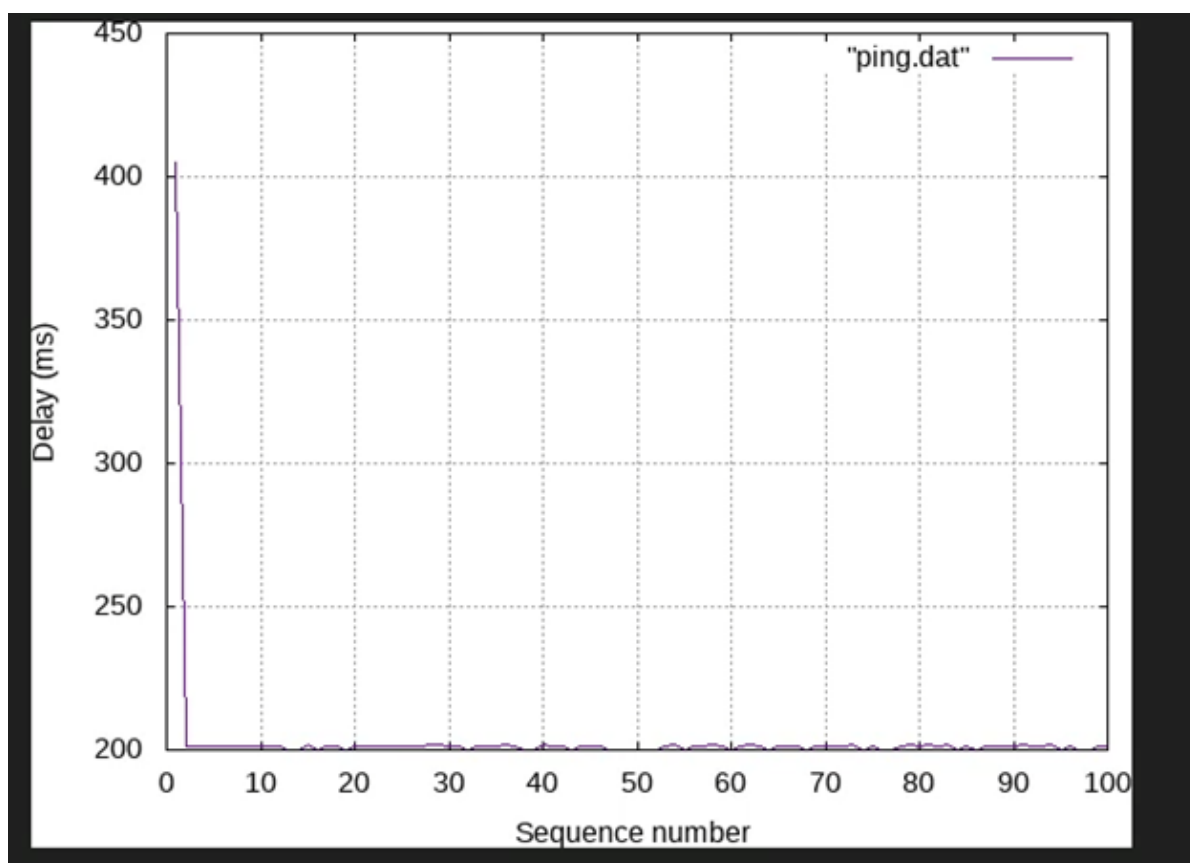
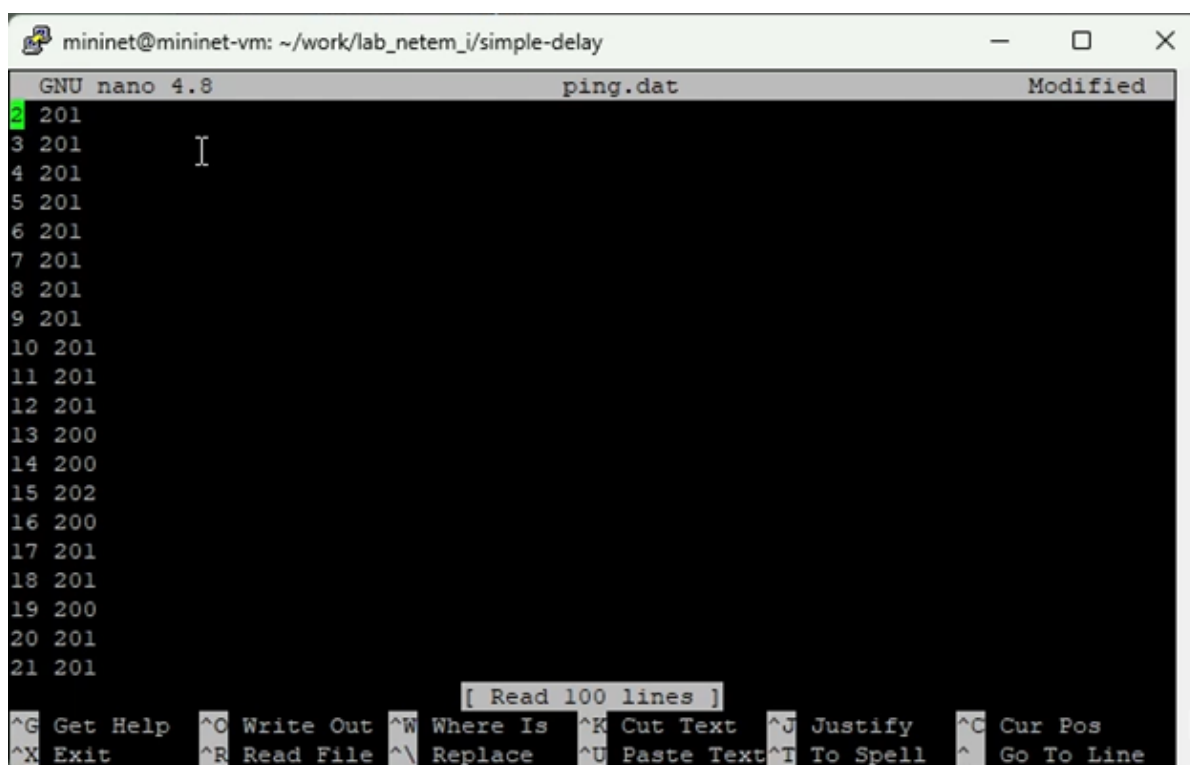


Рис. 2.33: Удаление первой строки из файла ping.dat



```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
GNU nano 4.8 ping.dat Modified
2 201
3 201
4 201
5 201
6 201
7 201
8 201
9 201
10 201
11 201
12 201
13 200
14 200
15 202
16 200
17 201
18 201
19 200
20 201
21 201
[ Read 100 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

Рис. 2.34: Повторное построение графика

Посмотрим заново построенный график (рис. 2.35):

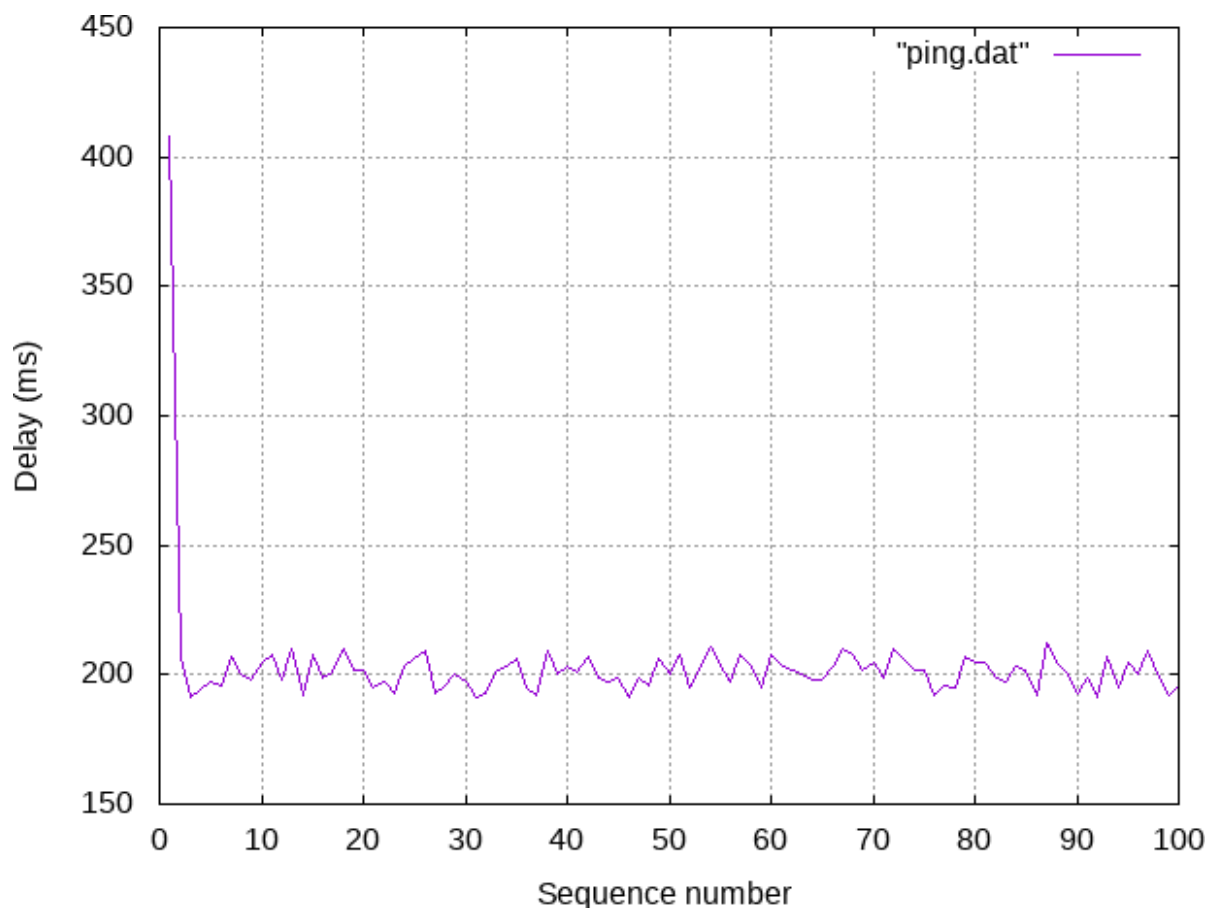


Рис. 2.35: Просмотр графика

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_1/simple-delay$ nano ping.dat
mininet@mininet-vm:~/work/lab_netem_1/simple-delay$ make ping.png
./ping_plot
mininet@mininet-vm:~/work/lab_netem_1/simple-delay$
```

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

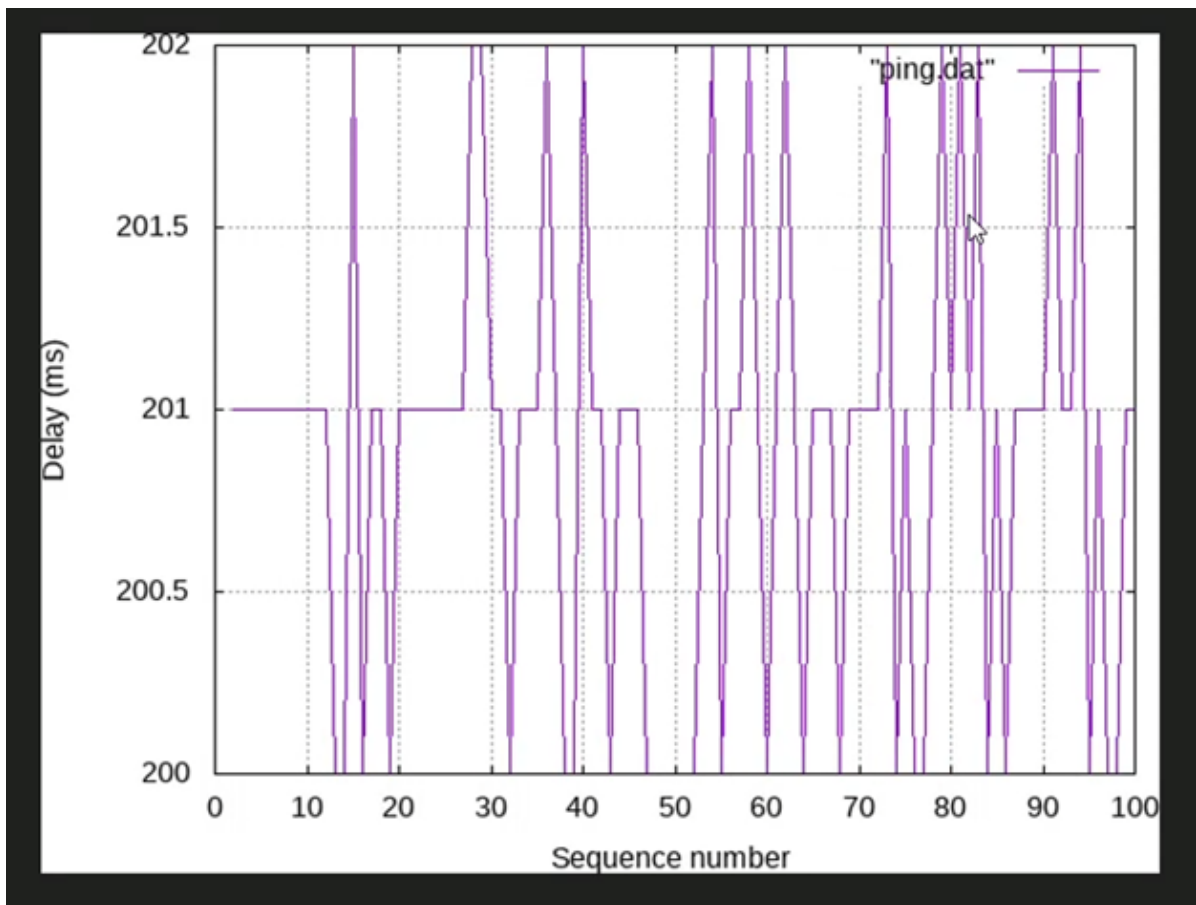
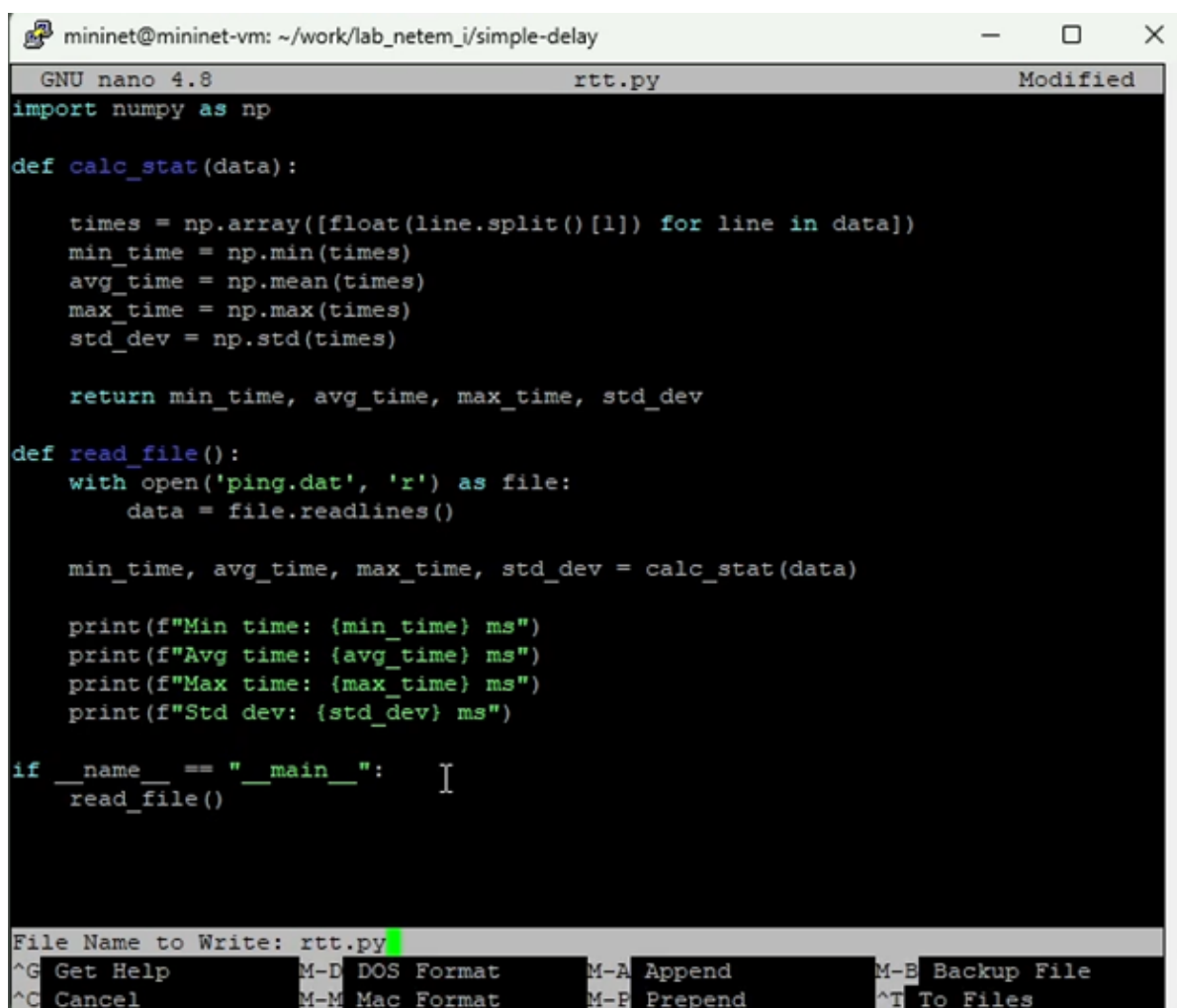


Рис. 2.37: Добавление правила запуска скрипта в Makefil





```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
GNU nano 4.8 rtt.py Modified
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()
```

File Name to Write: rtt.py

^G Get Help	M-D DOS Format	M-A Append	M-B Backup File
^C Cancel	M-M Mac Format	M-P Prepend	^T To Files

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

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

ping.png: ping.dat
    ./ping_plot

stats: ping.dat
    python rtt.py

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

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

Рис. 2.39: Воспроизводимый эксперимент по изменению задержки

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make stats
python rtt.py
Min time: 200.0 ms
Avg time: 200.87878787878788 ms
Max time: 202.0 ms
Std dev: 0.6239775843022425 ms
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$
```

Рис. 2.40: Воспроизводимый эксперимент по изменению задержки

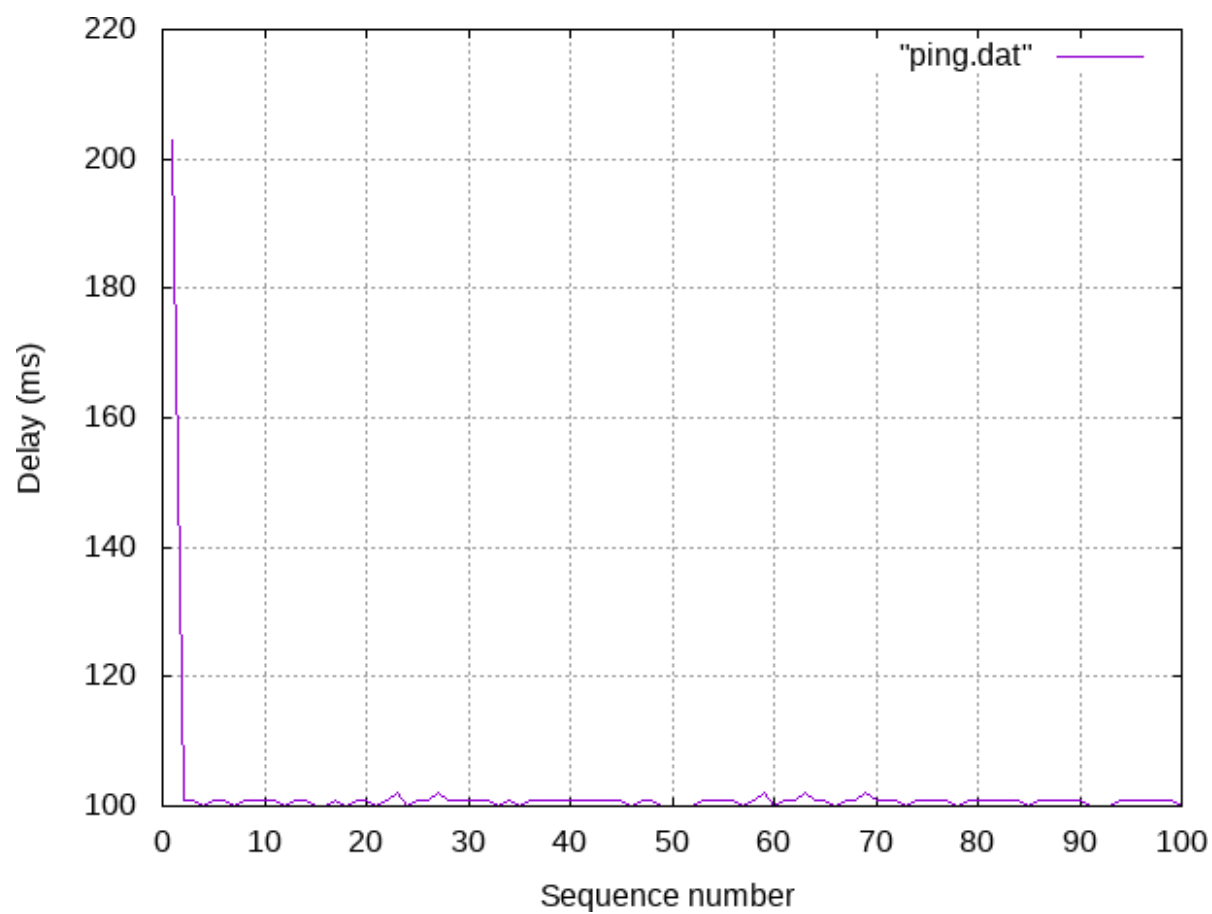
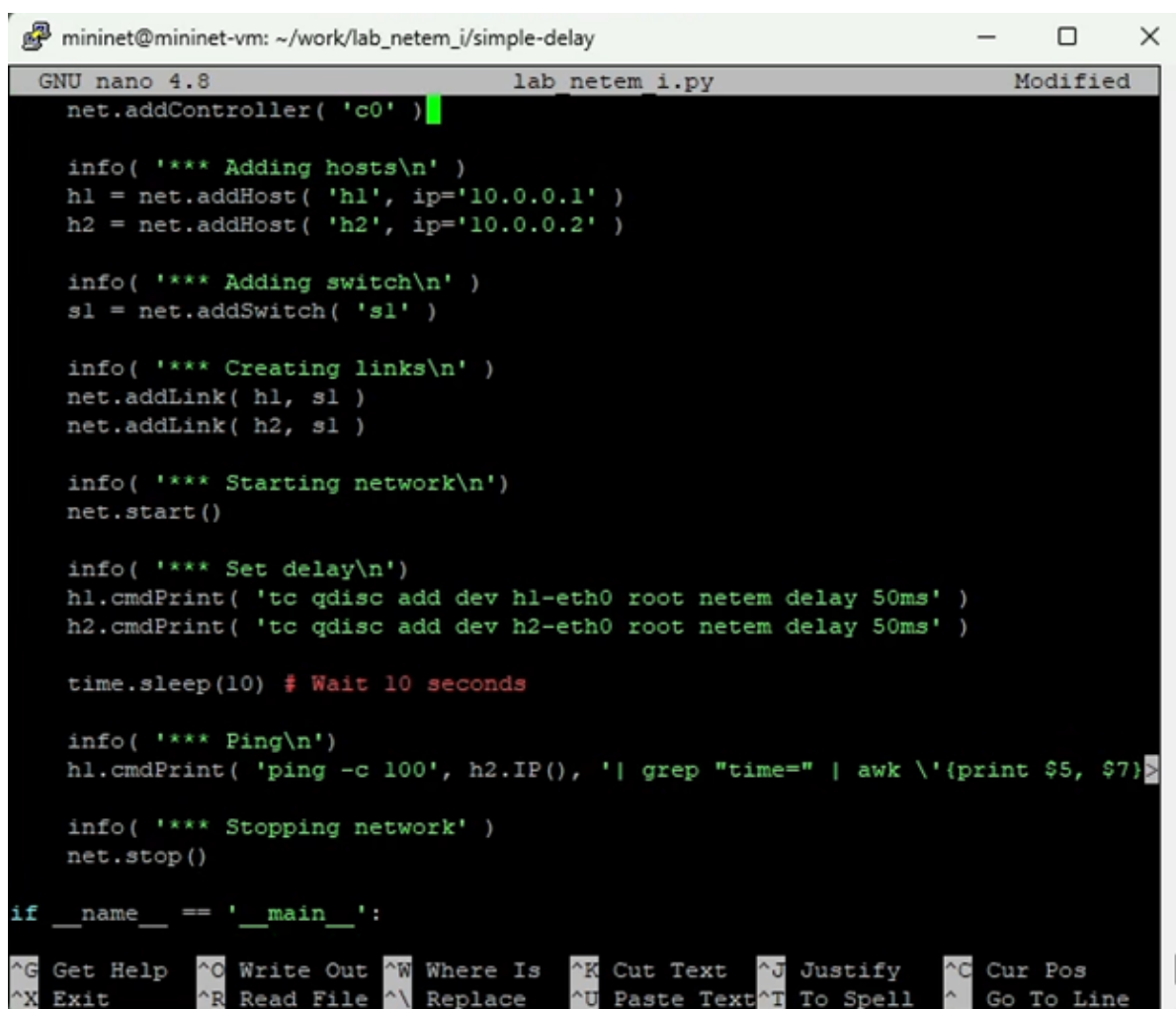


Рис. 2.41: Просмотр графика



The image shows a terminal window titled "mininet@mininet-vm: ~/work/lab\_netem\_i/simple-delay". The terminal is running the GNU nano 4.8 text editor, editing a file named "lab\_netem i.py". The script contains the following Python code:

```
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' )
net.stop()

if __name__ == '__main__':
```

At the bottom of the terminal, there is a status bar with various keyboard shortcuts for the nano editor, such as ^G Get Help, ^O Write Out, ^W Where Is, ^K Cut Text, ^J Justify, ^C Cur Pos, ^X Exit, ^R Read File, ^\ Replace, ^U Paste Text, ^T To Spell, and ^\_ Go To Line.

Рис. 2.42: 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 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: 101.82 ms
Max time: 203.0 ms
Std dev: 10.181728733373324 ms
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$

```

Рис. 2.43: Воспроизводимый эксперимент по изменению джиттера

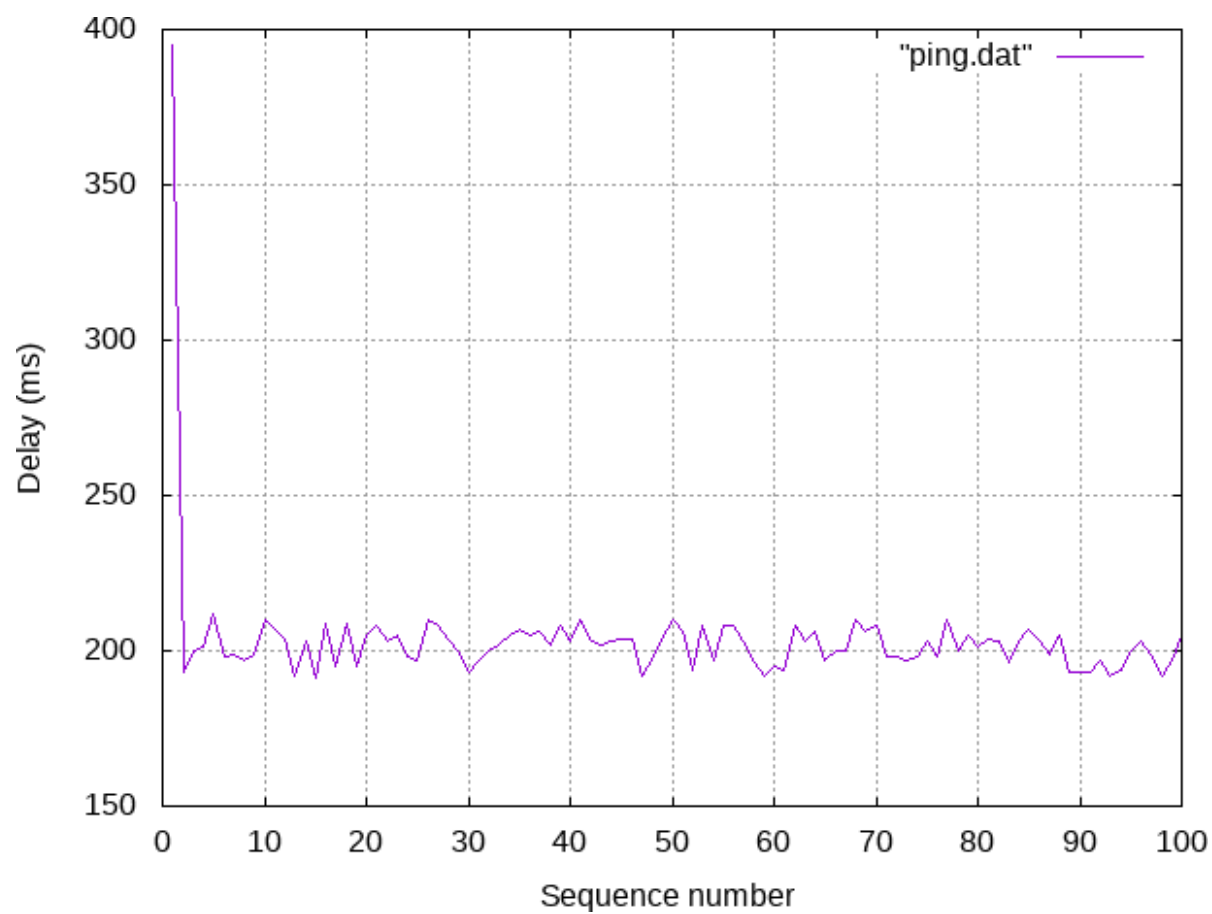


Рис. 2.44: Просмотр графика

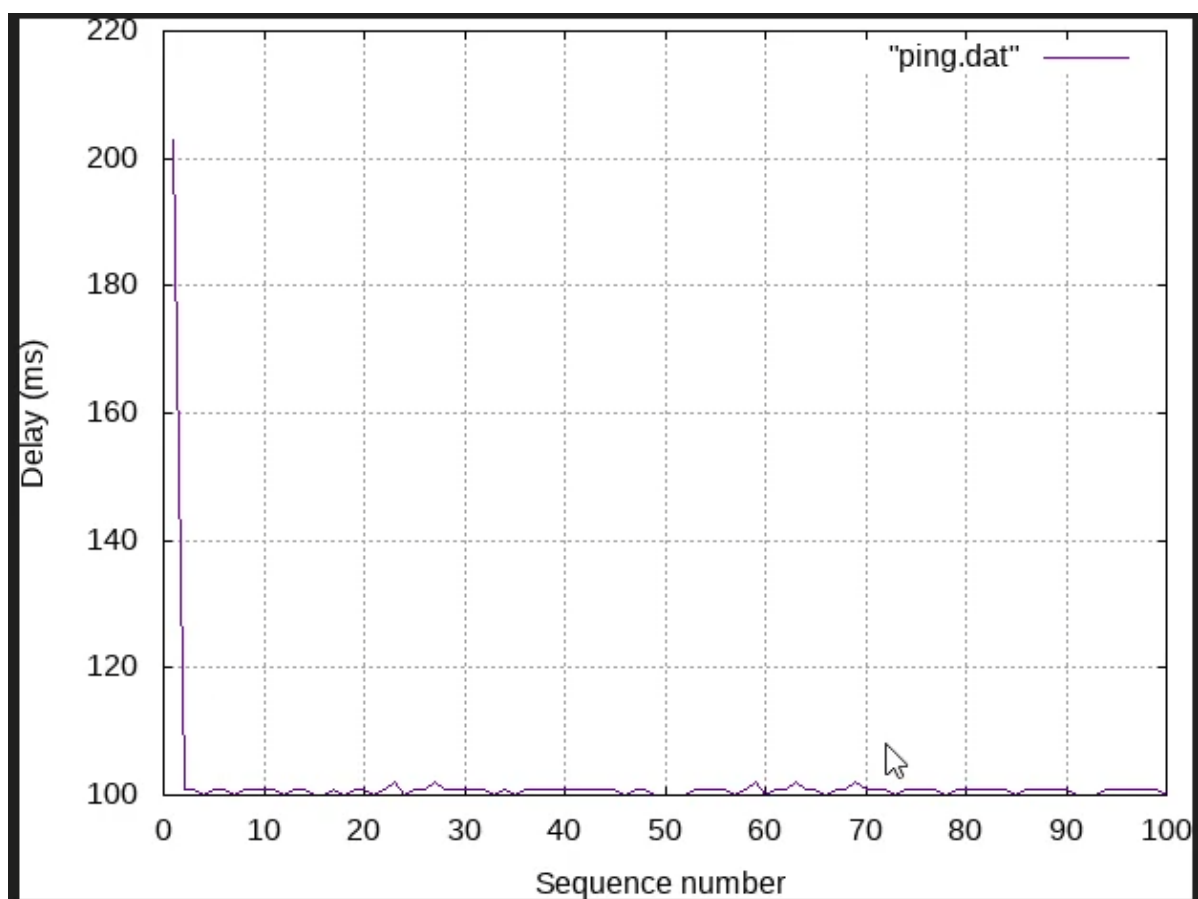
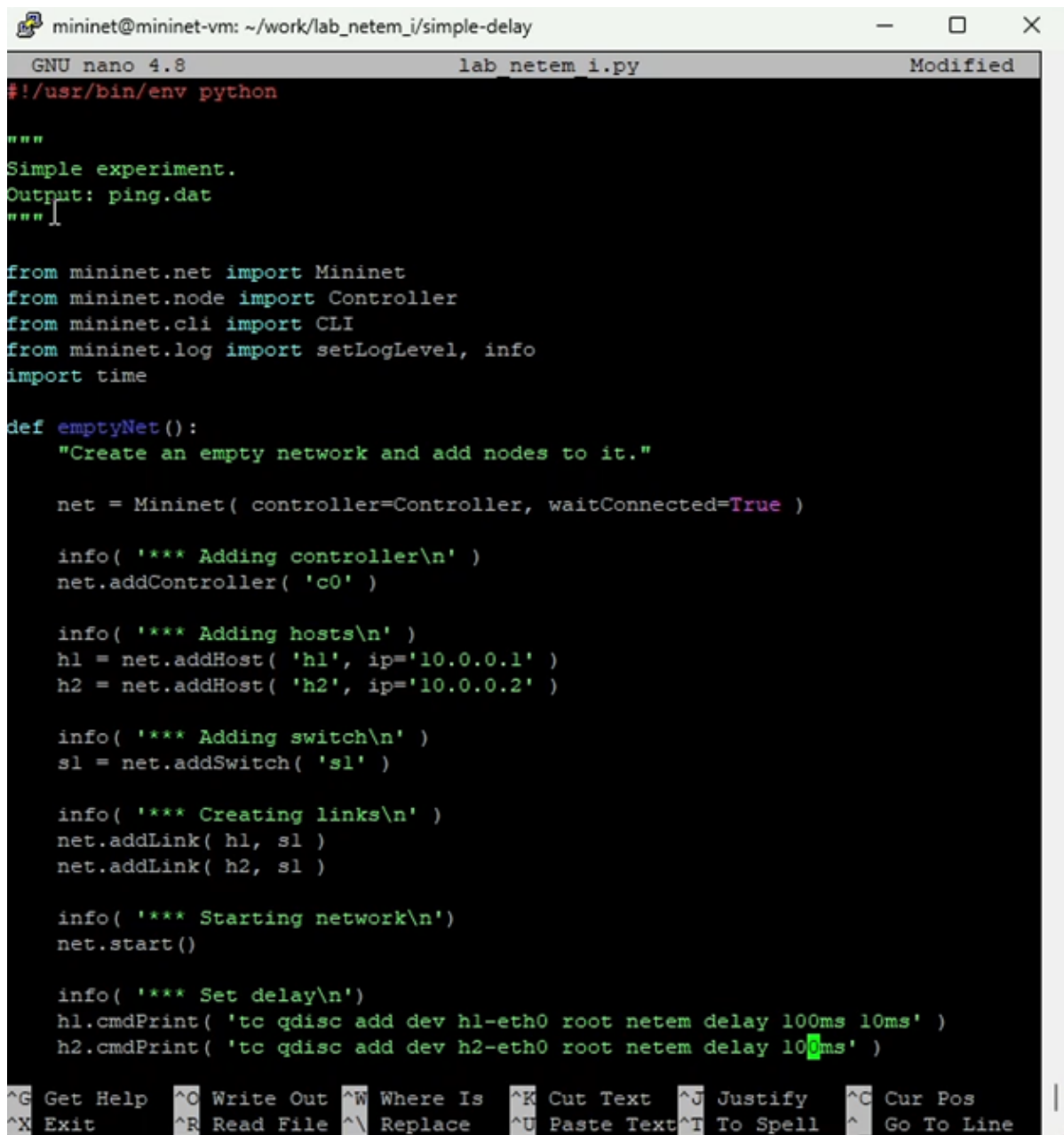


Рис. 2.45: Воспроизводимый эксперимент по изменению значения корреляции для джиттера и задержки



```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
GNU nano 4.8 lab_netem_i.py Modified
#!/usr/bin/env python

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

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

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

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

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

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

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

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

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

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

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

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



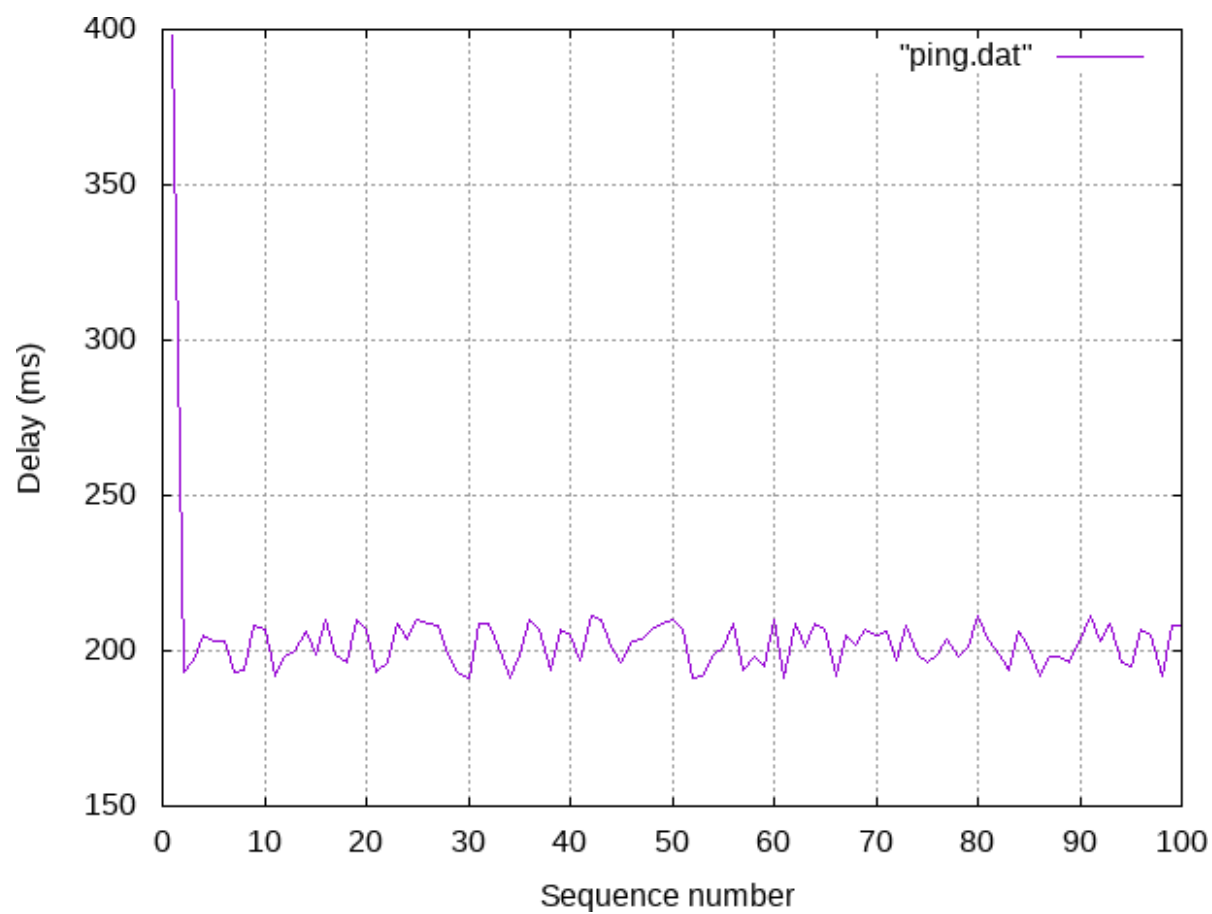


Рис. 2.47: Просмотр графика

```

mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make clean
rm -f *.dat *.png
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make
sudo python lab_netem_i.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
*** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
*** Set delay
*** h1 : ('tc qdisc add dev h1-eth0 root netem delay 100ms 10ms',)
*** h2 : ('tc qdisc add dev h2-eth0 root netem delay 100ms',)
*** Ping
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "time=" | awk \'{print $5, $7}\'' |
sed -e \'/s/time=//g\' -e \'/s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$

```

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

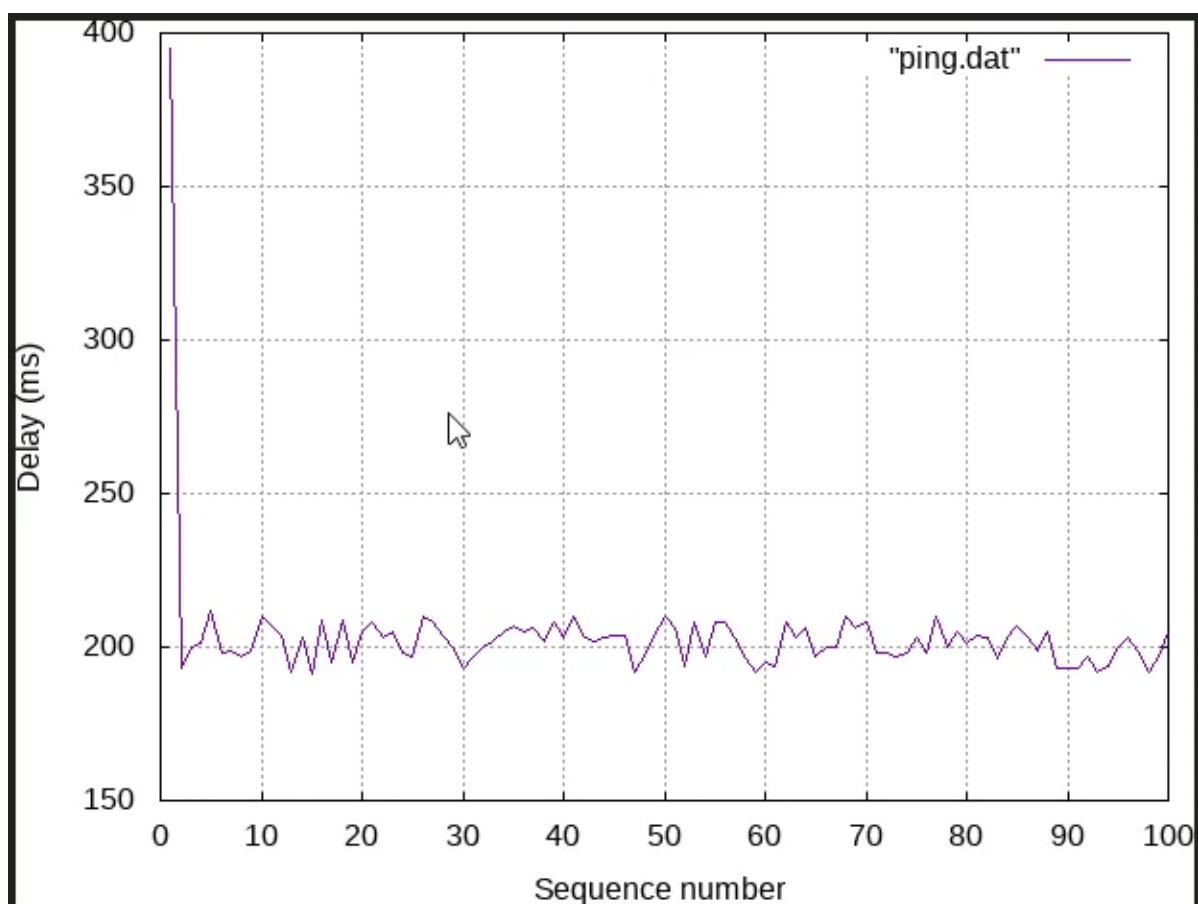


Рис. 2.49: Воспроизводимый эксперимент по изменению распределения времени задержки в эмулируемой глобальной сети

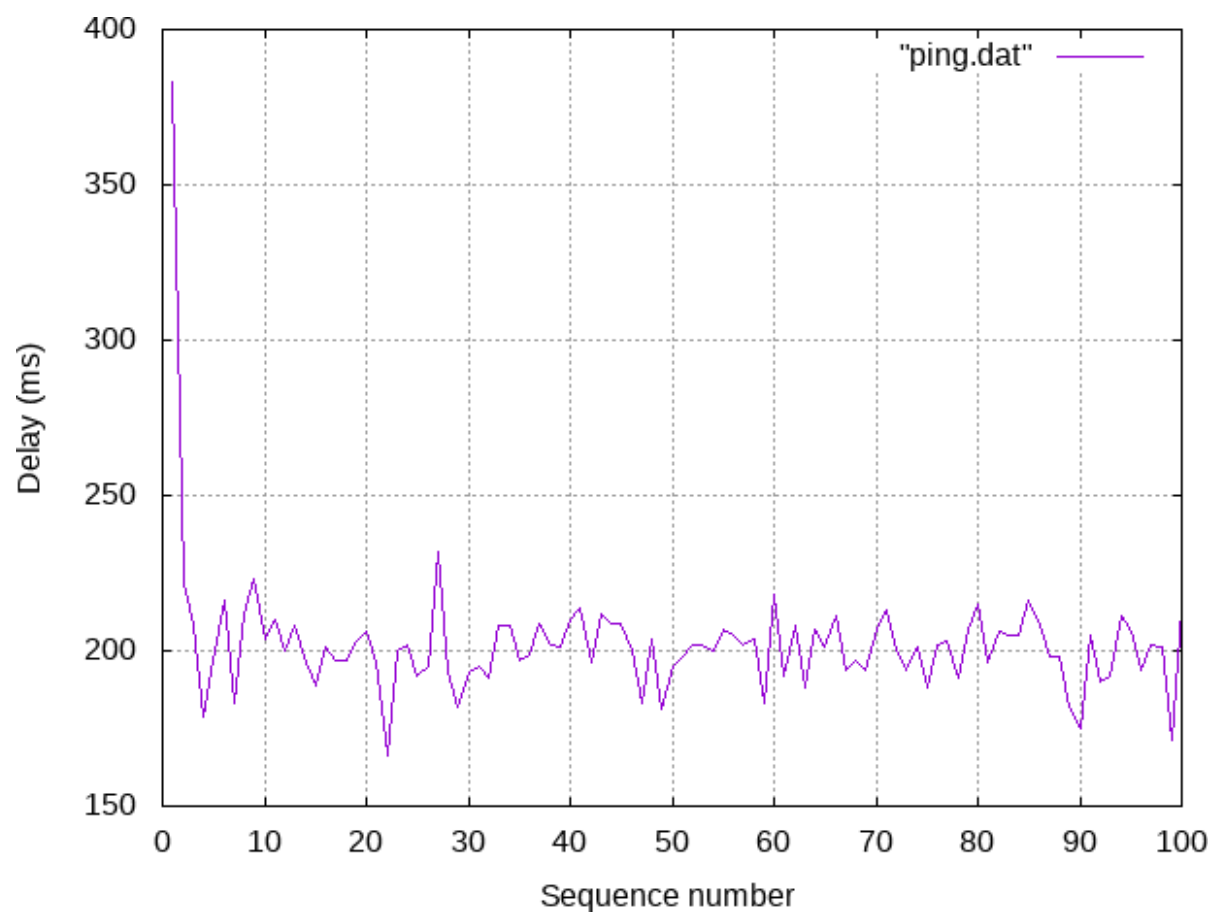


Рис. 2.50: Просмотр графика

## 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/>