Лабораторная работа № 4

Эмуляция и измерение задержек в глобальных сетях

Старовойтов Е. С.

5 декабря 2024

Информация

Докладчик

- Старовойтов Егор Сергеевич
- студент кафедры ТВиК
- Российский университет дружбы народов
- 1032212281@pfur.ru

Вводная часть

Цели и задачи

Основной целью работы является знакомство с NETEM — инструментом для тестирования производительности приложений в виртуальной сети, а также получение навыков проведения интерактивного и воспроизводимого экспе-риментов по измерению задержки и её дрожания (jitter) в моделируемой сети в среде Mininet.

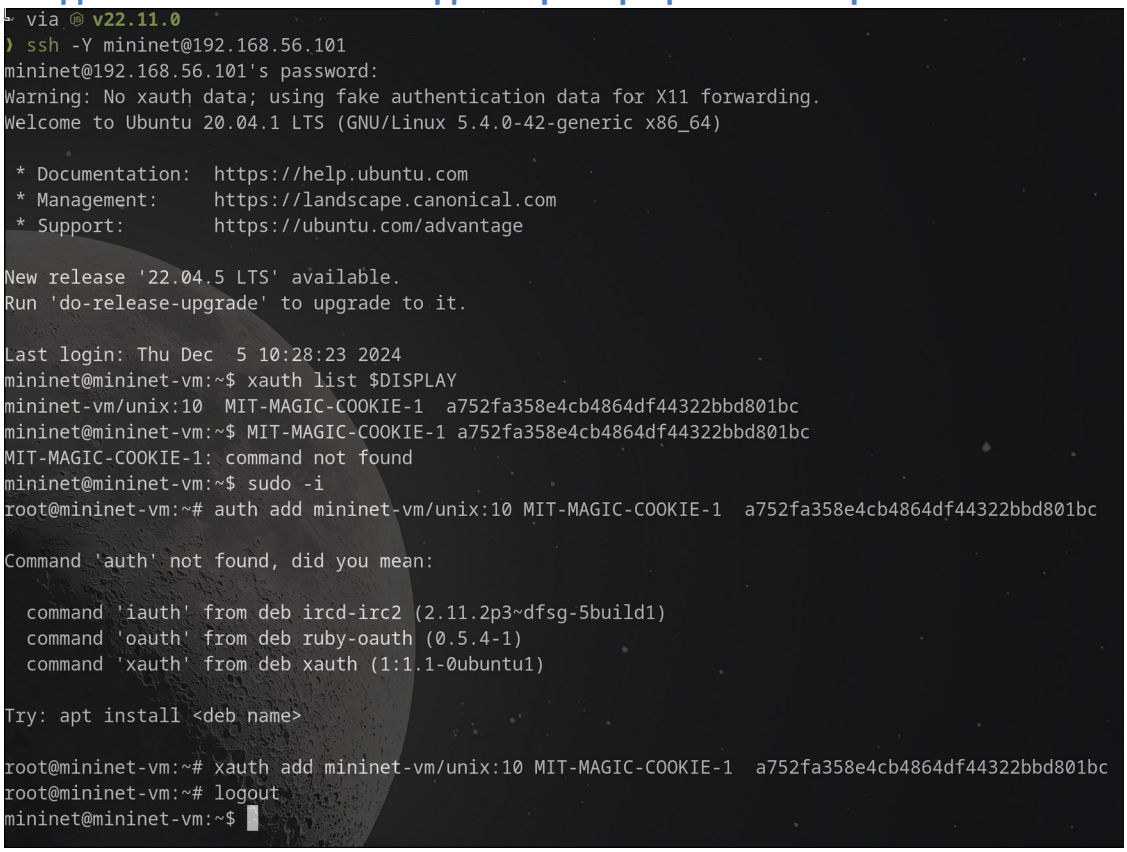
1. Задайте простейшую топологию, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8.
2. Проведите интерактивные эксперименты по добавлению/изменению задерж- ки, джиттера, значения корреляции для джиттера и задержки, распределения времени задержки в эмулируемой 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, рас- пределения времени задержки в эмулируемой глобальной сети. Постройте графики.

Результаты

Поставленные боевые задачи были выполнены, все цели достигнуты.

Выполнение лабораторной работы

1. Подключение по ssh и выдача прав графическим приложениям



Step 1

2. Терминалы хостов h1 и h2.

```
root@mininet-vm:/home/mininet# ifconfig
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTIC
AST>  mtu 1500
        inet 10.0.0.2  netmask 255.0.0.0  broad
cast 10.255.255.255
        ether fa:ed:9d:e7:df:33  txqueuelen 100
0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  fra
me 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carr
ier 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 1074  bytes 270456 (270.4 KB
)
        RX errors 0  dropped 0  overruns 0  fra
me 0
        TX packets 1074  bytes 270456 (270.4 KB
)
        TX errors 0  dropped 0 overruns 0  carr
ier 0  collisions 0

root@mininet-vm:/home/mininet# whoami
root
root@mininet-vm:/home/mininet# tty
/dev/pts/8
root@mininet-vm:/home/mininet#
```

```
root@mininet-vm:/home/mininet# tty
/dev/pts/5
root@mininet-vm:/home/mininet# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTIC
AST>  mtu 1500
        inet 10.0.0.1  netmask 255.0.0.0  broad
cast 10.255.255.255
        ether 9a:67:53:f7:d5:fb  txqueuelen 100
0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  fra
me 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carr
ier 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 1261  bytes 285372 (285.3 KB
)
        RX errors 0  dropped 0  overruns 0  fra
me 0
        TX packets 1261  bytes 285372 (285.3 KB
)
        TX errors 0  dropped 0 overruns 0  carr
ier 0  collisions 0

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

Step 2

3. ping (h2 - слева, h1 - справа)

```
root
root@mininet-vm:/home/mininet# tty
/dev/pts/8
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=2.22 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.124 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.027 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.020 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.029 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.026 ms

--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5093ms
rtt min/avg/max/mdev = 0.020/0.407/2.218/0.810 ms
root@mininet-vm:/home/mininet#
```

```
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=2.00 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.032 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=0.030 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.027 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.022 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5097ms
rtt min/avg/max/mdev = 0.022/0.356/1.998/0.734 ms
root@mininet-vm:/home/mininet#
```

Step 3

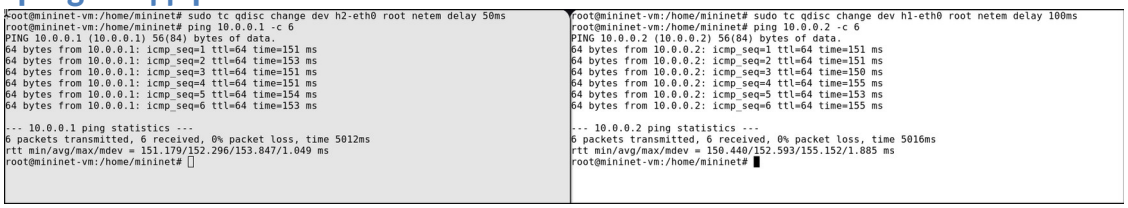
4. ping с установленными задержками

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms
Cannot find device "h1-eth0"
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 fa:ed:9d:e7:df:33  txqueuelen 1000  (Ethernet)
        RX packets 14  bytes 1260 (1.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 14  bytes 1260 (1.2 KB)
        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 2589  bytes 517844 (517.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2589  bytes 517844 (517.8 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=205 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=206 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=206 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=203 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=205 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=203 ms

--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5007ms
rtt min/avg/max/mdev = 202.636/204.563/206.384/1.426 ms
root@mininet-vm:/home/mininet#
```

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h2-eth0 root netem delay 100ms
Cannot find device "h2-eth0"
root@mininet-vm:/home/mininet#
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=201 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=203 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=202 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=201 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=201 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=202 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5017ms
rtt min/avg/max/mdev = 200.729/201.713/202.731/0.627 ms
root@mininet-vm:/home/mininet#
```

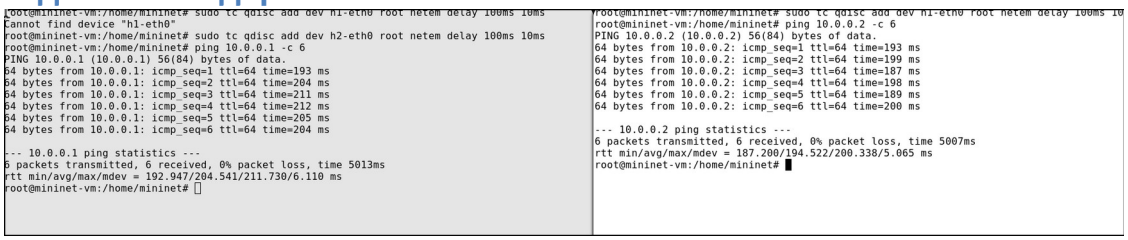
Step 4

5. ping с задержками 50мс



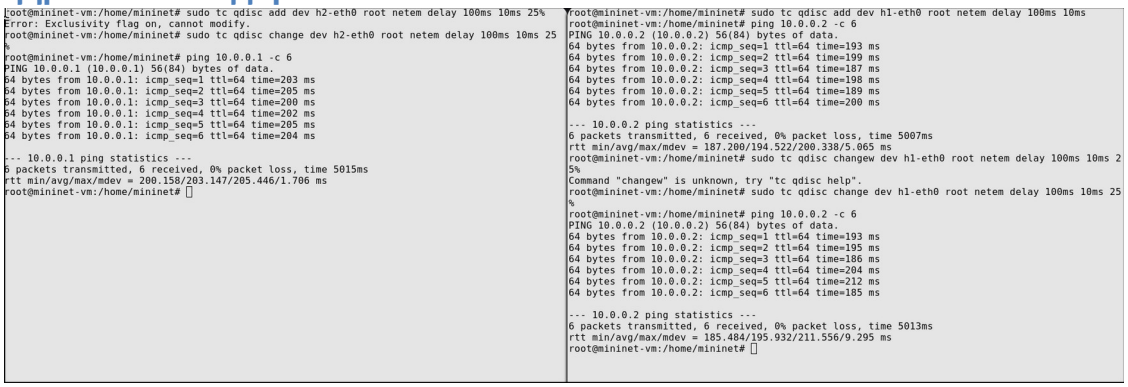
Step 5

6. Удаление задержек



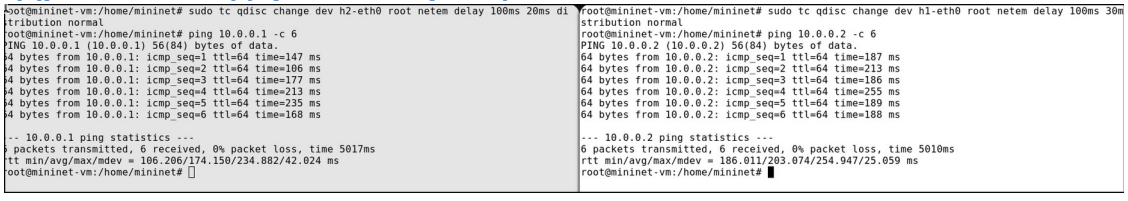
Step 6

7. Дрожание задержки



Step 7

8. Дрожание задержки с вариациями



Step 8

9. Задержки с нормальным распределением

```
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()
~
~
~
~
~
~
~
~
~
~
~
lab_netem_i.py                                                    44,0-1          Bot
"lab_netem_i.py" 47L, 1226C written
```

Step 9

10. lab_netem script

```
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
```

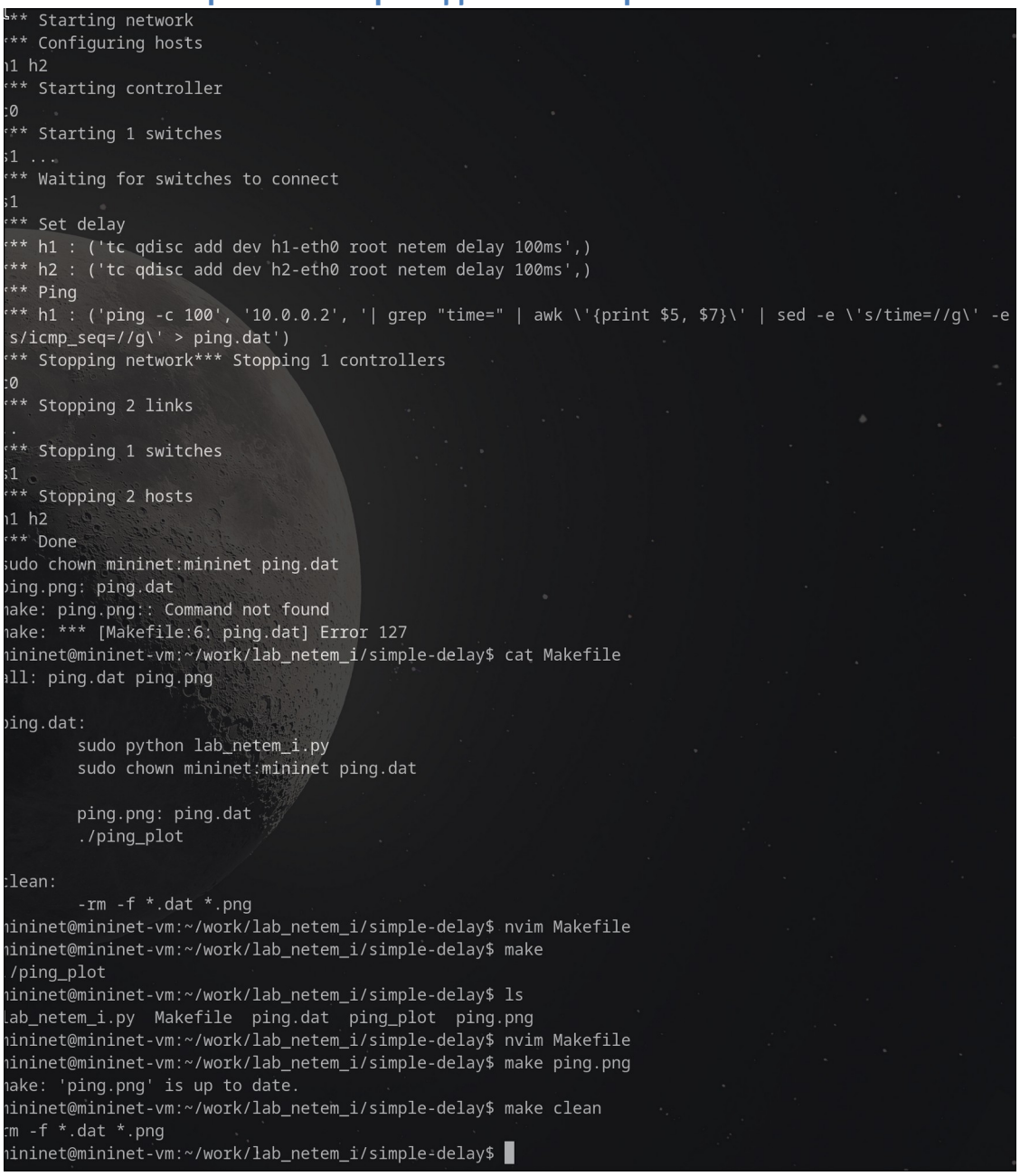
Step 10

11. 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
```

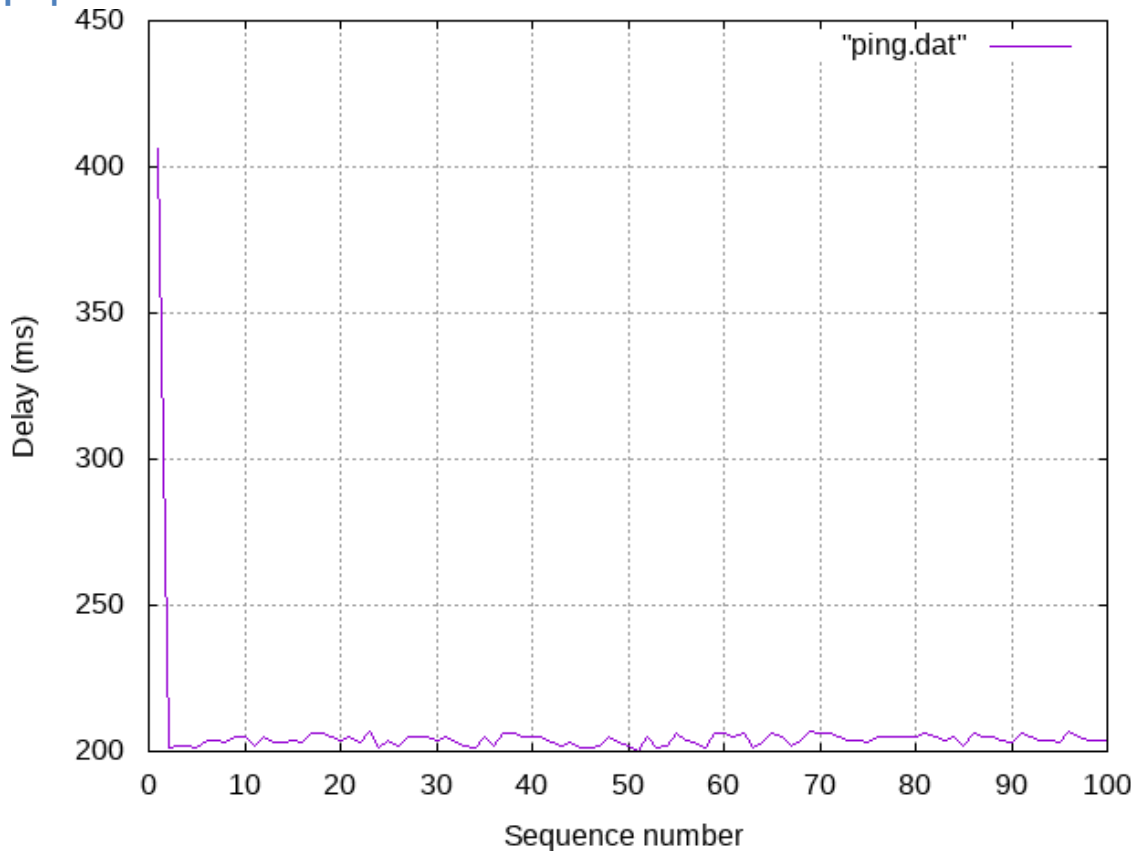
Step 11

12. Автоматизированное проведение эксперимента



Step 12

Графики



ping

Выводы

Я познакомился с NETEM — инструментом для тестирования производительности приложений в виртуальной сети, а также получение навыков проведения интерактивного и воспроизводимого экспе- риментов по измерению задержки и её дрожания (jitter) в моделируемой сети в среде Mininet.