Лабораторная работа № 4

Эмуляция и измерение задержек в глобальных сетях

Оразгелдиев Язгелди

2025-10-25

Содержание

Список иллюстраций

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 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

1. Запустил виртуальную среду с mininet и из основной ОС подключилась к виртуальной машине.

2. В виртуальной машине mininet исправил права запуска X-соединения. Скопировал значение куки своего пользователя mininet в файл для пользователя root.

Рисунок 3.1: *Исправление права запуска X-соединения*

3. Задал простейшую топологию, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8.

```
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>
```

Рисунок 3.2: *Создание простейшей топологии*

4. На хостах h1 и h2 ввел команду ifconfig, чтобы отобразить информацию, относящуюся к их сетевым интерфейсам и назначенным им IP-адресам.

```
"host: h1" (на 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 7e:04:c3:9b:62:9f  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 860  bytes 311956 (311.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 860  bytes 311956 (311.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Рисунок 3.3: *Команда ifconfig на хосте h1*

```
"host: h2" (на 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 de:e7:91:c3:cb:54  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 1027  bytes 325680 (325.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1027  bytes 325680 (325.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Рисунок 3.4: *Команда ifconfig на хосте h2*

5. Проверил подключение между хостами h1 и h2 с помощью команды ping с параметром -c 6. Минимальное RTT: 0,03; Среднее RTT: 0,548; Максимальное RTT: 1,3; Стандартное отклонение: 0,464.

```
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=1.30 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.142 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.030 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.075 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.072 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.035 ms
```

Рисунок 3.5: *Команда ping*

```
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=0.813 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.052 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.042 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.082 ms
```

Рисунок 3.6: *Команда ping*

6. На хосте h1 добавил задержку в 100 мс к выходному интерфейсу.

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

Рисунок 3.7: *Добавление задержки на хосте h1*

7. Проверил, что соединение от хоста h1 к хосту h2 имеет задержку 100 мс, используя команду ping с параметром -c 6 с хоста h1. Минимальное RTT: 100; Среднее RTT: 100,8; Максимальное RTT: 101; Стандартное отклонение: 0,374.

```
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=101 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=101 ms
```

Рисунок 3.8: *Команда ping*

8. Для эмуляции глобальной сети с двунаправленной задержкой к соответствующему интерфейсу на хосте h2 также добавил задержку в 100 миллисекунд.

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

Рисунок 3.9: *Добавление задержки на хосте h2*

9. Проверил, 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. Минимальное RTT: 201; Среднее RTT: 201; Максимальное RTT: 201; Стандартное отклонение: 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=201 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=201 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=201 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=201 ms
```

Рисунок 3.10: *Команда ping*

10. Изменил задержку со 100 мс до 50 мс для отправителя h1 и для получателя h2.

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

Рисунок 3.11: *Изменение задержки*

11. Проверил, что соединение от хоста h1 к хосту h2 имеет задержку 100 мс, используя команду ping с параметром -c 6 с терминала хоста h1. Минимальное RTT: 100; Среднее RTT: 101; Максимальное RTT: 102; Стандартное отклонение: 1.

```
root@mininet-vm:/home/mininet# ping -c 6 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=102 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=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=102 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
```

Рисунок 3.12: *Команда ping*

12. Восстановил конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса. Для отправителя h1 и для получателя h2.

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

Рисунок 3.13: *Удаление правил*

13. Проверил, что соединение между хостом h1 и хостом h2 не имеет явно установленной задержки, используя команду ping с параметром -c 6 с терминала хоста h1. Минимальное RTT: 0,033; Среднее RTT: 0,5118; Максимальное RTT: 1,11; Стандартное отклонение: 0,487.

```
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=1.11 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.971 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.114 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.033 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.034 ms
```

Рисунок 3.14: *Команда ping*

14. Добавил на узле h1 задержку в 100 мс со случайным отклонением 10 мс.

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

Рисунок 3.15: *Добавление задержки со случайным отклонением*

15. Проверил, 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. Минимальное RTT: 94; Среднее RTT: 102.46; Максимальное RTT: 111; Стандартное отклонение: 6.6.

```
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=111 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=94.0 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=95.1 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=109 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=99.7 ms
```

Рисунок 3.16: *Команда ping*

16. Восстановил конфигурацию интерфейса по умолчанию на узле h1.

```
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
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=0.239 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.035 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.058 ms
^C
```

Рисунок 3.17: *Удаление правил*

17. Добавил на интерфейсе хоста h1 задержку в 100 мс с вариацией ±10 мс и значением корреляции 25%.

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

Рисунок 3.18: *Добавление задержки со случайным отклонением и корреляцией*

18. Убедилась, 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%. Использовала для этого в терминале хоста h1 команду ping с параметром -c 6. Минимальное RTT: 91,5; Среднее RTT: 97,3; Максимальное RTT: 109; Стандартное отклонение: 6,43. Восстановил конфигурацию интерфейса по умолчанию на узле h1.

```
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=93.5 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=92.9 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=91.5 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=103 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=109 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=93.9 ms
```

Рисунок 3.19: *Команда ping*

19. Задал нормальное распределение задержки на узле h1 в эмулируемой сети.

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms 20ms dis
tribution normal
root@mininet-vm:/home/mininet#
```

Рисунок 3.20: *Добавление нормального распределения*

20. Убедился, что все пакеты, покидающие хост h1 на интерфейсе h1-eth0, будут иметь время задержки, которое распределено в диапазоне 100 мс ±20 мс. Использовал для этого команду ping на терминале хоста h1 с параметром -c 6. Минимальное RTT: 75,8; Среднее RTT: 106,3; Максимальное RTT: 135; Стандартное отклонение: 24,36. Завершила работу mininet в интерактивном режиме.

```
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=79.7 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=93.3 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=135 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=121 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=75.8 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=133 ms
```

Рисунок 3.21: *Команда ping*

21. Обновил репозитории программного обеспечения на виртуальной машине.

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ 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]
Fetched 383 kB in 1s (285 kB/s)
Reading package lists... Done
```

Рисунок 3.22: *Обновление репозиториев программного обеспечения*

22. Установил пакет geeqie — понадобится для просмотра файлов png.

```
mininet@mininet-vm:~$ sudo apt install geeqie
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  acl apg apport apport-symptoms aptdaemon aptdaemon-data avahi-daemon
  avahi-utils bluez bolt cheese-common colord colord-data cracklib-runtime
  cups-bsd cups-client cups-common cups-pk-helper dbus dbus-x11 dconf-cli
```

Рисунок 3.23: *Установка пакета geeqie*

23. Для каждого воспроизводимого эксперимента expname создал свой каталог, в котором будут размещаться файлы эксперимента.

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

Рисунок 3.24: *Создание каталога*

24. В виртуальной среде mininet в своём рабочем каталоге с проектами создал каталог simple-delay и перешла в него.

```
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$
```

Рисунок 3.25: *Создание подкаталога*

25. Создал скрипт для эксперимента lab_netem_i.py.

```
 GNU nano 4.8                                    /home/mininet/work/lab_netem_i/simple-delay/lab_netem_
#!/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' )
        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 \'s/icmp_seq=//g\' > ping.dat' )

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

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

Рисунок 3.26: *Создание скрипта lab_netem_i.py*

26. Создал скрипт для визуализации ping_plot результатов эксперимента.

```
        /home/mininet/work/lab_netem_i/simple-delay/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
```

Рисунок 3.27: *Создание скрипта ping_plot*

27. Задал права доступа к файлу скрипта.

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

Рисунок 3.28: *Изменение прав доступа*

28. Создал Makefile для управления процессом проведения эксперимента.

```
  GNU nano 4.8 /home/mininet/work/lab_netem_i/simple-delay/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

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

Рисунок 3.29: *Создание Makefile*

29. Выполнил эксперимент.

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

Рисунок 3.30: *Выполнение эксперимента*

30. Продемонстрировал построенный в результате выполнения скриптов график.

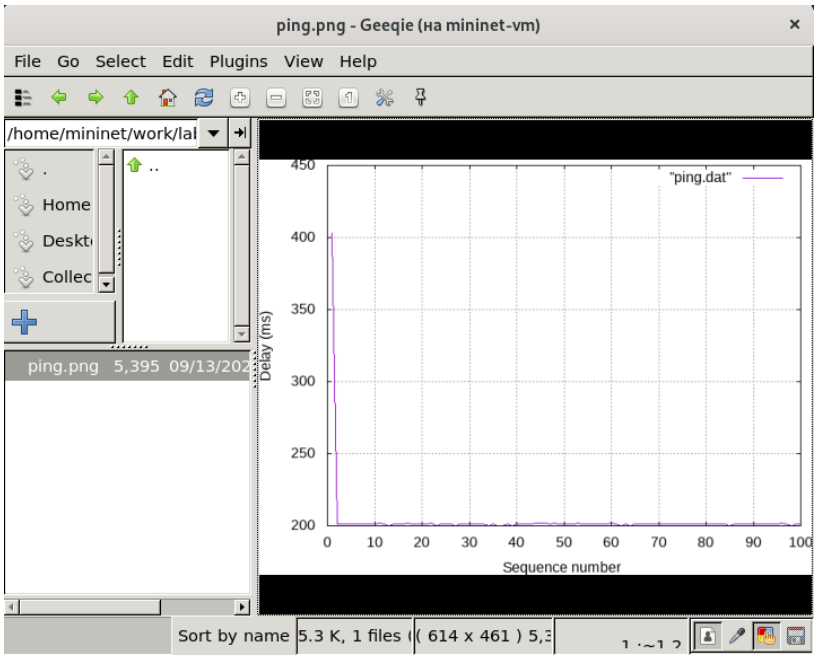

Рисунок 3.31: *График 1.1*

31. Из файла ping.dat удалил первую строку и заново постройте график.

Рисунок 3.32: *Удаление строки*

32. Продемонстрировал построенный в результате график.

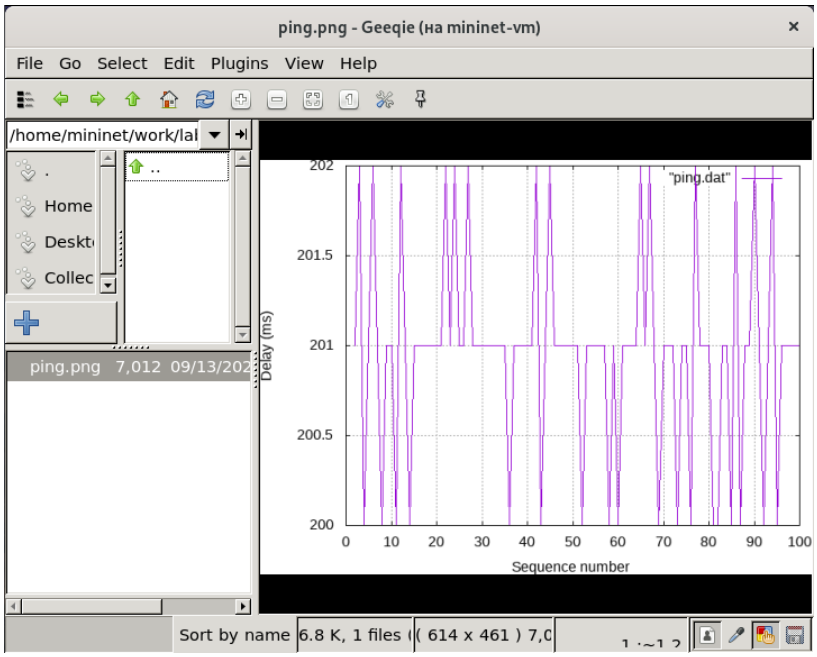

Рисунок 3.33: *График 1.2*

33. Разработал скрипт для вычисления на основе данных файла ping.dat минимального, среднего, максимального и стандартного отклонения времени приёма-передачи. Добавил правило запуска скрипта в Makefile. Продемонстрировал работу скрипта с выводом значений на экран. Очистил каталог от результатов проведения экспериментов.

```
 GNU nano 4.8          /home/mininet/work/lab_netem_i/simple-delay/rtt.py
with open('ping.dat', 'r') as f:
        s=[]
        for line in f.readlines():
                if '\n' in line:
                        line.replace('\n', "")
                        s.append([int(j) for j in (line.split(" "))])
        s = [j[1] for j in s]
        #print(s)
        std=(sum([(i-(sum(s)/len(s)))**2 for i in s])/(len(s)-1))**0.5
        print(f"min: {min(s)} \n max: {max(s)} \n avg: {sum(s)/len(s)} \n std: {std}")
```

Рисунок 3.34: *Скрипт для вычисления данных*

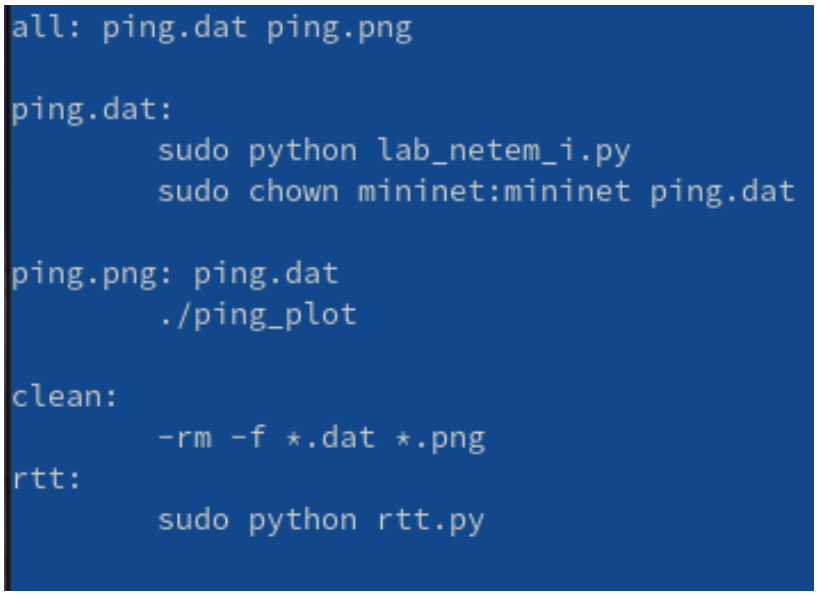

Рисунок 3.35: *Изменение Makefile*

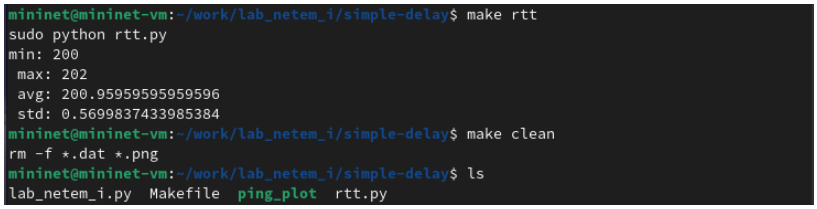

Рисунок 3.36: *Результат работы скрипта*

34. Самостоятельно реализовал воспроизводимые эксперименты по изменению задержки в эмулируемой глобальной сети. Построил графики. Вычислил минимальное, среднее, максимальное и стандартное отклонение времени приёма-передачи для каждого случая.

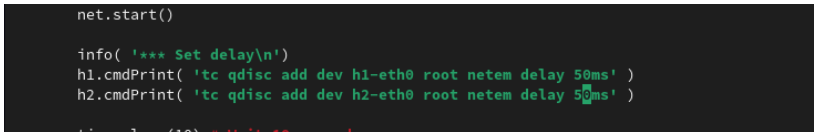

Рисунок 3.37: *Изменение файла lab_netem_i.py*

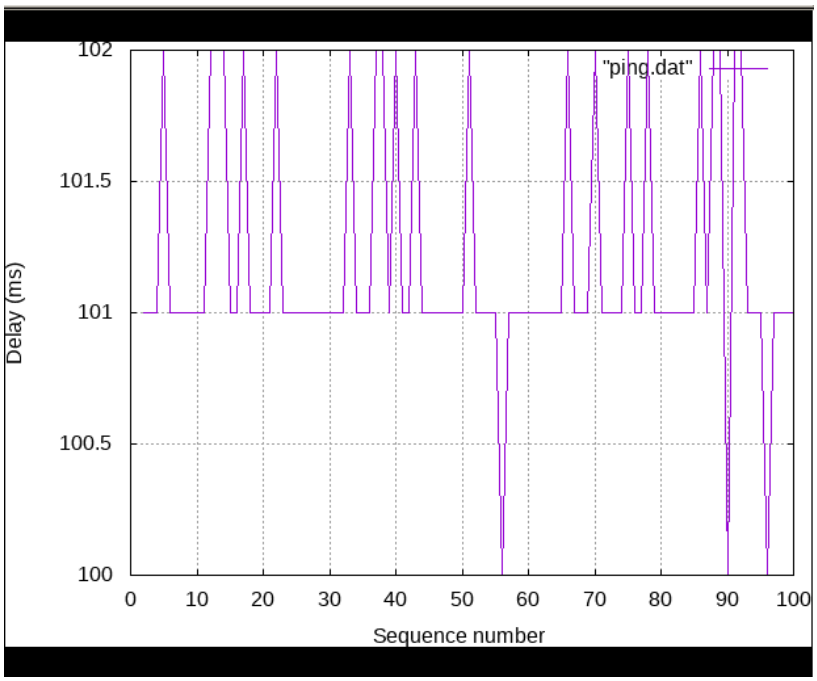

Рисунок 3.38: *График 2*

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make rtt
sudo python rtt.py
min: 100
 max: 102
 avg: 101.18181818181819
 std: 0.45989431713313733
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$
```

Рисунок 3.39: *Вычисленные значения*

35. Самостоятельно реализовал воспроизводимые эксперименты по изменению джиттера в эмулируемой глобальной сети. Построил графики. Вычислил минимальное, среднее, максимальное и стандартное отклонение времени приёма-передачи для каждого случая.

```
    info( '*** Set delay\n')
    h1.cmdPrint( 'tc qdisc add dev h1-eth0 root netem delay 50ms 10ms' )
    h2.cmdPrint( 'tc qdisc add dev h2-eth0 root netem delay 50ms 10ms' )
```

Рисунок 3.40: *Изменение файла lab_netem_i.py*

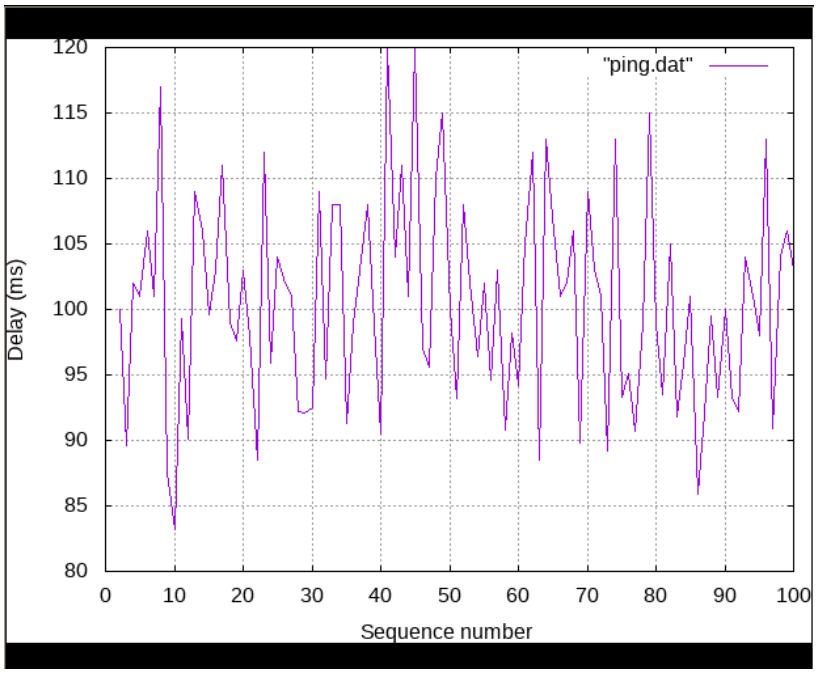

Рисунок 3.41: *График 3*

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make rtt
sudo python rtt.py
min: 83.2
 max: 120.0
 avg: 100.44343434343432
 std: 7.890360453422326
```

Рисунок 3.42: *Вычисленные значения*

36. Самостоятельно реализовал воспроизводимые эксперименты по изменению значения корреляции для джиттера и задержки в эмулируемой глобальной сети. Построил графики. Вычислил минимальное, среднее, максимальное и стандартное отклонение времени приёма-передачи для каждого случая.

```
        info( '*** Set delay\n')
        h1.cmdPrint( 'tc qdisc add dev h1-eth0 root netem delay 50ms 10ms 25%' )
        h2.cmdPrint( 'tc qdisc add dev h2-eth0 root netem delay 50ms' )
```

Рисунок 3.43: *Изменение файла lab_netem_i.py*

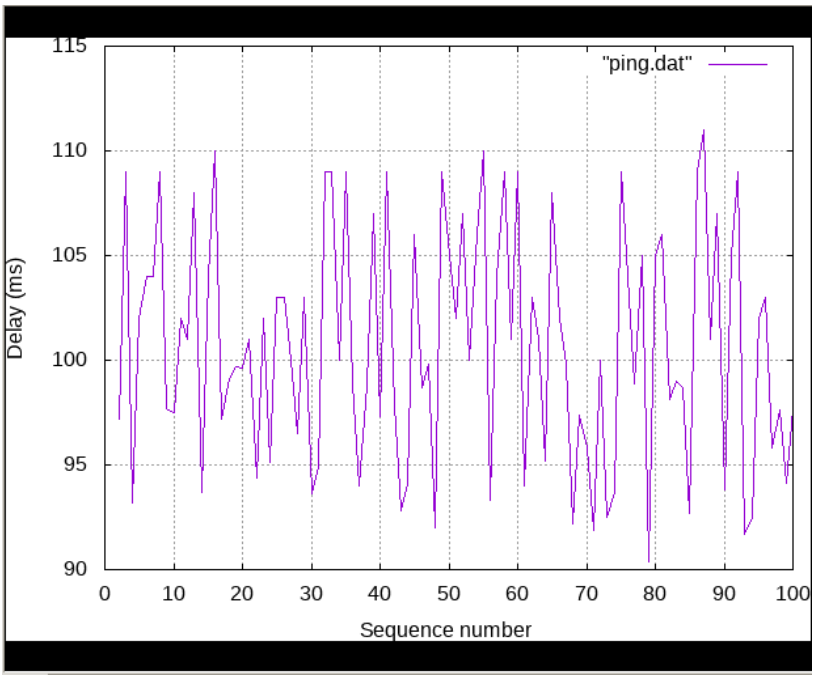

Рисунок 3.44: *График 4*

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make rtt
sudo python rtt.py
min: 90.4
 max: 111.0
 avg: 100.67070707070708
 std: 5.595392647768751
```

Рисунок 3.45: *Вычисленные значения*

37. Самостоятельно реализовал воспроизводимые эксперименты по изменению распределения времени задержки в эмулируемой глобальной сети. Построил графики. Вычислил минимальное, среднее, максимальное и стандартное отклонение времени приёма-передачи для каждого случая.

```
        info( '*** Set delay\n')
        h1.cmdPrint( 'tc qdisc add dev h1-eth0 root netem delay 50ms 10ms distribution normal' )
        h2.cmdPrint( 'tc qdisc add dev h2-eth0 root netem delay 50ms' )

        time.sleep(10) # Wait 10 seconds
```

Рисунок 3.46: *Изменение файла lab_netem_i.py*

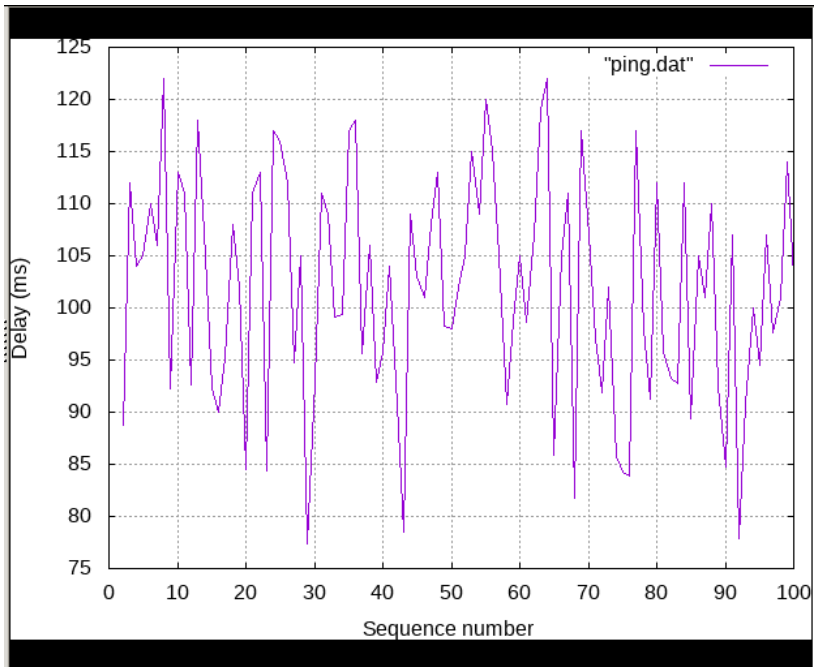

Рисунок 3.47: *График 5*

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make rtt
sudo python rtt.py
min: 77.4
 max: 122.0
 avg: 101.76060606060605
 std: 10.857676849479262
```

Рисунок 3.48: *Вычисленные значения*

4 Выводы

Я ознакомился с NETEM и получила навыки проведения интерактивного и воспроизводимого экспериментов по измерению задержки и её дрожания в моделируемой сети в среде Mininet.