Отчёт по лабораторной работе №4

Эмуляция и измерение задержек в глобальных сетях

Ким Реачна

Содержание

Список иллюстраций

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

1. Запустила виртуальную среду с mininet и исправила права запуска X-соединения.

2. Настройка простейшей топологии и проверка соединения между узлами h1 и h2

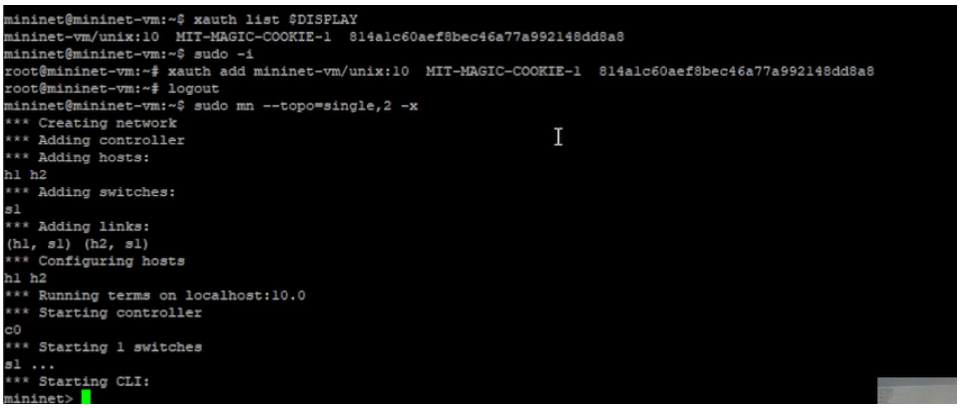

Рис. 2.1: Права запуска X-соединения

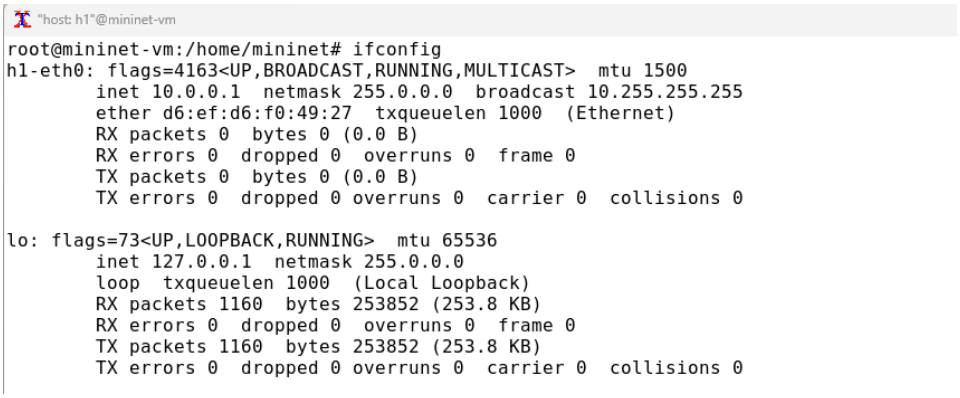

Рис. 2.2: Информация о сетевом интерфейсе и IP-адресе h1

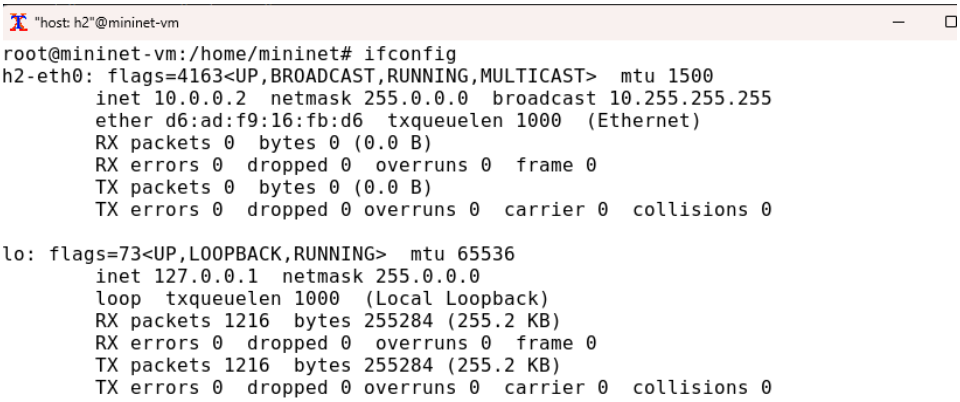

Рис. 2.3: Информация о сетевом интерфейсе и IP-адресе h2

```
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=2.09 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.247 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.045 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.046 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.082 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 5105ms
rtt min/avg/max/mdev = 0.045/0.426/2.092/0.748 ms
```

Рис. 2.4: Проверка соединения от h1 к h2

Минимальное, среднее, максимальное и стандартное отклонение $min = 0.045, avg = 0.426, max = 2.092, mdev = 0.748$.

```
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=0.918 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.639 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.181 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.050 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.054 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.050 ms

--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5075ms
rtt min/avg/max/mdev = 0.050/0.315/0.918/0.340 ms
```

Рис. 2.5: Проверка соединения от h1 к h2

Минимальное, среднее, максимальное и стандартное отклонение $min = 0.050, avg = 0.315, max = 0.918, mdev = 0.340$.

3. На хосте h1 добавила задержку в 100 мс и проверила соединения: $min = 100.543, avg = 100.781, max = 101.103, mdev = 0.233$

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms
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=101 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=101 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5010ms
rtt min/avg/max/mdev = 100.543/100.781/101.103/0.233 ms
```

Рис. 2.6: Добавление задержку в 100 мс на h1

4. На хосте h2 добавила задержку в 100 мс и проверила соединения: $min = 200.392, avg = 135.201, max = 201.682, mdev = 0.449$

```
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=202 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=200 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 5004ms
rtt min/avg/max/mdev = 200.392/201.135/201.682/0.449 ms
```

Рис. 2.7: Добавление задержку в 100 мс на h2

5. Изменила задержку со 100 мс на 50 мс для отправителя h1 и и для получателя h2 и проверила соединения: $min = 101.053, avg = 101.332, max = 101.918, mdev = 0.228$

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

Рис. 2.8: Добавление задержку в 50 мс на h2

```
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=101 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=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 5009ms
rtt min/avg/max/mdev = 101.053/101.332/101.918/0.288 ms
```

Рис. 2.9: Добавление задержку в 50 мс на h1

6. Восстановила конфигурацию по умолчанию, удалив все правила и проверила соединения: $min = 0.050, avg = 0.290, max = 1.031, mdev = 0.346$

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

Рис. 2.10: Восстановила конфигурацию по умолчанию для h2

```
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
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=1.03 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.356 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.129 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.124 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=0.053 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5082ms
rtt min/avg/max/mdev = 0.050/0.290/1.031/0.346 ms
```

Рис. 2.11: Восстановила конфигурацию по умолчанию для h1

7. Добавила на узле h1 задержку в 100 мс со случайным отклонением 10 мс и проверила соединения: $min = 94.499, avg = 101.029, max = 110.419, mdev = 5.781$

```
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=110 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=104 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=94.5 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=104 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=94.6 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=98.2 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 94.499/101.029/110.419/5.781 ms
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
```

Рис. 2.12: Дрожания задержки

8. Добавила на интерфейсе хоста h1 задержку в 100 мс с вариацией ±10 мс и значением корреляции в 25% и проверила соединения: $min = 96.642, avg = 103.190, max = 109.136, mdev = 5.424$

```
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 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=98.5 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=108 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=109 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=98.2 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=96.6 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=108 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 96.642/103.190/109.136/5.424 ms
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
```

Рис. 2.13: Корреляции для джиттера

9. Установила нормальное распределение задержки на узле h1 в эмулируемой сети и проверила соединения: $min = 87.047, avg = 98.155, max = 108.772, mdev = 7.835$

```
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 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=91.2 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=104 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=94.1 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=104 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=87.0 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=109 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5003ms
rtt min/avg/max/mdev = 87.047/98.155/108.772/7.835 ms
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
root@mininet-vm:/home/mininet#
```

Рис. 2.14: Распределение задержки в интерфейсе подключения к эмулируемой глобальной сети

10. Обновила репозитории программного обеспечения на виртуальной машине и установила пакет geeqie и создала каталог, в который будут размещаться файлы эксперимента:

```
mininet@mininet-vm:~$ sudo apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/main i386 Packages [680 kB]
Hit:5 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:6 http://us.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [913 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [2,604 kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [402 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [2,411 kB]
Get:10 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en [336 kB]
Get:11 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [914 kB]
Get:12 http://security.ubuntu.com/ubuntu focal-security/universe i386 Packages [633 kB]
Get:13 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [192 kB]
Get:14 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [2,994 kB]
Get:15 http://us.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [484 kB]
Get:16 http://us.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [2,528 kB]
Get:17 http://us.archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [353 kB]
Get:18 http://us.archive.ubuntu.com/ubuntu focal-updates/universe i386 Packages [761 kB]
```

Рис. 2.15: Обновление репозиториев программного обеспечения

```
mininet@mininet-vm:~$ sudo apt install geeqie
Reading package lists... Done
Building dependency tree
Reading state information... Done
geeqie is already the newest version (1:1.5.1-8build1).
0 upgraded, 0 newly installed, 0 to remove and 361 not upgraded.
mininet@mininet-vm:~$
```

Рис. 2.16: Установка пакета geeqie

```
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
```

Рис. 2.17: Создание каталогов

11. Провела вопроизводимый эксперимент

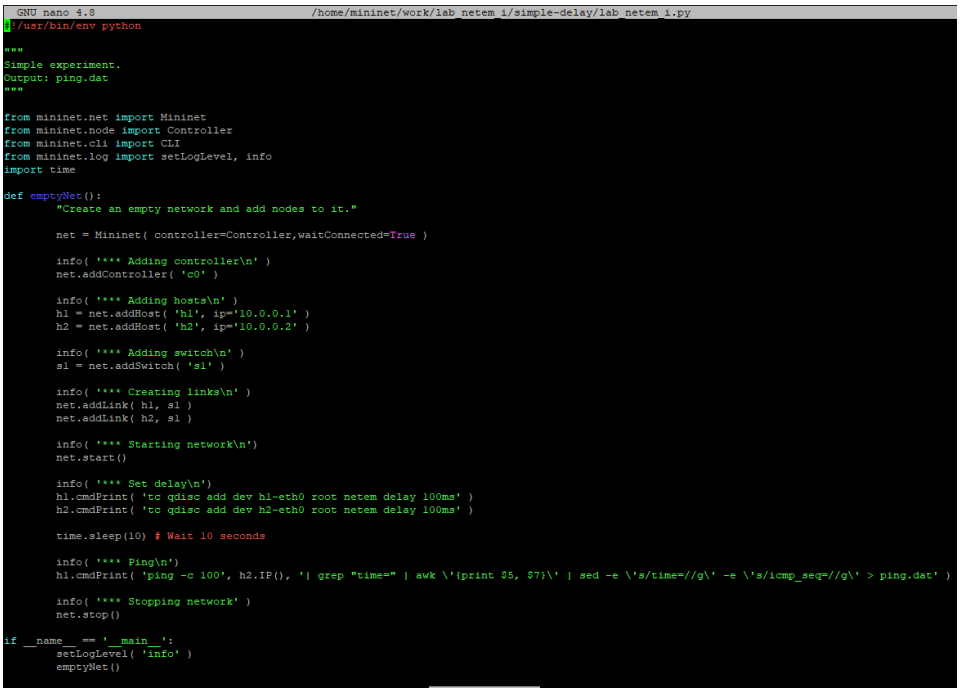

Рис. 2.18: Скрипт lab_netem_i.py

Рис. 2.19: Скрипт ping_plot

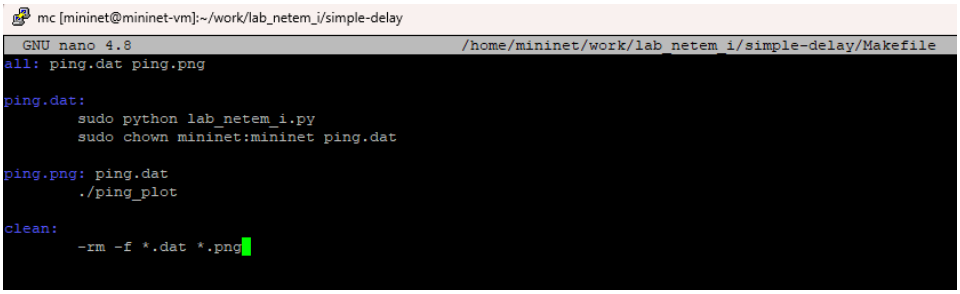

Рис. 2.20: Makefile

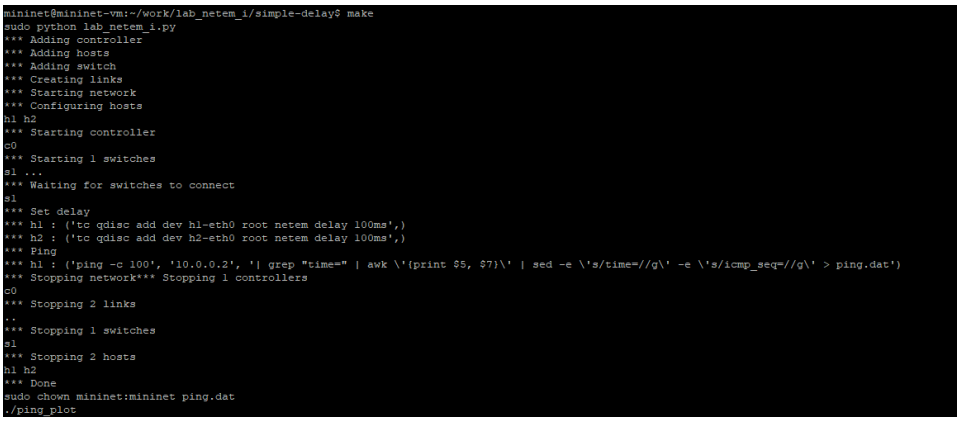

Рис. 2.21: Выполнение эксперимент

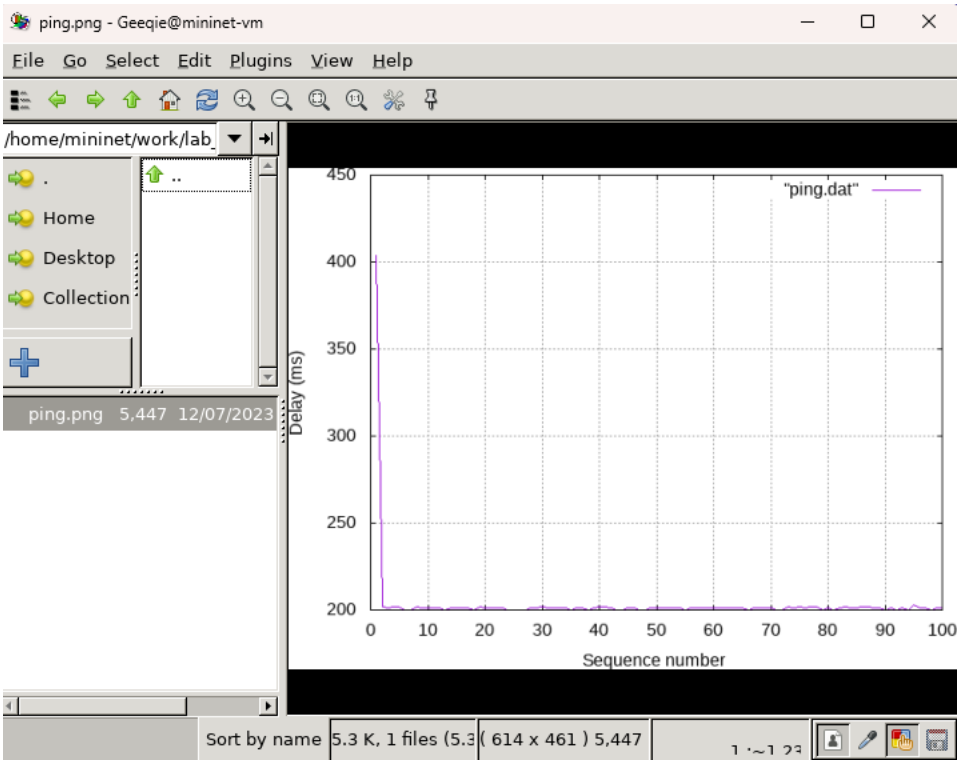

Рис. 2.22: График

- Из файла ping.dat удалите первую строку и заново постройте график: `make ping.png`

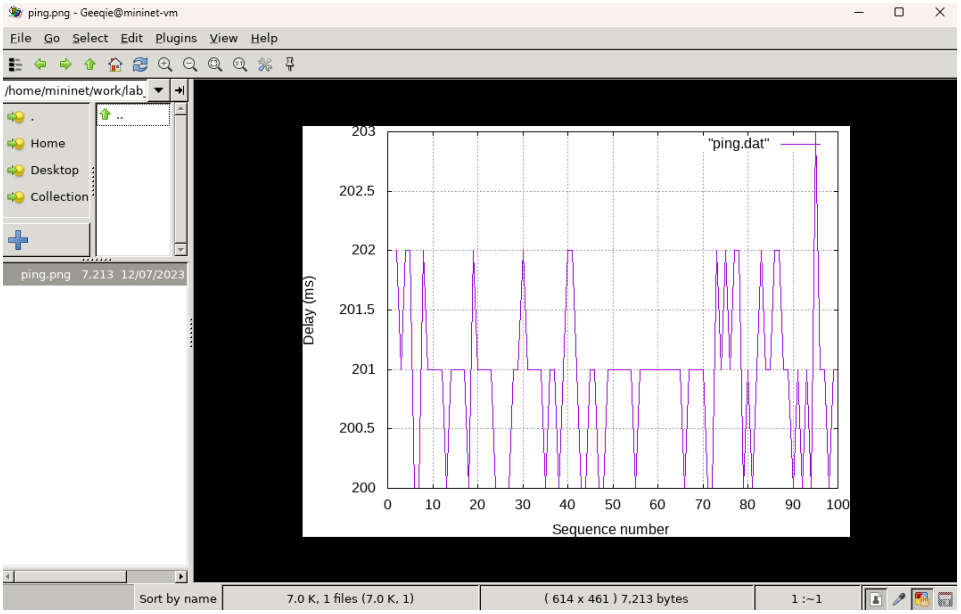

Рис. 2.23: График после удаления одной строки из ping.dat

12. Разработала скрипт для вычисления минимального, среднего, максимального и стандартного отклонения времени приема-передачи на основе данных файла ping.dat, добавила правило запуска скрипта в Makefile.

```
  GNU nano 4.8                    /home/mininet/work/lab_netem_i/simple-delay/ping_statistics.py
import statistics

def compute_statistics(ping_data):
    times = [float(line.split()[1]) for line in ping_data]

    min_time = min(times)
    avg_time = statistics.mean(times)
    max_time = max(times)
    std_dev = statistics.stdev(times)
    return min_time, avg_time, max_time, std_dev

def main():
    with open('ping.dat', 'r') as file:
        data = file.readlines()

    min_time, avg_time, max_time, std_dev = compute_statistics(data)

    print(f"Min time: {min_time} ms")
    print(f"Average time: {avg_time} ms")
    print(f"Max time: {max_time} ms")
    print(f"Standard dev: {std_dev} ms")

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

Рис. 2.24: Скрипт с выводом значений

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make clean
rm -f *.dat *.png ping_result.txt
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
python ping_statistics.py > ping_result.txt
cat ping_result.txt
Min time: 200.0 ms
Average time: 203.02 ms
Max time: 404.0 ms
Standard dev: 20.31070772978747 ms
```

Рис. 2.25: Запуск

2.0.1 Задание для самостоятельной работы

1. Реализовала воспроизводимые эксперименты по изменению задержки:

```python
#!/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 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}\' | sed -e \'s/time=//g\' -e \'s/icmp_seq=//g\' > ping.dat' )

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

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

Рис. 2.26: Скрипт lab_netem_i.py для изменения задержки

```
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
python ping_statistics.py > ping_result.txt
cat ping_result.txt
Min time: 100.0 ms
Average time: 101.96 ms
Max time: 204.0 ms
Standard dev: 10.317113369679927 ms
```

Рис. 2.27: Выполнение эксперимент

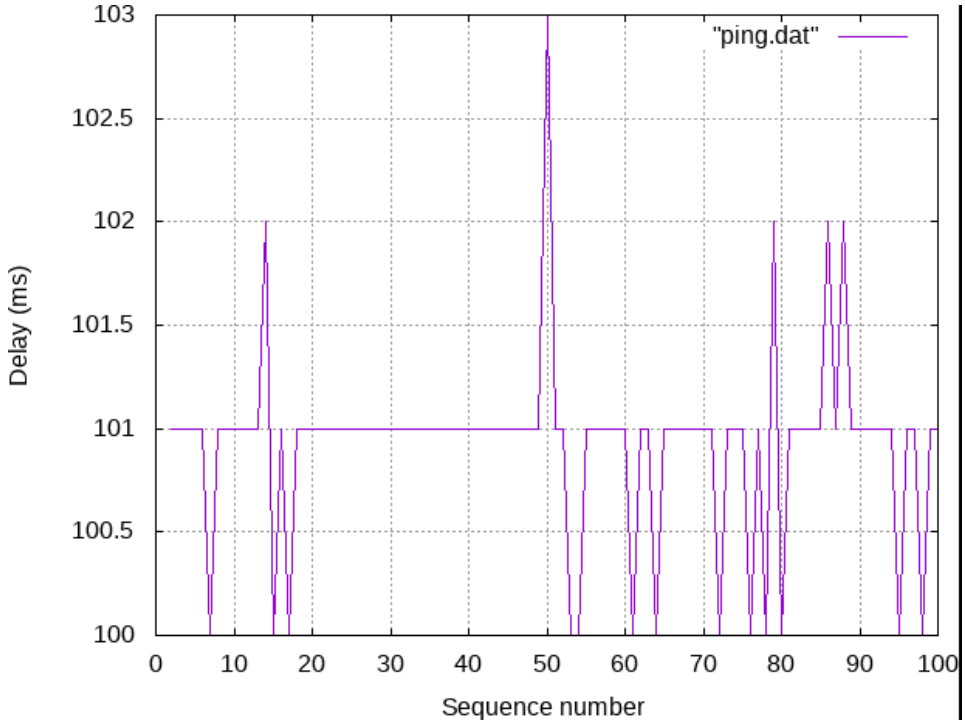

Рис. 2.28: График

2. Реализовала воспроизводимые эксперименты по джиттера, значения корреляции для джиттера и задержки:

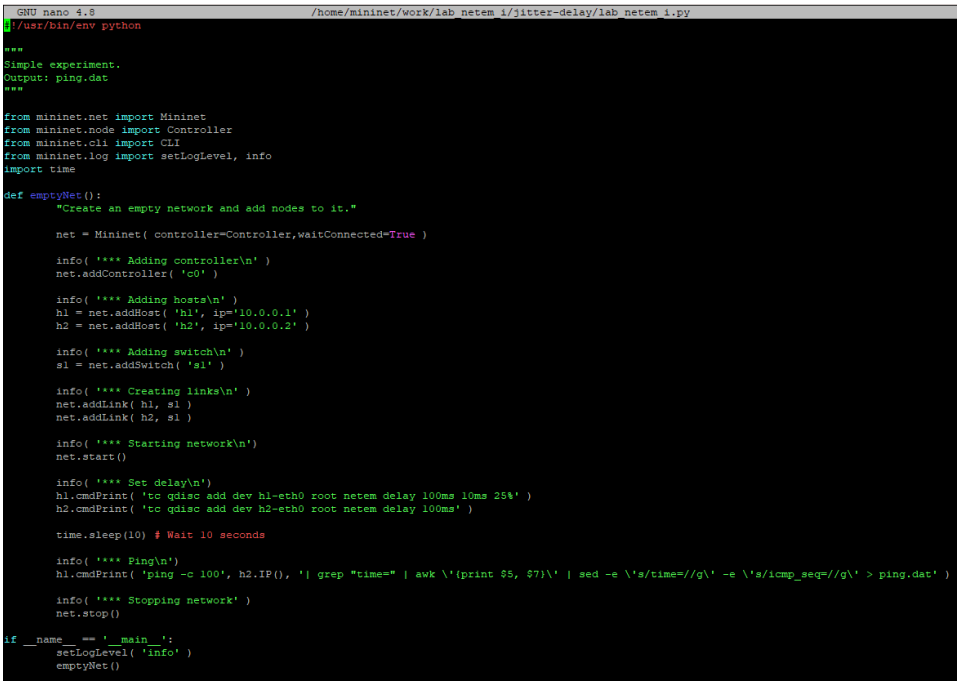

Рис. 2.29: Скрипт lab_netem_i.py для джиттера

Рис. 2.30: Выполнение эксперимент

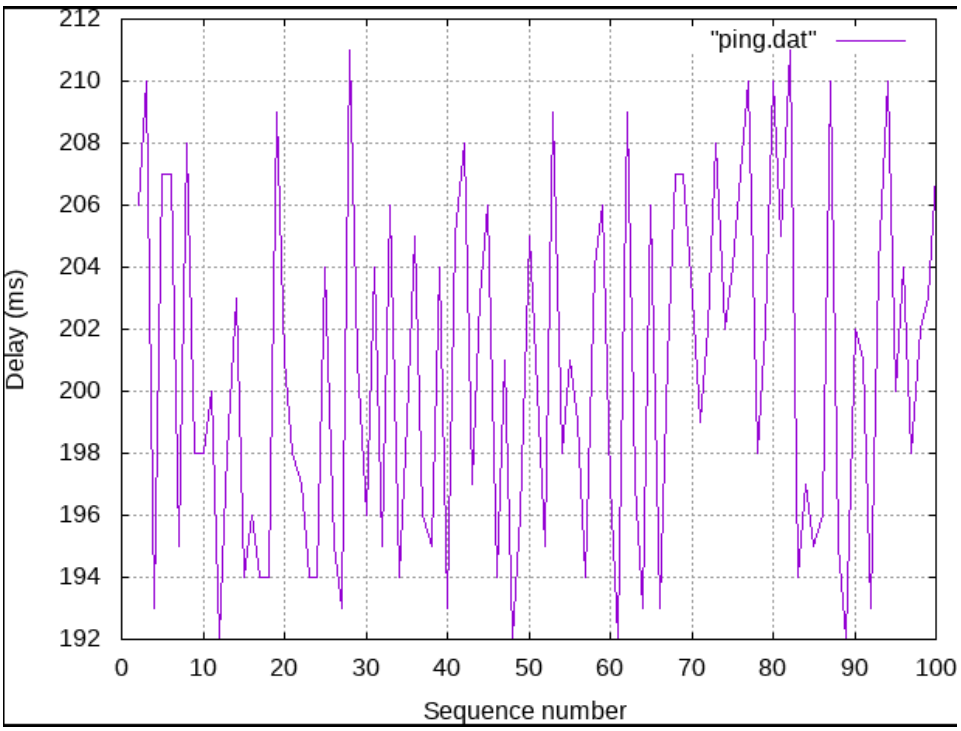

Рис. 2.31: График

3. Реализовала воспроизводимые эксперименты распределение задержки:

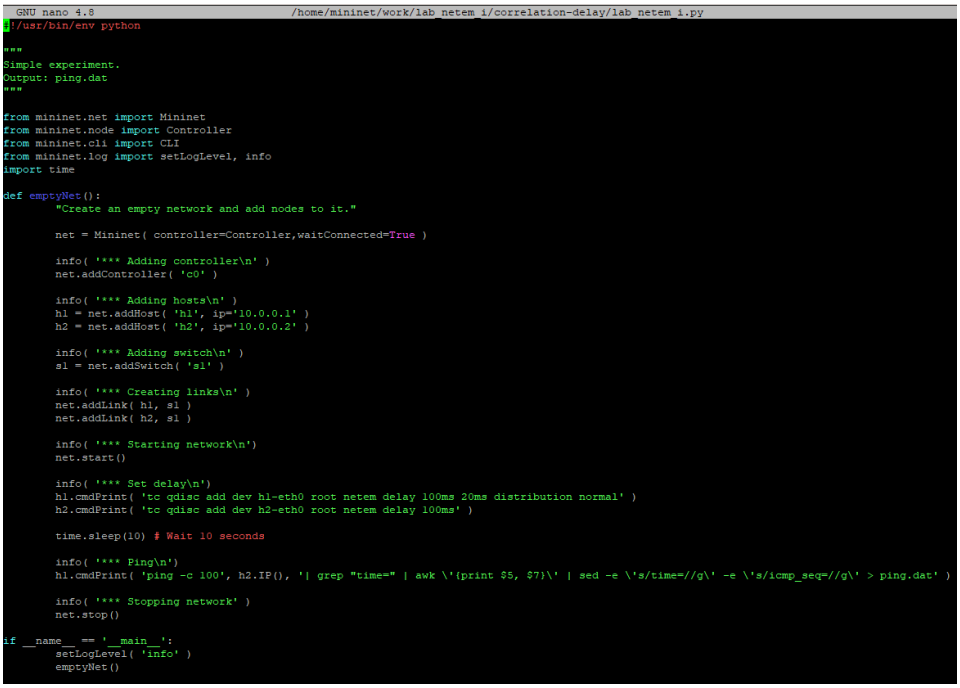

Рис. 2.32: Скрипт lab_netem_i.py для распределения времени задержки в эмулируемой глобальной сети

Рис. 2.33: Выполнение эксперимент

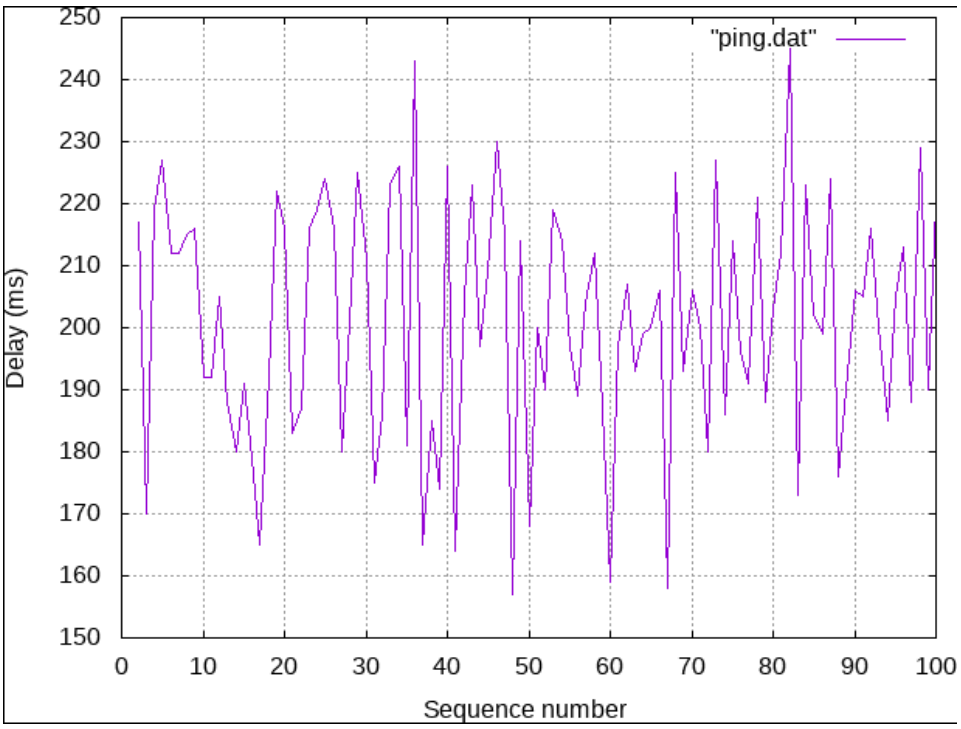

Рис. 2.34: График

3 Листинги программы

- Скрипт lab_netem_i.py для simple-delay:

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

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

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

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

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

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

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

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

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

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

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

    time.sleep(10) # Wait 10 seconds

    info( '*** Ping\n')
    h1.cmdPrint( 'ping -c 100', h2.IP(), '| grep "time="
    | awk \'{print $5, $7}\' | sed -e \'s/time=//g\'
    -e \'s/icmp_seq=//g\' > ping.dat' )

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

if __name__ == '__main__':
```

```
  setLogLevel( 'info' )
  emptyNet()
```

- Скрипт lab_netem_i.py для изменению задержки:

```
#!/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 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}\' | sed -e \'s/time=//g\'
    -e \'s/icmp_seq=//g\' > ping.dat' )

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

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

- Скрипт lab_netem_i.py для джиттера:

```
#!/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 25%' )
  h2.cmdPrint( 'tc qdisc add dev h2-eth0 root netem delay 100ms' )

  time.sleep(10) # Wait 10 seconds

  info( '*** Ping\n')
  h1.cmdPrint( 'ping -c 100', h2.IP(), '| grep "time="
  | awk \'{print $5, $7}\' | sed -e \'s/time=//g\'
  -e \'s/icmp_seq=//g\' > ping.dat' )

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

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

- Скрипт lab_netem_i.py для распределение задержки:

```
#!/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 20ms
  distribution normal' )
  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()
```

- Скрипт ping_statistic.py:

```
import statistics

def compute_statistics(ping_data):
    times = [float(line.split()[1]) for line in ping_data]

    min_time = min(times)
    avg_time = statistics.mean(times)
    max_time = max(times)
    std_dev = statistics.stdev(times)
    return min_time, avg_time, max_time, std_dev
```

```
def main():
    with open('ping.dat', 'r') as file:
        data = file.readlines()

    min_time, avg_time, max_time, std_dev = compute_statistics(data)

    print(f"Min time: {min_time} ms")
    print(f"Average time: {avg_time} ms")
    print(f"Max time: {max_time} ms")
    print(f"Standard dev: {std_dev} ms")

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

4 Вывод

Я познакомилась 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.