

Отчёт по лабораторной работе №4 по дисциплине Моделирование сетей передачи данных

Эмуляция и измерение задержек в глобальных сетях

Шаповалова Диана Дмитриевна, НПИбд-02-21, 1032211220

Содержание

1	Цель работы	5
2	Выполнение работы	6
2.1	Запуск лабораторной топологии	6
2.2	Интерактивные эксперименты	7
2.3	Добавление/изменение задержки в эмулируемой глобальной сети	7
2.4	Изменение задержки в эмулируемой глобальной сети	9
2.5	Восстановление исходных значений (удаление правил) задержки в эмулируемой глобальной сети	10
2.6	Добавление значения дрожания задержки в интерфейс подключения к эмулируемой глобальной сети	11
2.7	Добавление значения корреляции для джиттера и задержки в интерфейс подключения к эмулируемой глобальной сети	12
2.8	Распределение задержки в интерфейсе подключения к эмулируемой глобальной сети	13
2.9	Воспроизведение экспериментов	15
2.10	Предварительная подготовка	15
2.11	Добавление задержки для интерфейса, подключающегося к эмулируемой глобальной сети	15
2.12	Задание для самостоятельной работы	19
2.13	Эксперимент по изменению задержки	21
2.14	Эксперимент по изменению джиттера	24
2.15	Эксперимент по изменению значения корреляции для джиттера и задержки	28
2.16	Распределения времени задержки в эмулируемой глобальной сети	31
3	Выводы	32
4	Список литературы	33

Список иллюстраций

2.1	Создаем топологию	7
2.2	Добавляем задержку	9
2.3	Изменяем задержку	10
2.4	Восстанавливаем конфигурацию	11
2.5	Добавление значения дрожания задержки	12
2.6	Добавление значения корреляции для джиттера и задержки	13
2.7	Распределение задержки в интерфейсе подключения	14
2.8	Создаем репозитории	15
2.9	Создаем скрипт lab_netem_i.py	16
2.10	Создаем скрипт для визуализации ping_plot	17
2.11	График simple-delay	18
2.12	График simple-delay без 1ой строчки в ping.dat	19
2.13	Эксперимент по изменению задержки	21
2.14	Эксперимент по изменению задержки	22
2.15	Эксперимент по изменению задержки	23
2.16	Эксперимент по изменению джиттера	24
2.17	Эксперимент по изменению джиттера	25
2.18	Эксперимент по изменению джиттера	26
2.19	Эксперимент по изменению значения корреляции для джиттера и задержки	28
2.20	Эксперимент по изменению значения корреляции для джиттера и задержки	29
2.21	Эксперимент по изменению значения корреляции для джиттера и задержки	30
2.22	Распределения времени задержки в эмулируемой глобальной сети	31

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 работы

2.1 Запуск лабораторной топологии

Запустите виртуальную среду с mininet.

Из основной ОС подключитесь к виртуальной машине:

```
ssh -Y mininet@192.168.x.y
```

В виртуальной машине mininet при необходимости исправьте права запуска X-соединения. Скопируйте значение куки (MIT magic cookie)¹ своего пользователя mininet в файл для пользователя root

После выполнения этих действий графические приложения должны запускаться под пользователем mininet.

Задайте простейшую топологию, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8:

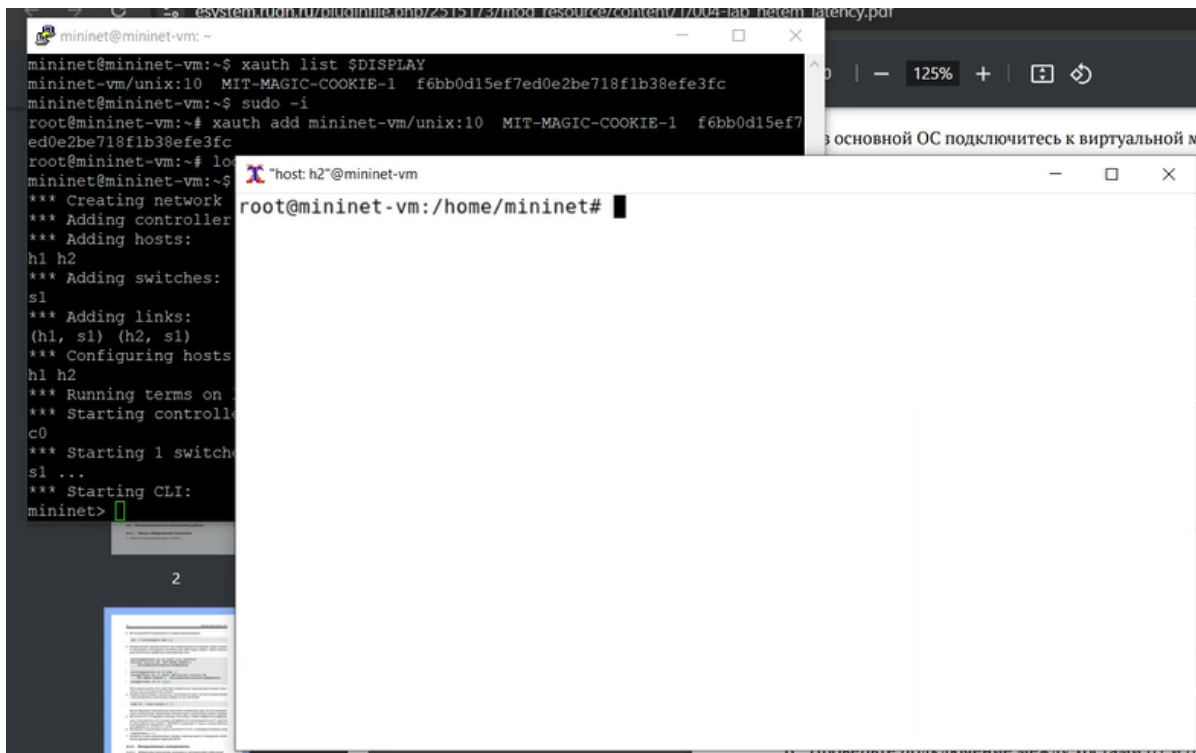


Рис. 2.1: Создаем топологию

2.2 Интерактивные эксперименты

2.3 Добавление/изменение задержки в эмулируемой глобальной сети

1. На хосте h1 добавьте задержку в 100 мс к выходному интерфейсу:

```
sudo tc qdisc add dev h1-eth0 root netem delay 100ms
```

Здесь:

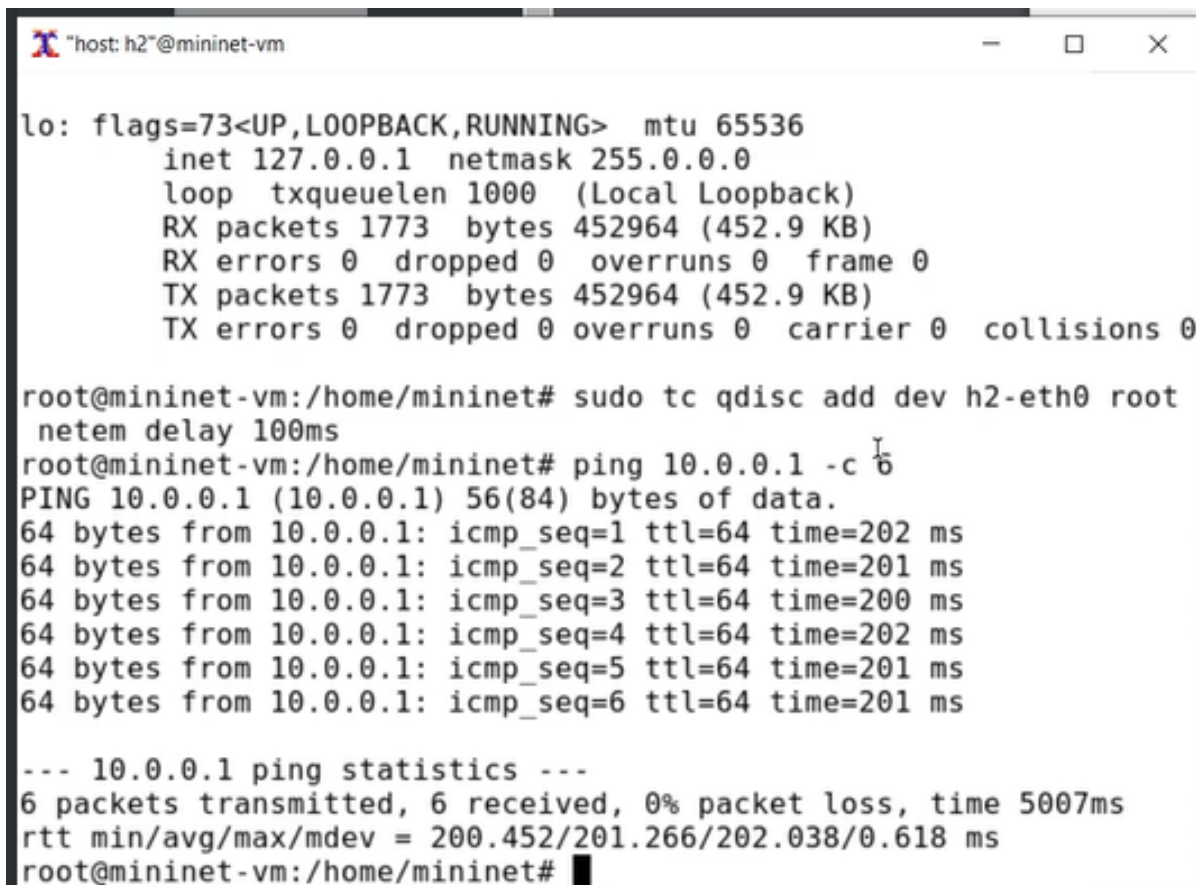
- sudo: выполнить команду с более высокими привилегиями;
- tc: вызвать управление трафиком Linux;
- qdisc: изменить дисциплину очередей сетевого планировщика;
- add: создать новое правило;
- dev h1-eth0: указать интерфейс, на котором будет применяться правило;

- netem: использовать эмулятор сети;
- delay 100ms: задержка ввода 100 мс.

2. Проверьте, что соединение от хоста h1 к хосту h2 имеет задержку 100 мс, используя команду ping с параметром -c 6 с хоста h1. Укажите в отчёте минимальное, среднее, максимальное и стандартное отклонение времени приёма-передачи (RTT).
3. Для эмуляции глобальной сети с двунаправленной задержкой необходимо к соответствующему интерфейсу на хосте h2 также добавить задержку в 100 миллисекунд:

```
sudo tc qdisc add dev h2-eth0 root netem delay 100ms
```

4. Проверьте, 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. Укажите в отчёте минимальное, среднее, максимальное и стандартное отклонение времени приёма-передачи (RTT).



```
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 1773 bytes 452964 (452.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1773 bytes 452964 (452.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet# sudo tc qdisc add dev h2-eth0 root
netem delay 100ms
root@mininet-vm:/home/mininet# ping 10.0.0.1 -c 6
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=202 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=201 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=200 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 5007ms
rtt min/avg/max/mdev = 200.452/201.266/202.038/0.618 ms
root@mininet-vm:/home/mininet#
```

Рис. 2.2: Добавляем задержку

2.4 Изменение задержки в эмулируемой глобальной сети

1. Измените задержку со 100 мс до 50 мс для отправителя h1:

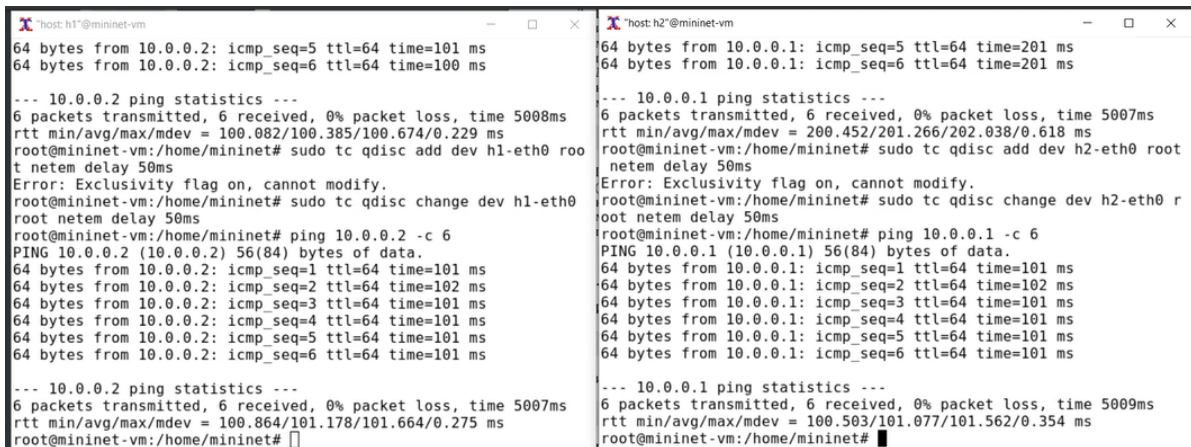
```
sudo tc qdisc change dev h1-eth0 root netem delay 50ms
```

и для получателя h2:

```
sudo tc qdisc change dev h2-eth0 root netem delay 50ms
```

2. Проверьте, что соединение от хоста h1 к хосту h2 имеет задержку 100 мс, используя команду ping с параметром -c 6 с терминала хоста h1. Укажите в

отчёте минимальное, среднее, максимальное и стандартное отклонение времени приёма-передачи (RTT).



The image shows two terminal windows side-by-side. The left window is titled 'host: h1@mininet-vm' and the right is 'host: h2@mininet-vm'. Both show the execution of 'tc qdisc add' on their respective interfaces (h1-eth0 and h2-eth0) with a netem delay of 50ms. This is followed by a 'ping' command to the other host. The ping statistics show a 0% packet loss and an average RTT of approximately 100ms. The right window also shows an 'Error: Exclusivity flag on, cannot modify.' message before the ping command is executed.

```
host: h1@mininet-vm
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 5008ms
rtt min/avg/max/mdev = 100.082/100.385/100.674/0.229 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# ping 10.0.0.2 -c 6
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=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 5007ms
rtt min/avg/max/mdev = 100.864/101.178/101.664/0.275 ms
root@mininet-vm:/home/mininet#

host: h2@mininet-vm
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 5007ms
rtt min/avg/max/mdev = 200.452/201.266/202.038/0.618 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 r
oot netem delay 50ms
root@mininet-vm:/home/mininet# ping 10.0.0.1 -c 6
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=101 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=102 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=101 ms
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=101 ms
--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 100.503/101.077/101.562/0.354 ms
root@mininet-vm:/home/mininet#
```

Рис. 2.3: Изменяем задержку

2.5 Восстановление исходных значений (удаление правил) задержки в эмулируемой глобальной сети

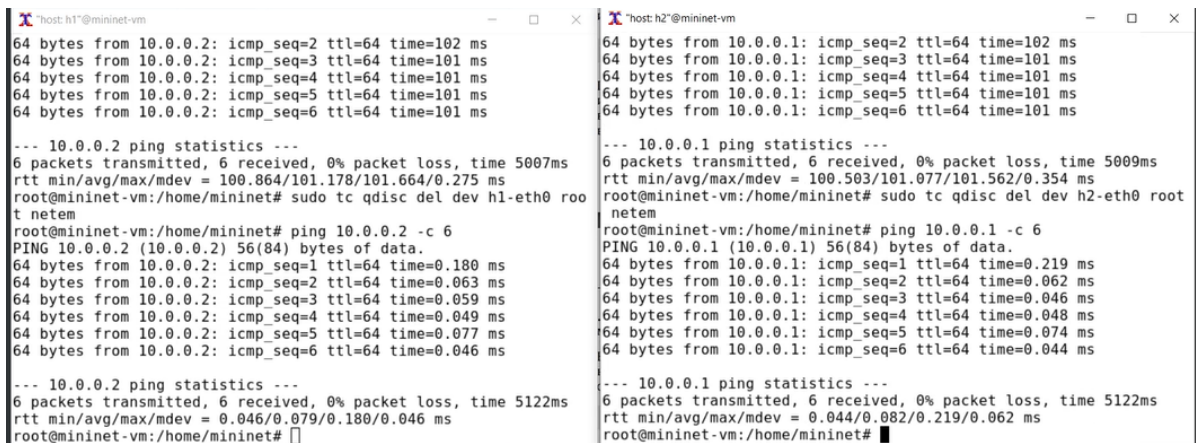
1. Восстановите конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса. Для отправителя h1:

```
sudo tc qdisc del dev h1-eth0 root netem
```

Для получателя h2:

```
sudo tc qdisc del dev h2-eth0 root netem
```

2. Проверьте, что соединение между хостом h1 и хостом h2 не имеет явно установленной задержки, используя команду ping с параметром -c 6 с терминала хоста h1. Укажите в отчёте минимальное, среднее, максимальное и стандартное отклонение времени приёма-передачи (RTT).



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

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5007ms
rtt min/avg/max/mdev = 100.864/101.178/101.664/0.275 ms
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root
root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 6
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.180 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.063 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.059 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.049 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.077 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.046 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5122ms
rtt min/avg/max/mdev = 0.046/0.079/0.180/0.046 ms
root@mininet-vm:/home/mininet#

host: h2@mininet-vm
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=102 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=101 ms
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=101 ms

--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 100.503/101.077/101.562/0.354 ms
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h2-eth0 root
root@mininet-vm:/home/mininet# ping 10.0.0.1 -c 6
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data:
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.219 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.062 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.046 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.048 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.074 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.044 ms

--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5122ms
rtt min/avg/max/mdev = 0.044/0.082/0.219/0.062 ms
root@mininet-vm:/home/mininet#
```

Рис. 2.4: Восстанавливаем конфигурацию

2.6 Добавление значения дрожания задержки в интерфейс подключения к эмулируемой глобальной сети

В сетях нет постоянной задержки. Она может варьироваться в зависимости от других потоков трафика, конкурирующих за тот же путь. Джиттер (jitter) — это изменение времени задержки. Параметры задержки описываются в терминах теории вероятностей средним значением μ , стандартным отклонением σ и корреляцией. По умолчанию NETEM использует равномерное распределение, так что задержка находится в пределах $\mu \pm \sigma$. Параметр корреляции управляет отношением между последовательными псевдослучайными значениями. 1. При необходимости восстановите конфигурацию интерфейсов по умолчанию на узлах h1 и h2:

```
sudo tc qdisc del dev h1-eth0 root netem
sudo tc qdisc del dev h2-eth0 root netem
```

2. Добавьте на узле h1 задержку в 100 мс со случайным отклонением 10 мс:

```
sudo tc qdisc add dev h1-eth0 root netem delay 100ms 10ms
```

3. Проверьте, 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`. Укажите в отчёте минимальное, среднее, максимальное и стандартное отклонение времени приёма-передачи (RTT).
4. Восстановите конфигурацию интерфейса по умолчанию на узле h1.

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms 10ms
root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 6
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=92.0 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=92.7 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=108 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=109 ms
```

Рис. 2.5: Добавление значения дрожания задержки

2.7 Добавление значения корреляции для джиттера и задержки в интерфейс подключения к эмулируемой глобальной сети

1. При необходимости восстановите конфигурацию интерфейсов по умолчанию на узлах h1 и h2.
2. Добавьте на интерфейсе хоста h1 задержку в 100 мс с вариацией ± 10 мс и значением корреляции в 25%:

```
sudo tc qdisc add dev h1-eth0 root netem delay 100ms 10ms 25%
```

3. Убедитесь, 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%. Используйте для этого в терминале хоста h1 команду ping с параметром -s 20. Укажите в отчёте минимальное, среднее, максимальное и стандартное отклонение времени приёма-передачи (RTT).

4. Восстановите конфигурацию интерфейса по умолчанию на узле h1.

```
t netem
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root
t netem delay 100ms 10ms 25%
root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 6
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=94.5 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=93.7 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=102 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=105 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=98.6 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=107 ms
```

Рис. 2.6: Добавление значения корреляции для джиттера и задержки

2.8 Распределение задержки в интерфейсе подключения к эмулируемой глобальной сети

NETEM позволяет пользователю указать распределение, которое описывает, как задержки изменяются в сети. В реальных сетях передачи данных задержки неравномерны, поэтому при моделировании может быть удобно использовать некоторое случайное распределение, например, нормальное, парето или парето-нормальное.

1. При необходимости восстановите конфигурацию интерфейсов по умолчанию на узлах h1 и h2.
2. Задайте нормальное распределение задержки на узле h1 в эмулируемой сети:

```
sudo tc qdisc add dev h1-eth0 root netem delay 100ms 20ms distribution normal
```

3. Убедитесь, что все пакеты, покидающие хост h1 на интерфейсе h1-eth0, будут иметь время задержки, которое распределено в диапазоне 100 мс \pm 20 мс. Используйте для этого команду ping на терминале хоста h1 с параметром -c 10.

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root
netem delay 100ms 20ms distribution normal
root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 10
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=91.8 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=82.9 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=79.7 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=105 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=122 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=108 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=84.4 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=130 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=115 ms

--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9015ms
rtt min/avg/max/mdev = 79.729/104.064/129.859/17.341 ms
root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 6
```

Рис. 2.7: Распределение задержки в интерфейсе подключения

4. Восстановите конфигурацию интерфейса по умолчанию на узле h1.
5. Завершите работу mininet в интерактивном режиме, введя в интерфейсе mininet:

```
mininet> exit
```


2.9 Воспроизведение экспериментов

2.10 Предварительная подготовка

1. Обновите репозитории программного обеспечения на виртуальной машине:

```
sudo apt-get update
```

2. Установите пакет `gexie` — понадобится для просмотра файлов `png`:

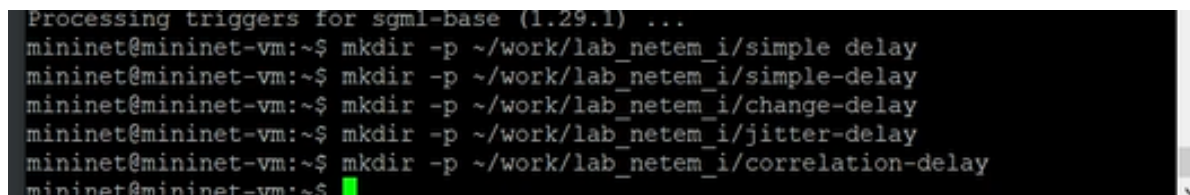
```
sudo apt install gexie
```

3. Для каждого воспроизводимого эксперимента `exname` создайте свой каталог, в котором будут размещаться файлы эксперимента:

```
mkdir -p ~/work/lab_netem_i/exname
```

Здесь `exname` может принимать значения `simple-delay`, `change-delay`, `jitter-delay`, `correlation-delay` и т.п.

4. Для каждого случая создайте скрипт для проведения эксперимента `lab_netem_i.py` и скрипт для визуализации результатов `ping_plot`.



```
Processing triggers for sgml-base (1.29.1) ...
mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_i/simple delay
mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_i/simple-delay
mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_i/change-delay
mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_i/jitter-delay
mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_i/correlation-delay
mininet@mininet-vm:~$
```

Рис. 2.8: Создаем репозитории

2.11 Добавление задержки для интерфейса, подключающегося к эмулируемой глобальной сети

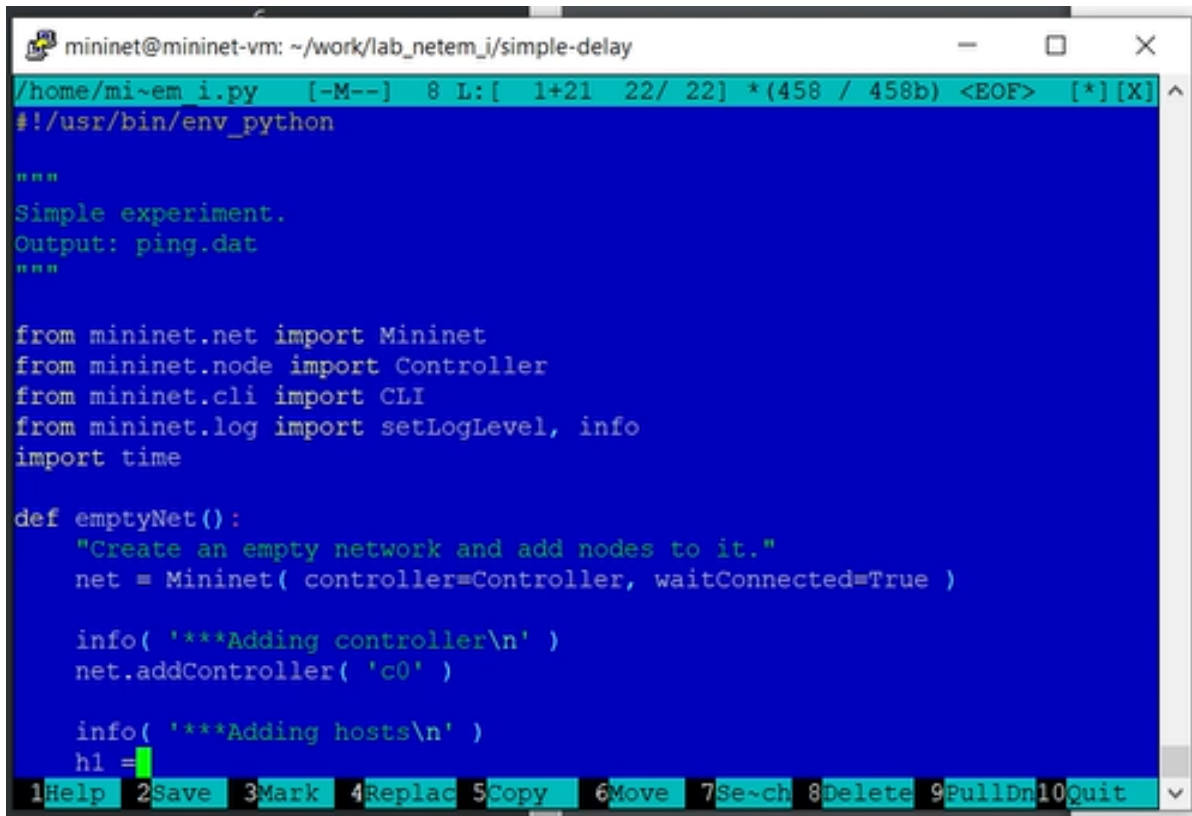
С помощью API Mininet воспроизведите эксперимент по добавлению задержки для интерфейса хоста, подключающегося к эмулируемой глобальной сети.

1. В виртуальной среде mininet в своём рабочем каталоге с проектами создайте каталог simple-delay и перейдите в него:

```
mkdir -p ~/work/lab_netem_i/simple-delay
```

```
cd ~/work/lab_netem_i/simple-delay
```

2. Создаёте скрипт для эксперимента lab_netem_i.py:



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

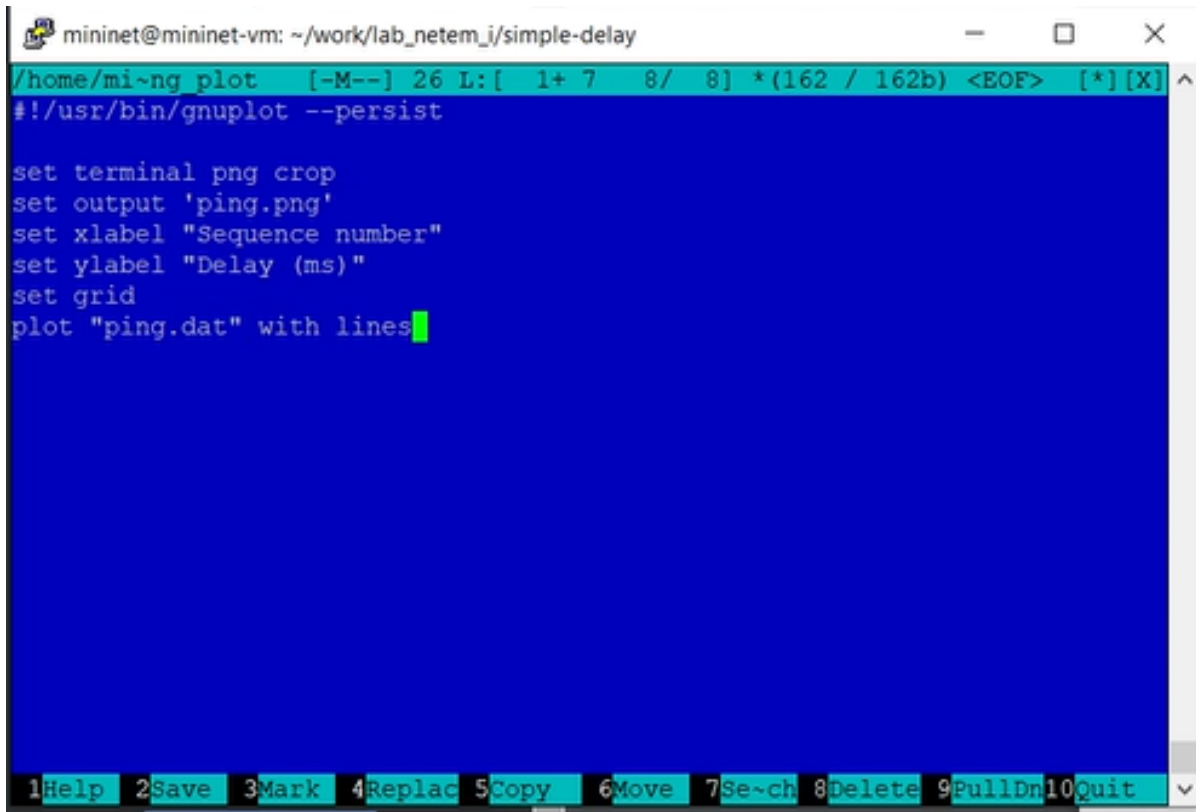
def emptyNet():
    "Create an empty network and add nodes to it."
    net = Mininet( controller=Controller, waitConnected=True )

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

    info( '***Adding hosts\n' )
    h1 =
```

Рис. 2.9: Создаем скрипт lab_netem_i.py

Создайте скрипт для визуализации ping_plot результатов эксперимента:



```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
/home/mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
#!/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. 2.10: Создаем скрипт для визуализации ping_plot

5. Задайте права доступа к файлу скрипта:

```
chmod +x ping_plot
```

6. Создайте Makefile для управления процессом проведения эксперимента

7. Выполните эксперимент:

```
make
```

8. Продемонстрируйте построенный в результате выполнения скриптов график.

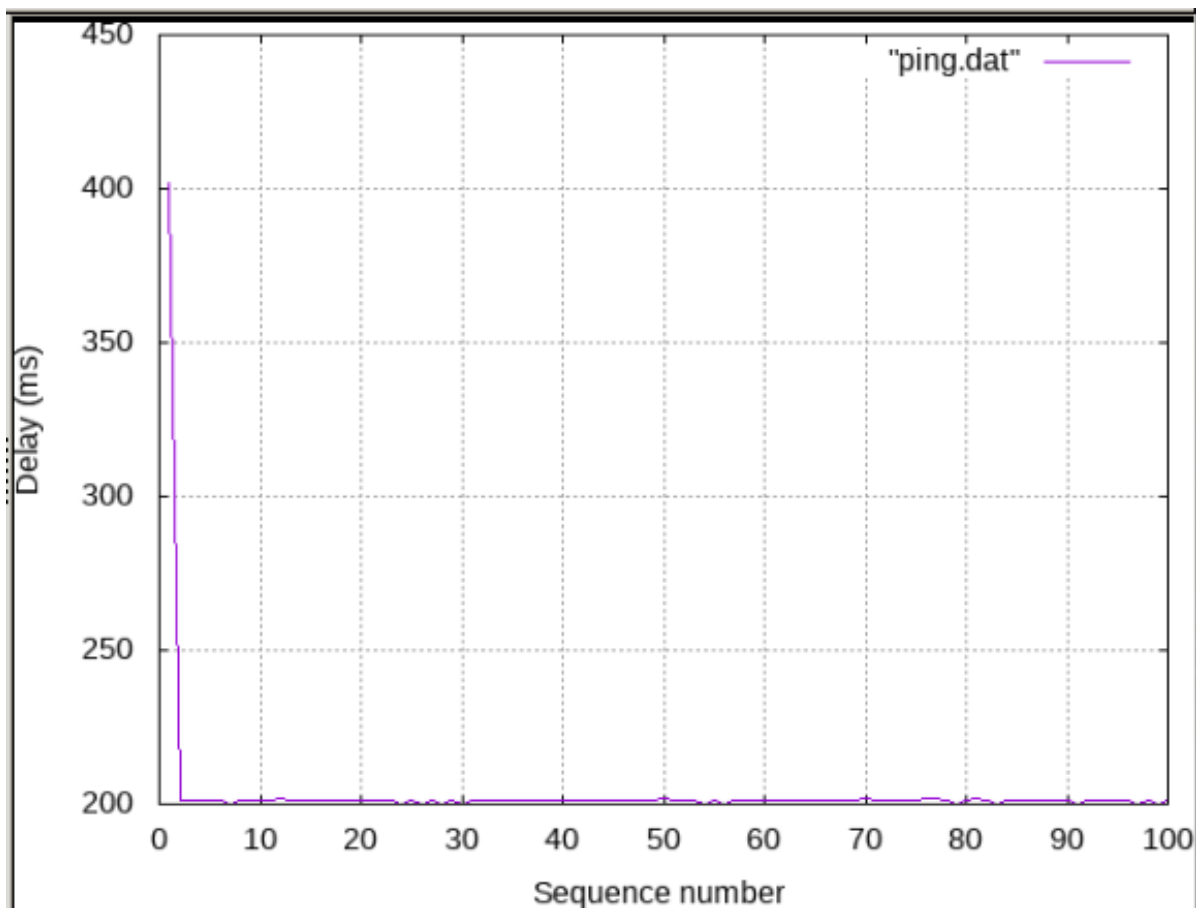


Рис. 2.11: График simple-delay

9. Из файла ping.dat удалите первую строку и заново постройте график:

```
make ping.png
```

10. Продемонстрируйте построенный в результате график.

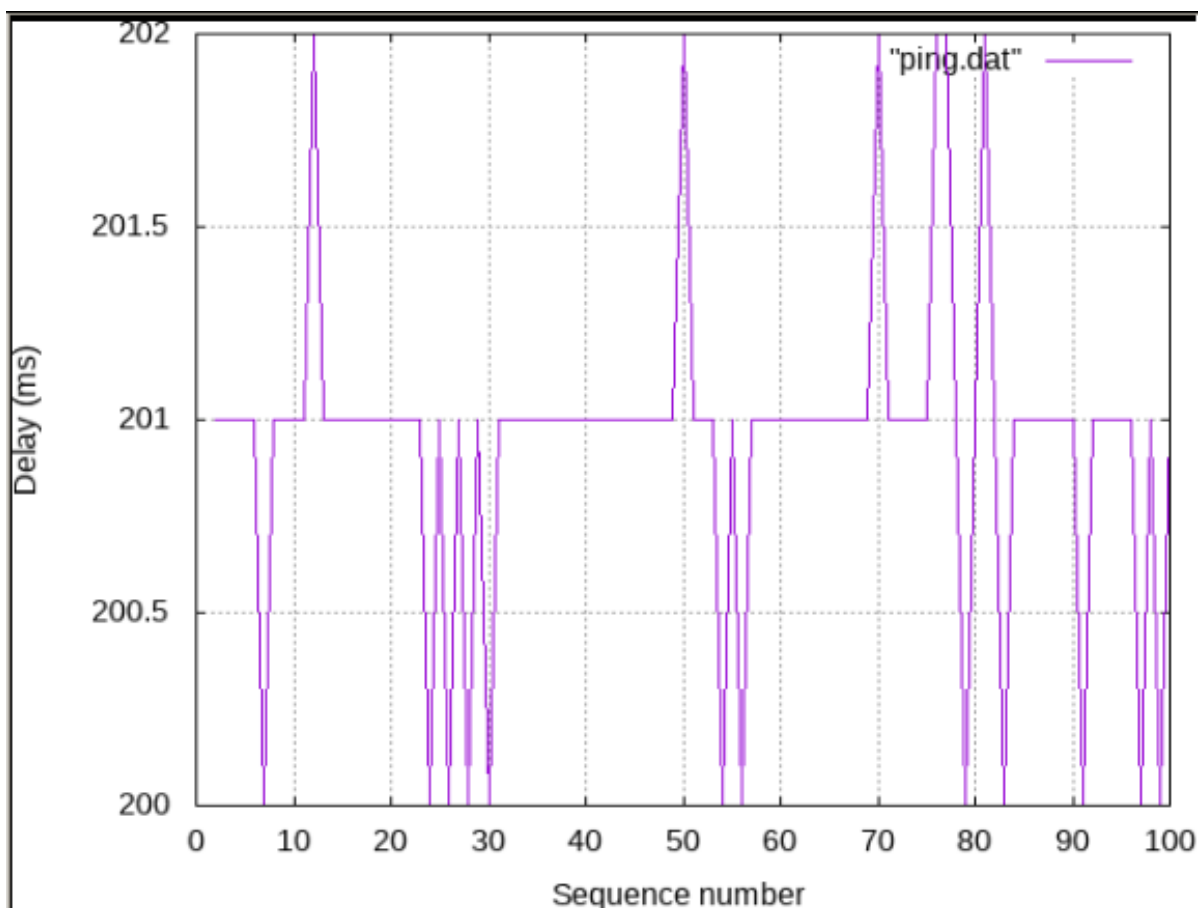


Рис. 2.12: График simple-delay без 1ой строчки в ping.dat

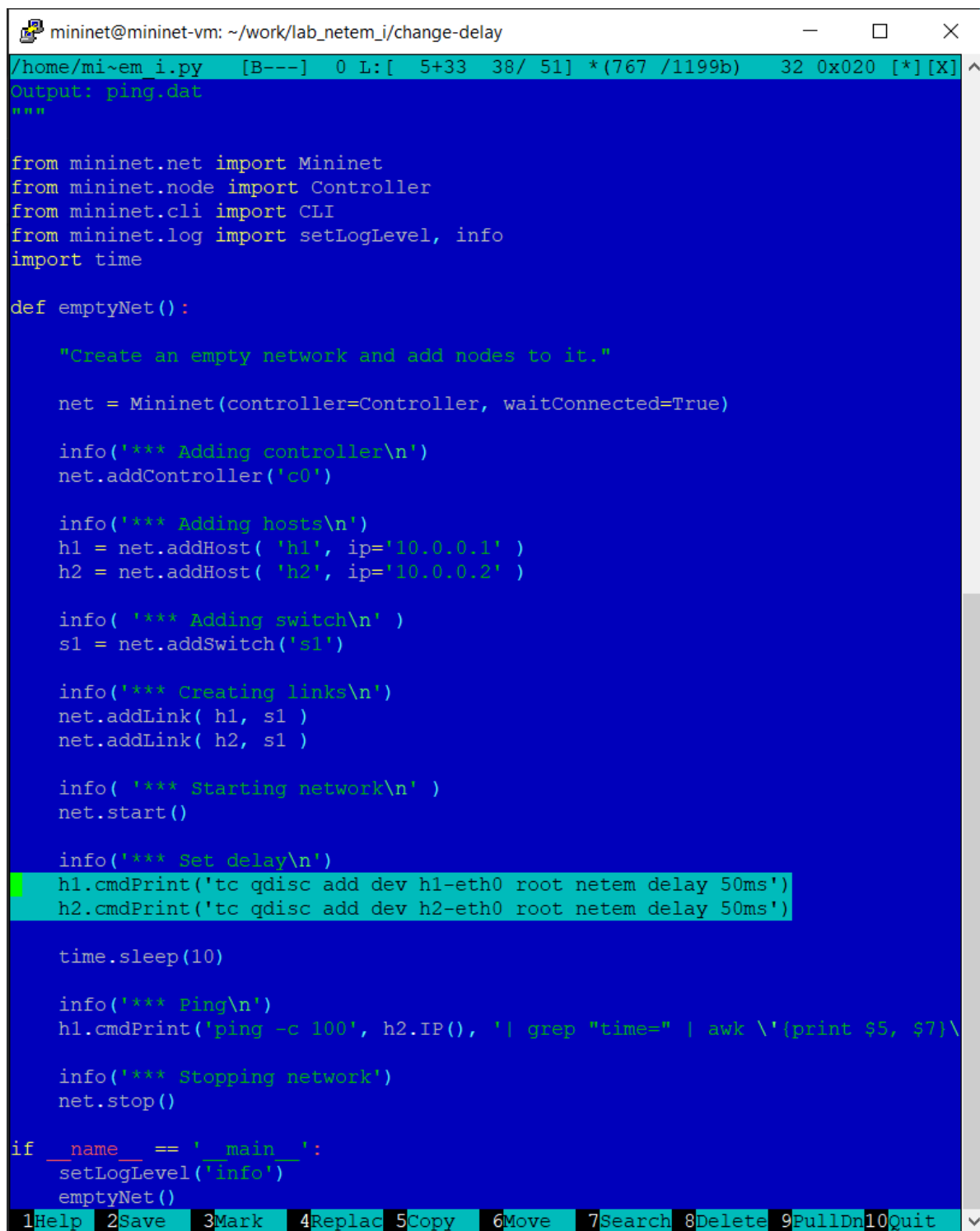
11. Разработайте скрипт для вычисления на основе данных файла ping.dat минимального, среднего, максимального и стандартного отклонения времени приёма-передачи. Добавьте правило запуска скрипта в Makefile. Продемонстрируйте работу скрипта с выводом значений на экран или в отдельный файл.

2.12 Задание для самостоятельной работы

Самостоятельно реализуйте воспроизводимые эксперименты по изменению задержки, джиттера, значения корреляции для джиттера и задержки, распределения времени задержки в эмулируемой глобальной сети. Постройте графики.

Вычислите минимальное, среднее, максимальное и стандартное отклонение времени приёма-передачи для каждого случая.

2.13 Эксперимент по изменению задержки



```
mininet@mininet-vm: ~/work/lab_netem_i/change-delay
/home/mininet@mininet-vm: [B---] 0 L:[ 5+33 38/ 51] *(767 /1199b) 32 0x020 [*][X] ^
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)

    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()

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

Рис. 2.13: Эксперимент по изменению задержки

```
mininet@mininet-vm: ~/work/lab_netem_i/change-delay
mininet@mininet-vm:~/work/lab_netem_i/change-delay$ make clean
rm -f *.dat *.png
mininet@mininet-vm:~/work/lab_netem_i/change-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, $7}\'' |
sed -e \'s/time=//g\' -e \'s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/change-delay$ make stats
python rtt.py
Min time: 100.0 ms
Avg time: 102.87 ms
Max time: 203.0 ms
Std dev: 10.62605759442325 ms
mininet@mininet-vm:~/work/lab_netem_i/change-delay$
```

Рис. 2.14: Эксперимент по изменению задержки

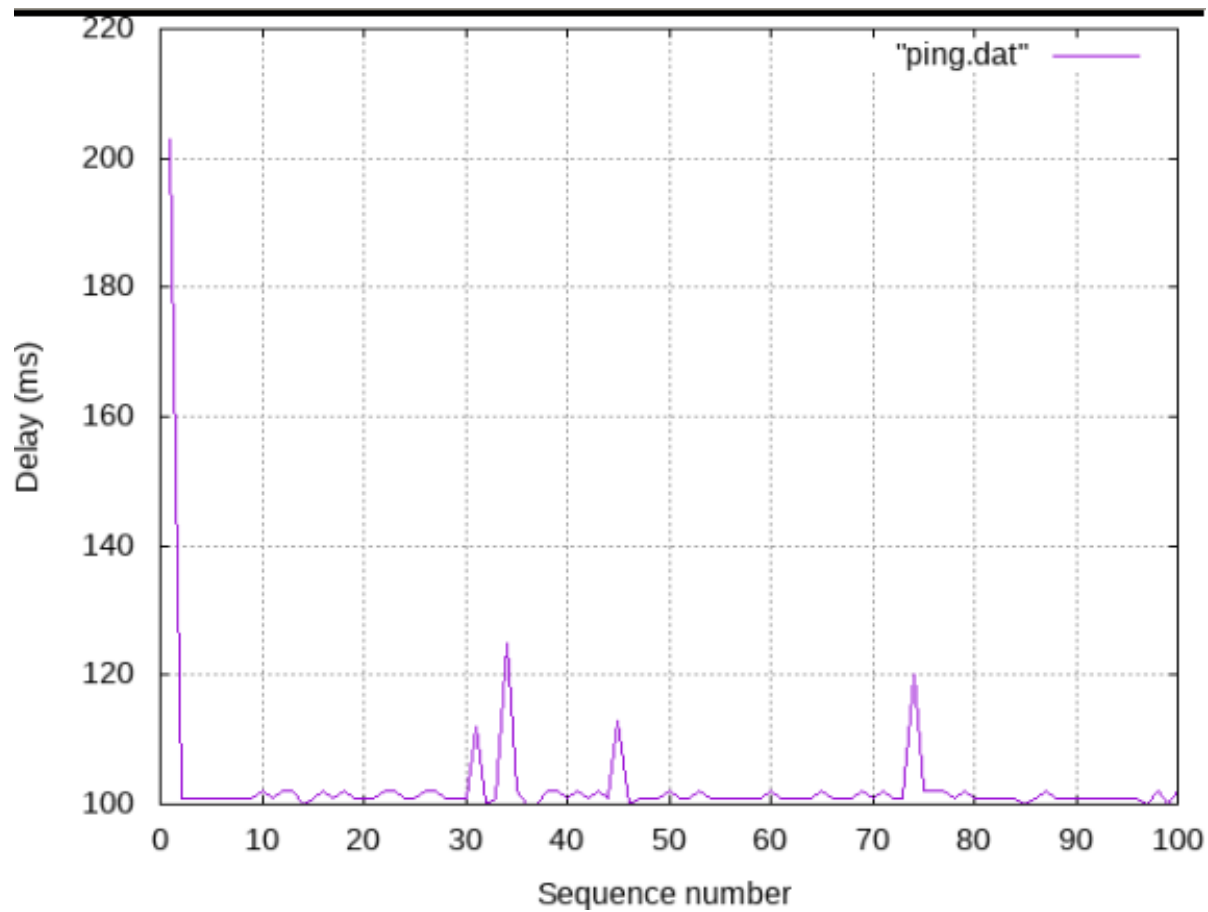
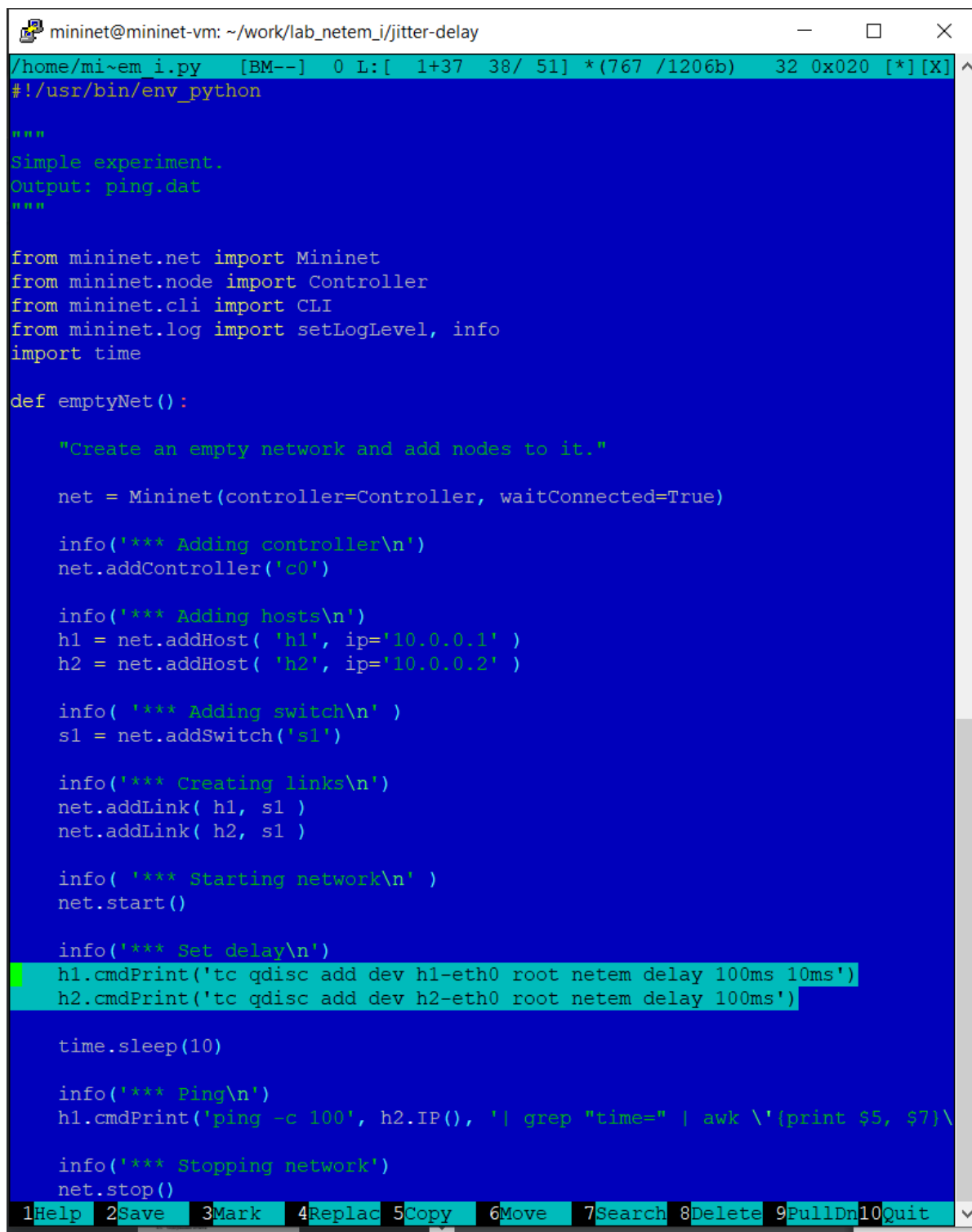


Рис. 2.15: Эксперимент по изменению задержки

2.14 Эксперимент по изменению джиттера



```
mininet@mininet-vm: ~/work/lab_netem_i/jitter-delay
/home/mi~em i.py [BM--] 0 L:[ 1+37 38/ 51] *(767 /1206b) 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)

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

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

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

Рис. 2.16: Эксперимент по изменению джиттера


```
mininet@mininet-vm: ~/work/lab_netem_i/jitter-delay
mininet@mininet-vm:~/work/lab_netem_i/jitter-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, $7}\'' |
sed -e 's/time=//g\' -e 's/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/jitter-delay$ make stats
python rtt.py
  File "rtt.py", line 16
    data = file.readlines()
                ^
TabError: inconsistent use of tabs and spaces in indentation
make: *** [Makefile:11: stats] Error 1
mininet@mininet-vm:~/work/lab_netem_i/jitter-delay$ mcedit rtt.py

mininet@mininet-vm:~/work/lab_netem_i/jitter-delay$ make stats
python rtt.py
Min time: 190.0 ms
Avg time: 203.13 ms
Max time: 392.0 ms
Std dev: 19.928198614024296 ms
mininet@mininet-vm:~/work/lab_netem_i/jitter-delay$
```

Рис. 2.17: Эксперимент по изменению джиттера

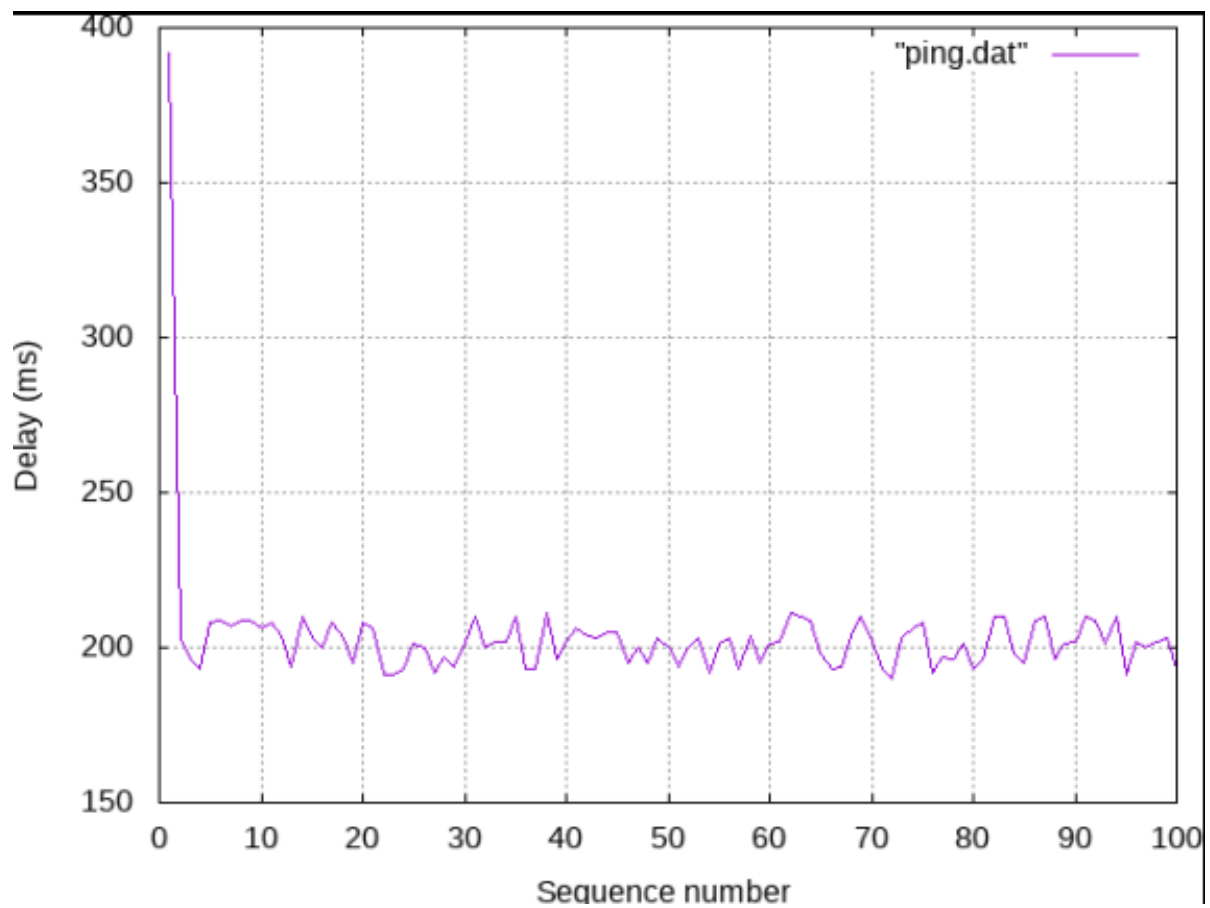
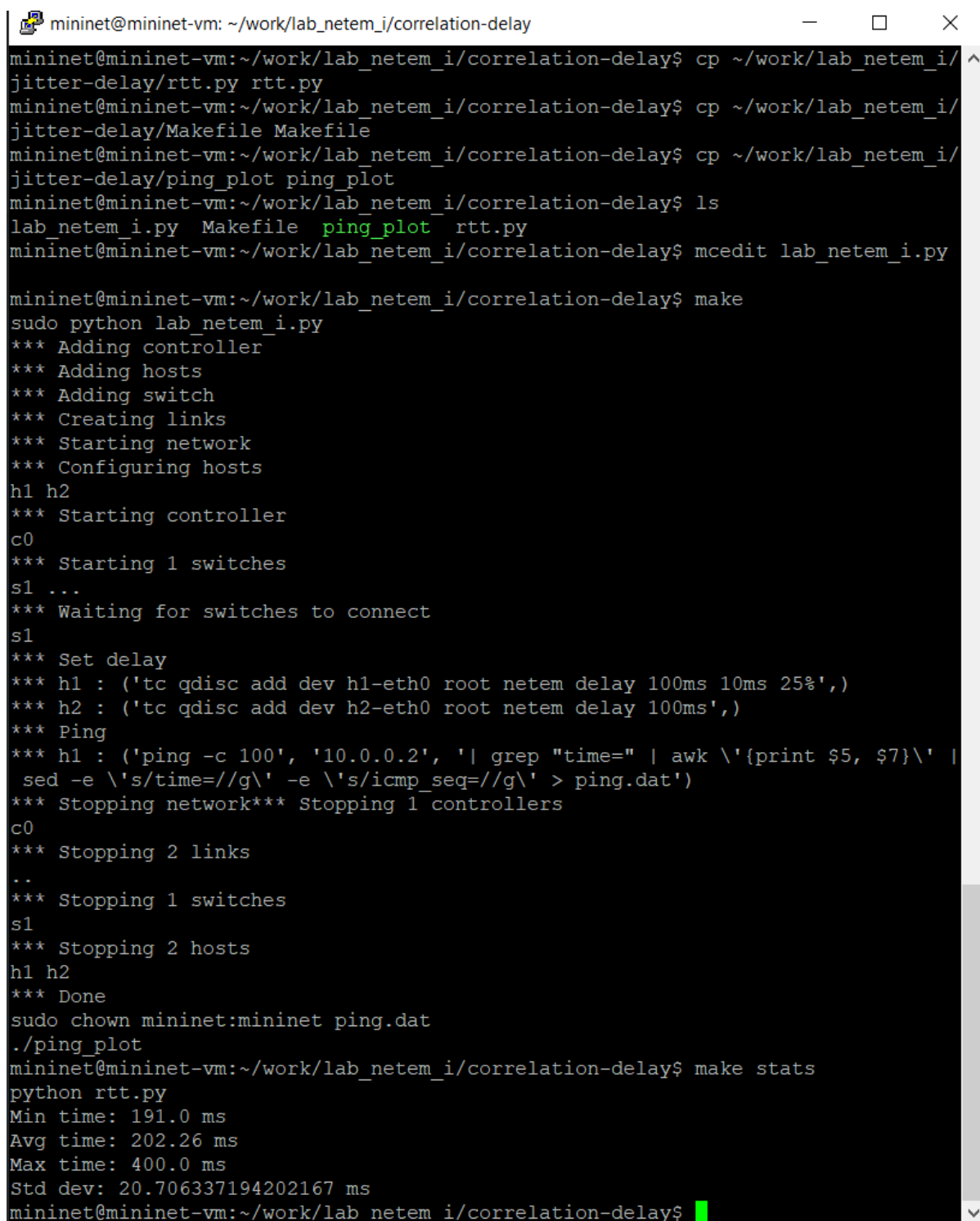


Рис. 2.18: Эксперимент по изменению джиттера

2.15 Эксперимент по изменению значения корреляции для джиттера и задержки



```
mininet@mininet-vm: ~/work/lab_netem_i/correlation-delay
mininet@mininet-vm:~/work/lab_netem_i/correlation-delay$ cp ~/work/lab_netem_i/jitter-delay/rtt.py rtt.py
mininet@mininet-vm:~/work/lab_netem_i/correlation-delay$ cp ~/work/lab_netem_i/jitter-delay/Makefile Makefile
mininet@mininet-vm:~/work/lab_netem_i/correlation-delay$ cp ~/work/lab_netem_i/jitter-delay/ping_plot ping_plot
mininet@mininet-vm:~/work/lab_netem_i/correlation-delay$ ls
lab_netem_i.py  Makefile  ping_plot  rtt.py
mininet@mininet-vm:~/work/lab_netem_i/correlation-delay$ mcedit lab_netem_i.py

mininet@mininet-vm:~/work/lab_netem_i/correlation-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, $7}\\' | sed -e \'/s/time=//g\' -e \'/s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/correlation-delay$ make stats
python rtt.py
Min time: 191.0 ms
Avg time: 202.26 ms
Max time: 400.0 ms
Std dev: 20.706337194202167 ms
mininet@mininet-vm:~/work/lab_netem_i/correlation-delay$
```

Рис. 2.19: Эксперимент по изменению значения корреляции для джиттера и задержки

```
mininet@mininet-vm: ~/work/lab_netem_i/correlation-delay
/home/mininet@mininet-vm:~/work/lab_netem_i/correlation-delay [B---] 0 L: [ 5+33 38/ 51] *(767 /1210b) 32 0x020 [*][X] ^
Output: ping.dat
"""

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

def emptyNet():

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

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

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

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

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

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

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

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

    time.sleep(10)

    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()

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

Рис. 2.20: Эксперимент по изменению значения корреляции для джиттера и задержки

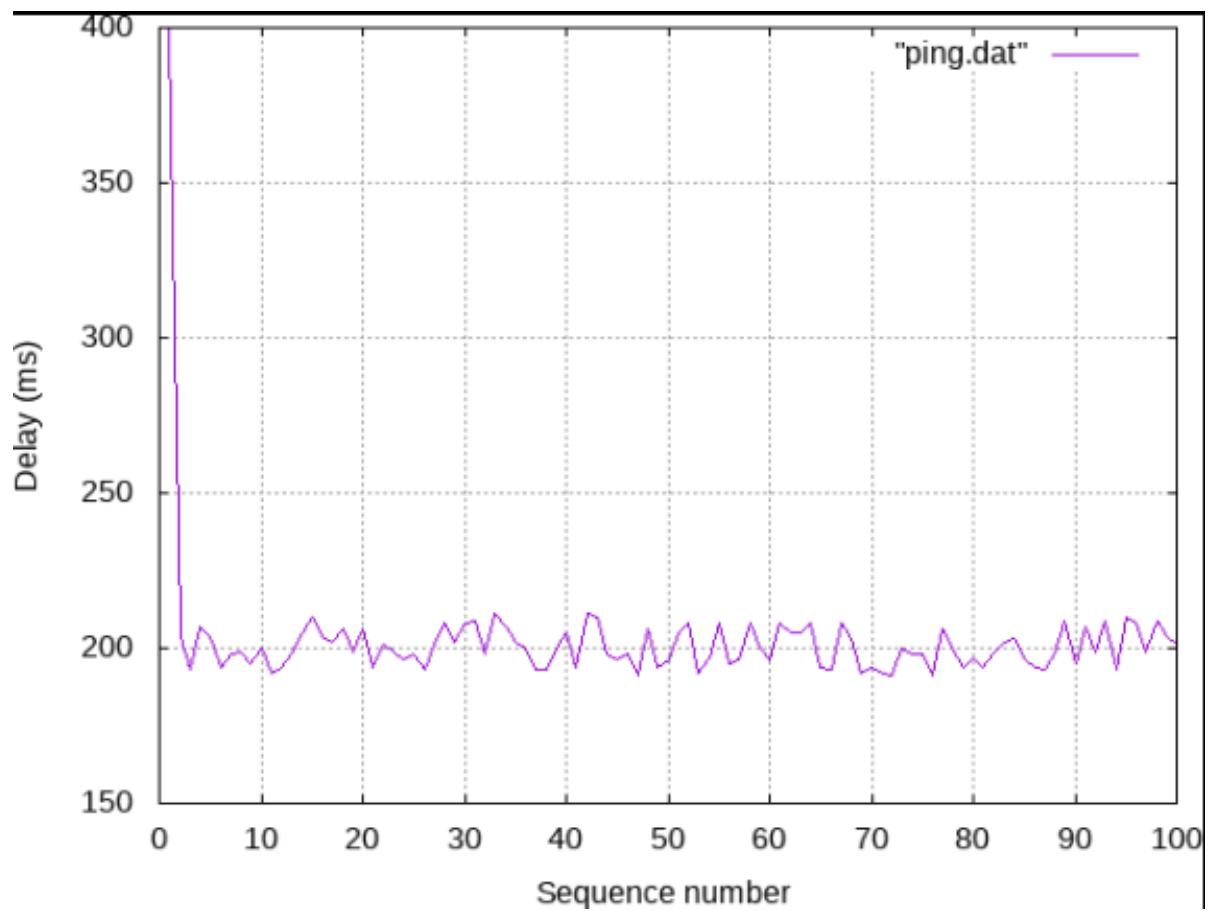


Рис. 2.21: Эксперимент по изменению значения корреляции для джиттера и задержки

2.16 Распределения времени задержки в эмулируемой глобальной сети

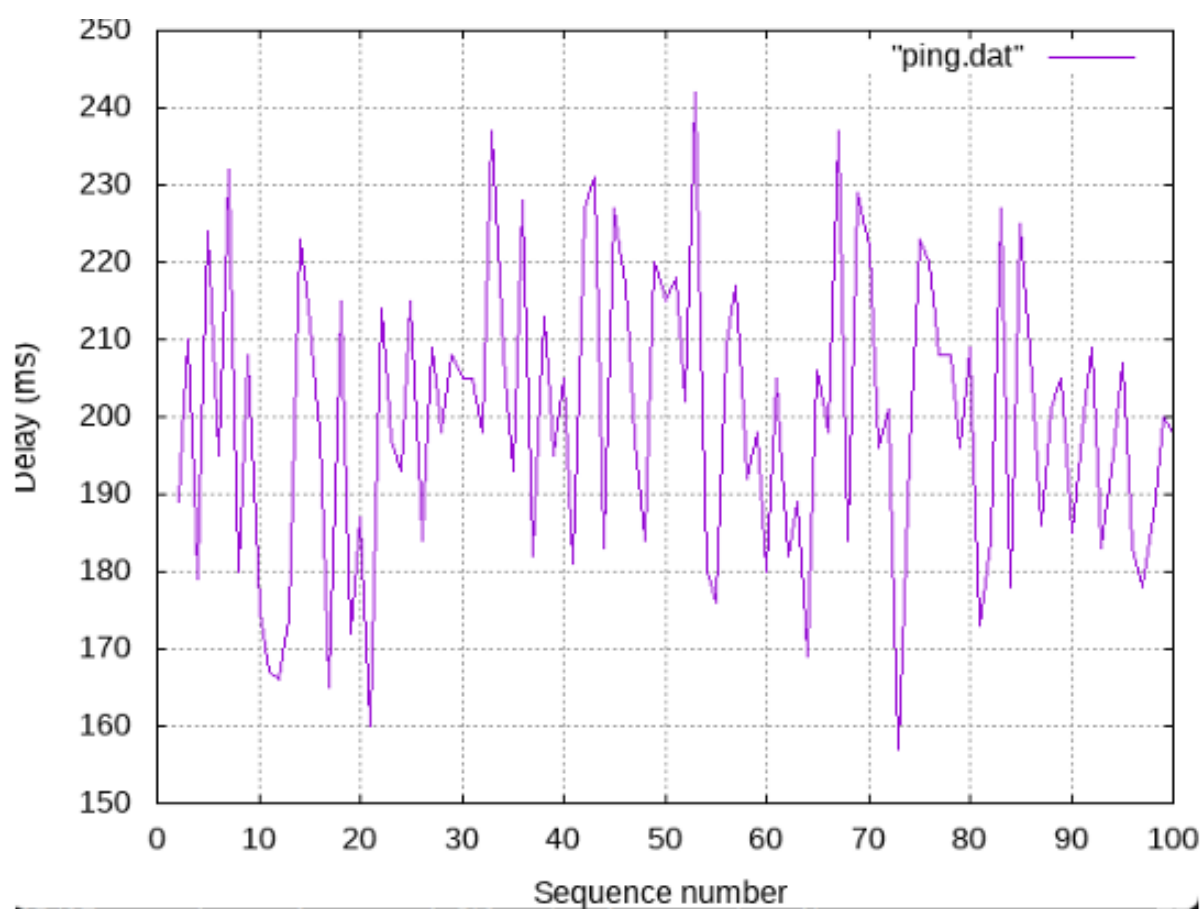


Рис. 2.22: Распределения времени задержки в эмулируемой глобальной сети

3 Выводы

Мы 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

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