

Отчет по лабораторной работе 4

Петрушов Дмитри1, 1032212287

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Вывод	17
	Список литературы	18

Список иллюстраций

2.1	Создание простейшей топологии	6
2.2	Отображение информации их сетевых интерфейсов и IP-адресов	7
2.3	Проверка подключения между хостами h1 и h2	8
2.4	Добавление задержки в 100 мс к выходному интерфейсу на хосте h1	8
2.5	Проверка	9
2.6	Добавление задержки в 100 мс к выходному интерфейсу на хосте h2 и Проверка	9
2.7	Изменение задержки со 100 мс до 50 мс	10
2.8	Проверка	10
2.9	Восстановление конфигураций по умолчанию	10
2.10	Добавление на узле h1 задержки в 100 мс со случайным отклонением 10 мс и проверка	11
2.11	добавление корреляции	11
2.12	Проверка	12
2.13	Настройка нормального распределения задержки на узле h1 в эмулируемой сети	12
2.14	Проверка	12
2.15	Создание нового каталога	13
2.16	а	14
2.17	Просмотр графика	15
2.18	Повторное построение графика	16

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 и Зададим простейшую топологию, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8 (рис. [2.1]):

```
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 2112a7885808e22f78c1dedb5fa0f837
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth add mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 2112a7885808e22f78c1dedb5fa0f837
root@mininet-vm:~# logout
mininet@mininet-vm:~$ sudo mn --topo=single,2 -x
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Running terms on localhost:10.0
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

Рис. 2.1: 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 даль-

нейшем при работе с NETEM и командой tc будут использоваться интерфейсы h1-eth0 и h2-eth0 и Проверим подключение между хостами h1 и h2 с помощью команды ping с параметром -c 6 (рис. [2.2]):

```
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 12:94:45:1e:fe:e0 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 1242 bytes 574692 (574.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1242 bytes 574692 (574.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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=5.42 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.403 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.079 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.080 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.050 ms
-----
2-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 4e:ed:41:91:da:44 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 1294 bytes 577116 (577.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1294 bytes 577116 (577.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

Рис. 2.2: Отображение информации их сетевых интерфейсов и IP-адресов

Проверим подключение между хостами h1 и h2 с помощью команды ping с параметром -c 6 (рис. [2.3]):

```
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=5.42 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.403 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.079 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.080 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.050 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.102 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5075ms
rtt min/avg/max/mdev = 0.050/1.022/5.418/1.969 ms
```

Рис. 2.3: Проверка подключения между хостами h1 и h2

На хосте h1 добавим задержку в 100 мс к выходному интерфейсу (рис. [2.4]):

```
10.0.0.2 ping statistics
6 packets transmitted, 6 received, 0% packet loss, time 5075ms
rtt min/avg/max/mdev = 0.050/1.022/5.418/1.969 ms
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms
root@mininet-vm:/home/mininet#
```

Рис. 2.4: 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. [2.5]):


```

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=101 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=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 5007ms
rtt min/avg/max/mdev = 100.183/100.549/101.315/0.393 ms
root@mininet-vm:/home/mininet#

```

Рис. 2.5: Проверка

Для эмуляции глобальной сети с двунаправленной задержкой необходимо к соответствующему интерфейсу на хосте h2 также добавить задержку в 100 миллисекунд

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. [2.6]):

```

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

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5012ms
rtt min/avg/max/mdev = 200.196/200.599/201.276/0.334 ms
root@mininet-vm:/home/mininet#

```

Рис. 2.6: Добавление задержки в 100 мс к выходному интерфейсу на хосте h2 и Проверка

Изменим задержку со 100 мс до 50 мс для отправителя h1 и для получателя h2

(рис. [2.7]):

```
root@mininet-vm:/home/mininet# sudo tc qdisc change dev h1-eth0 root netem delay 50ms
root@mininet-vm:/home/mininet# 
root@mininet-vm:/home/mininet# sudo tc qdisc change dev h2-eth0 root netem delay 50ms
root@mininet-vm:/home/mininet#
```

Рис. 2.7: 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. [2.8]):

```
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=101 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=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 5008ms
rtt min/avg/max/mdev = 100.255/101.216/103.815/1.231 ms
```

Рис. 2.8: Проверка

Восстановим конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса (рис. [2.9]):

```
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
root@mininet-vm:/home/mininet# 
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h2-eth0 root netem
root@mininet-vm:/home/mininet#
```

Рис. 2.9: Восстановление конфигураций по умолчанию

Добавим на узле h1 задержку в 100 мс со случайным отклонением 10 мс. Проверим, что соединение от хоста h1 к хосту h2 имеет задержку 100 мс со случайным отклонением ± 10 мс, используя в терминале хоста h1 команду ping с параметром -c 6 (рис. [2.10]):

```
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=114 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=97.3 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=98.5 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=96.1 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=93.1 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 93.103/99.885/114.116/6.727 ms
root@mininet-vm:/home/mininet#
```

Рис. 2.10: Добавление на узле h1 задержки в 100 мс со случайным отклонением 10 мс и проверка

Добавим на интерфейсе хоста h1 задержку в 100 мс с вариацией ± 10 мс и значением корреляции в 25%. Убедимся, что все пакеты, покидающие устройство h1 на интерфейсе h1-eth0, будут иметь время задержки 100 мс со случайным отклонением ± 10 мс, при этом время передачи следующего пакета зависит от предыдущего значения на 25%. Используем для этого в терминале хоста h1 команду ping с параметром -c 20 (рис. [2.11]):

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms 10ms 25%
root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 20
```

Рис. 2.11: добавление корреляции

Проверим (рис. [2.12]):

```
--- 10.0.0.2 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19028ms
rtt min/avg/max/mdev = 91.186/101.347/108.803/4.813 ms
root@mininet-vm:/home/mininet#
```

Рис. 2.12: Проверка

Зададим нормальное распределение задержки на узле h1 в эмулируемой сети (рис. [2.13]):

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

Рис. 2.13: 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. [2.14]):

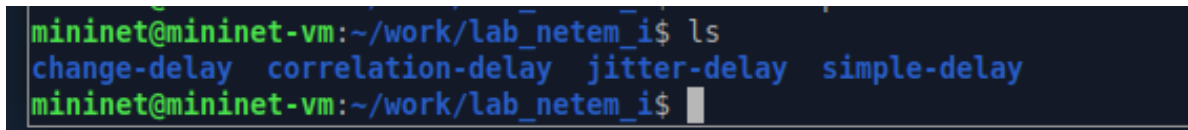
```
10 packets transmitted, 10 received, 0% packet loss, time 9015ms
rtt min/avg/max/mdev = 67.913/103.996/137.376/22.628 ms
root@mininet-vm:/home/mininet#
```

Рис. 2.14: Проверка

Восстановим конфигурацию интерфейса по умолчанию на узле h1
Завершим работу mininet в интерактивном режиме
Обновим репозитории программного обеспечения на виртуальной машине

Установим пакет `ghee` для просмотра файлов `png`

Для каждого воспроизводимого эксперимента `exname` создадим свой каталог, в котором будут размещаться файлы эксперимента (рис. [2.15]):

A terminal window with a dark background. The prompt is 'mininet@mininet-vm:~/work/lab_netem_i\$'. The command 'ls' has been entered, and the output shows four directories: 'change-delay', 'correlation-delay', 'jitter-delay', and 'simple-delay'.

```
mininet@mininet-vm:~/work/lab_netem_i$ ls
change-delay correlation-delay jitter-delay simple-delay
mininet@mininet-vm:~/work/lab_netem_i$
```

Рис. 2.15: Создание нового каталога

В виртуальной среде `mininet` в своём рабочем каталоге с проектами создадим каталог `simple-delay` и перейдём в него. Создадим скрипт для эксперимента `lab_netem_i.py`. Создадим файл `ping_plot`. Затем создадим скрипт для визуализации `ping_plot` результатов эксперимента. Зададим права доступа к файлу скрипта. Создадим файла `Makefile`. нутри файла `Makefile` поместим скрипт для управления процессом проведения эксперимента (рис. [2.16]):

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

Рис. 2.16: а

Выполним эксперимент и Просмотрим построенный в результате выполнения скриптов график (рис. [2.17]):

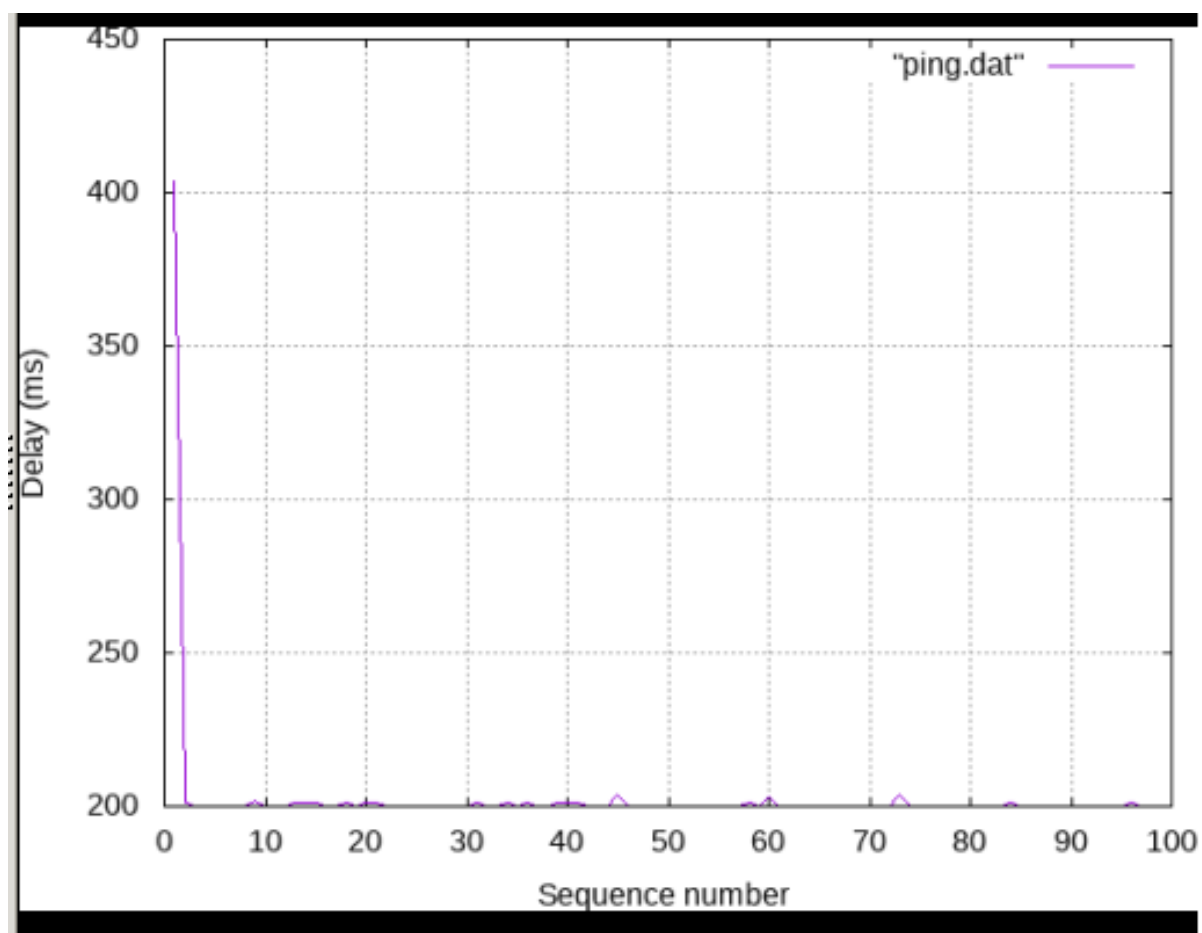


Рис. 2.17: Просмотр графика

Из файла ping.dat удалим первую строку и заново построим график (рис. [2.18]):

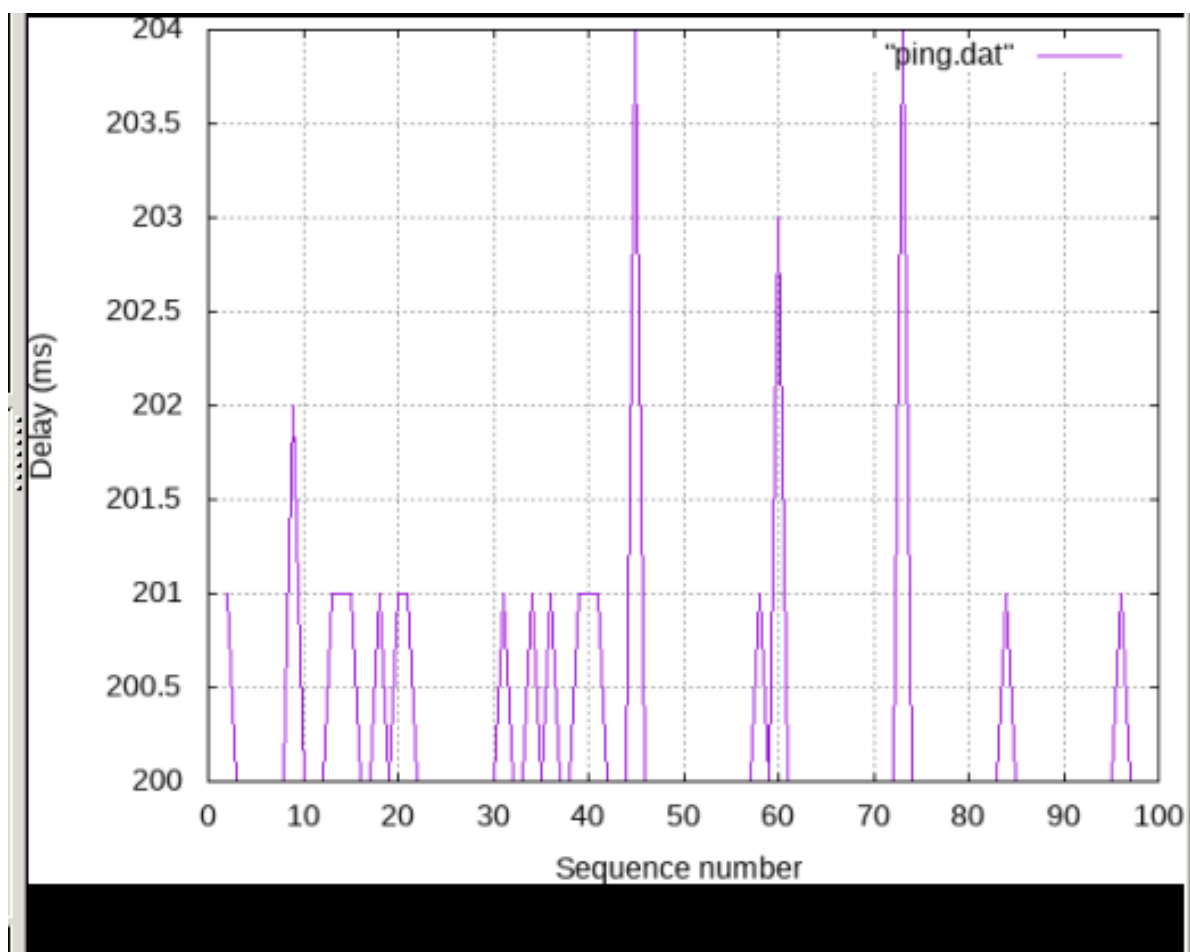


Рис. 2.18: Повторное построение графика

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. 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.

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.

Список литературы