

Отчёт по лабораторной работе №4

НПИбд-02-22

Чесноков Артемий Павлович

Содержание

1 Цель работы	5
2 Выполнение лабораторной работы	6
3 Выводы	22
Список литературы	23

Список иллюстраций

2.1 mn --topo=single, 2 x	6
2.2 host-1	6
2.3 host-2	7
2.4 ping	7
2.5 cli = tc	8
2.6 cli = tc	8
2.7 заметна задержка	8
2.8 результат пинга	9
2.9 опция в деле	9
2.10 после delay вторым параметром	9
2.11 третий параметр после delay	10
2.12 distribution normal	10
2.13 ждем....	10
2.14 для наших воспроизводимых экспов	11
2.15 на питоне	11
2.16 несложно	11
2.17 несложно	12
2.18 test	12
2.19 красивый	13
2.20 видим отступ в начале	13
2.21 на питоне	14
2.22 добавили rtt.py	14
2.23 test	15
2.24 4шт	15
2.25 сделали	16
2.26 change	16
2.27 change	17
2.28 dev	17
2.29 dev	18
2.30 dev	18
2.31 cor	19
2.32 cor	19
2.33 cor	20
2.34 pareto	20
2.35 pareto	21
2.36 pareto	21

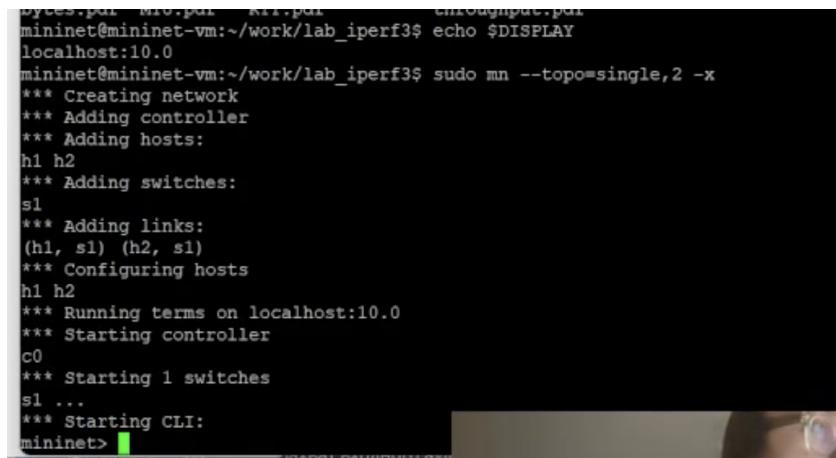
Список таблиц

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

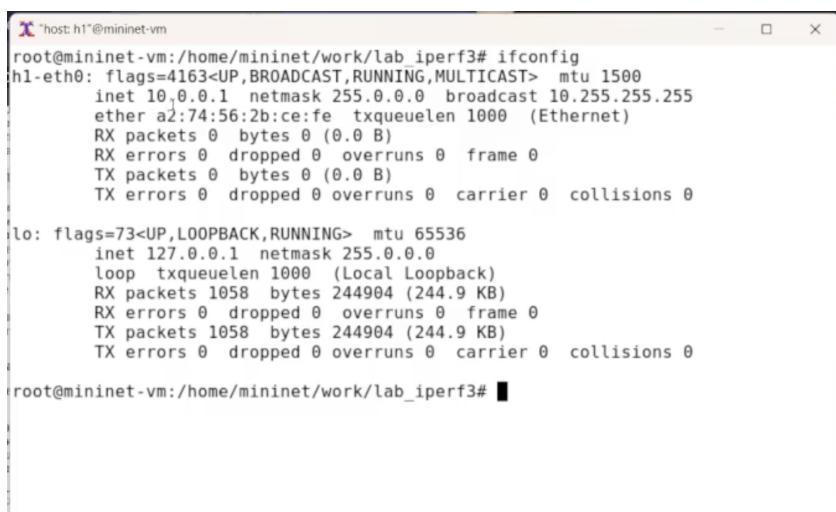
Задаем простейшую топологию (рис. 2.1).



```
bytes-per-second-pair max-pair    mininet@mininet-vm:~/work/lab_iperf3$ echo $DISPLAY
localhost:10.0
mininet@mininet-vm:~/work/lab_iperf3$ 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: mn –topo=single, 2 x

Проверяем автоматическое выставление ip (рис. 2.2).



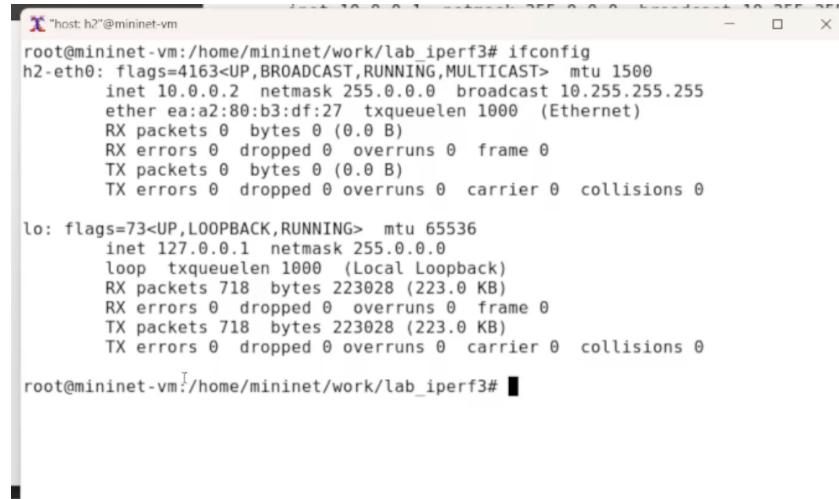
```
"host: h1"@mininet-vm
root@mininet-vm:/home/mininet/work/lab_iperf3# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 10.0.0.1  netmask 255.0.0.0  broadcast 10.255.255.255
        ether a2:74:56:2b:ce:fe  txqueuelen 1000  (Ethernet)
          RX packets 0  bytes 0 (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 1058  bytes 244904 (244.9 KB)
          RX errors 0  dropped 0  overruns 0  frame 0
          TX packets 1058  bytes 244904 (244.9 KB)
          TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@mininet-vm:/home/mininet/work/lab_iperf3#
```

Рис. 2.2: host-1

Проверяем автоматическое выставление ip (рис. 2.3).



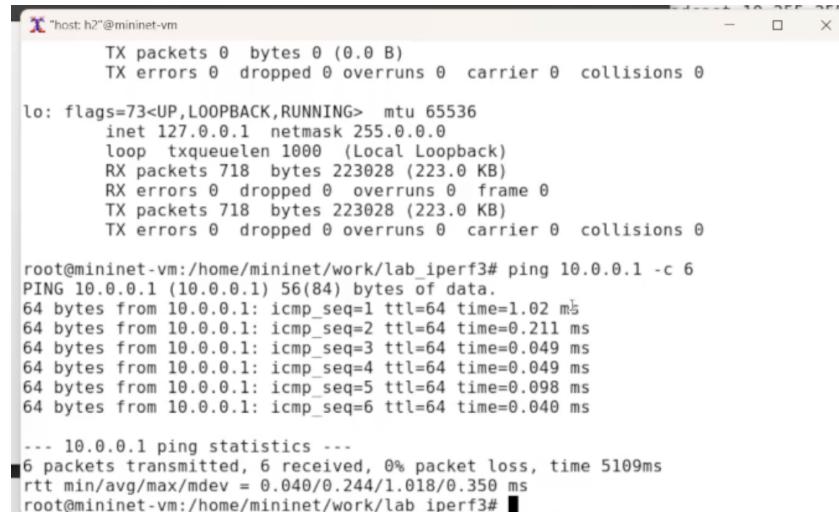
```
root@mininet-vm:/home/mininet/work/lab_iperf3# ifconfig
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.2 netmask 255.0.0.0 broadcast 10.255.255.255
        ether ea:02:b0:b3:df:27 txqueuelen 1000 (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 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 718 bytes 223028 (223.0 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 718 bytes 223028 (223.0 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet/work/lab_iperf3#
```

Рис. 2.3: host-2

Проверяем соединение (рис. 2.4).



```
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 718 bytes 223028 (223.0 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 718 bytes 223028 (223.0 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet/work/lab_iperf3# 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=1.02 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.211 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.049 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.049 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.098 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.040 ms

--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5109ms
rtt min/avg/max/mdev = 0.040/0.244/1.018/0.350 ms
root@mininet-vm:/home/mininet/work/lab_iperf3#
```

Рис. 2.4: ping

Задаем задержку на первый хост 100мс (рис. 2.5).

```

root@mininet-vm:/home/mininet/work/lab_iperf3# sudo tc qdisc change dev h1-eth0 root netem delay 100ms
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=209 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=218 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=216 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 = 100.143/157.449/218.092/57.032 ms
root@mininet-vm:/home/mininet/work/lab_iperf3#

```

Рис. 2.5: cli = tc

Задаем задержку на второй хост (рис. 2.6).

```

64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=128 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 5017ms
rtt min/avg/max/mdev = 100.163/185.001/128.278/10.411 ms
root@mininet-vm:/home/mininet/work/lab_iperf3# sudo tc qdisc add dev h2-eth0 root netem delay 100ms
Cannot find device "h2-eth0"
root@mininet-vm:/home/mininet/work/lab_iperf3# 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=215 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=202 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=390 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=388 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=285 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=212 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5115ms
rtt min/avg/max/mdev = 201.694/282.045/390.239/80.433 ms
root@mininet-vm:/home/mininet/work/lab_iperf3#

```

Рис. 2.6: cli = tc

Проверяем задержку (рис. 2.7).

```

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=102 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=104 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=104 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5045ms
rtt min/avg/max/mdev = 101.563/187.308/121.825/6.855 ms
root@mininet-vm:/home/mininet/work/lab_iperf3# 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=171 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=103 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5030ms
rtt min/avg/max/mdev = 100.787/112.994/170.524/25.737 ms
root@mininet-vm:/home/mininet/work/lab_iperf3#

```

Рис. 2.7: заметна задержка

Убираем задержку (рис. 2.8).

```

loop txqueuelen 1000 (Local Loopback)
RX packets 718 bytes 223028 (223.0 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 718 bytes 223028 (223.0 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet/work/lab_iperf3# 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=1.02 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.211 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.049 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.049 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.098 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.040 ms

--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5109ms
rtt min/avg/max/mdev = 0.040/0.244/1.018/0.350 ms
root@mininet-vm:/home/mininet/work/lab_iperf3# sudo tc qdisc add dev h2-eth0 root netem delay 100ms
root@mininet-vm:/home/mininet/work/lab_iperf3# sudo tc qdisc change dev h2-eth0 root netem delay 50ms
root@mininet-vm:/home/mininet/work/lab_iperf3# sudo tc qdisc del dev h2-eth0 root netem
root@mininet-vm:/home/mininet/work/lab_iperf3#

```

Рис. 2.8: результат пинга

Задержка убирается опцией del (рис. 2.9).

```

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=103 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5030ms
rtt min/avg/max/mdev = 100.787/112.994/170.524/25.737 ms
root@mininet-vm:/home/mininet/work/lab_iperf3# sudo tc qdisc del dev h1-eth0 root netem
root@mininet-vm:/home/mininet/work/lab_iperf3# 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=0.208 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.051 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.078 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.058 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.053 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.054 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5123ms
rtt min/avg/max/mdev = 0.051/0.083/0.208/0.056 ms
root@mininet-vm:/home/mininet/work/lab_iperf3#

```

Рис. 2.9: опция в деле

Добавляем случайное отклонение 10мс

```

Cannot find device "h2-eth0"
root@mininet-vm:/home/mininet/work/lab_iperf3# sudo tc qdisc add dev h1-eth0 root netem delay 100ms 10ms
root@mininet-vm:/home/mininet/work/lab_iperf3# 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=109 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=94.1 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=106 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=102 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=105 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5057ms
rtt min/avg/max/mdev = 94.057/103.199/108.892/4.696 ms
root@mininet-vm:/home/mininet/work/lab_iperf3#

```

Рис. 2.10: после delay вторым параметром

Добавляем корреляцию

```

root@mininet-vm:/home/mininet/work/lab_iperf3# sudo tc qdisc add dev h1-eth0 root netem delay 100ms 10ms 25%
root@mininet-vm:/home/mininet/work/lab_iperf3# 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=100 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=110 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=105 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=108 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=106 ms
--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5054ms
rtt min/avg/max/mdev = 100.146/105.616/110.414/3.315 ms
root@mininet-vm:/home/mininet/work/lab_iperf3#

```

Рис. 2.11: третий параметр после delay

Добавляем распределение

```

root@mininet-vm:/home/mininet/work/lab_iperf3# ping 10.0.0.2 -c 10
rtt min/avg/max/mdev = 92.439/105.263/173.646/17.050 ms
root@mininet-vm:/home/mininet/work/lab_iperf3# sudo tc qdisc del dev h1-eth0 root netem
root@mininet-vm:/home/mininet/work/lab_iperf3# sudo tc qdisc add dev h1-eth0 root netem delay 100ms 20ms distribution normal
root@mininet-vm:/home/mininet/work/lab_iperf3# 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=127 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=78.7 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=94.2 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=56.7 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=133 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=92.9 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=109 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=92.7 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=118 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=105 ms
--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9030ms
rtt min/avg/max/mdev = 56.703/100.768/133.205/21.714 ms
root@mininet-vm:/home/mininet/work/lab_iperf3#

```

Рис. 2.12: distribution normal

Обновляем пакеты, ставим geequi

```

mininet@mininet-vm:~/work/lab_iperf3$ 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 2s (190 kB/s)
Reading package lists... Done
mininet@mininet-vm:~/work/lab_iperf3$ sudo apt-get install geeqie
Reading package lists... Done
Building dependency tree... 50%

```

Рис. 2.13: ждем....

Организовываем папочки

```

Processing triggers for rygel (0.38.3-1ubuntu1) ...
Processing triggers for sgml-base (1.29.1) ...
mininet@mininet-vm:~/work/lab_iperf3$ mkdir -p ~/work/lab_netem_i/simple-delay
mininet@mininet-vm:~/work/lab_iperf3$ cd ..
mininet@mininet-vm:~/work$ ls
demo.mn lab_iperf3 lab_netem_i
mininet@mininet-vm:~/work$ cd lab_netem_i/
mininet@mininet-vm:~/work/lab_netem_i$ ls
simple-delay
mininet@mininet-vm:~/work/lab_netem_i$ cd simple-delay/
[mininet@mininet-vm:~/work/lab_netem_i/simple-delay]$ nano lab_netem_i.py

```

Рис. 2.14: для наших воспроизведимых экспов

Создали скрипт mininet топологии

```

GNU nano 4.0          lab_netem_i.py          Modified
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
info( '*** Stopping network'
net.stop()

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

Save modified buffer?
Y Yes
N No      ^C Cancel

```

Рис. 2.15: на питоне

Написали скрипт для gnu plot

```

GNU nano 4.0          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

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^A Replace  ^U Paste Text  ^T To Spell  ^L Go To Line

```

Рис. 2.16: несложно

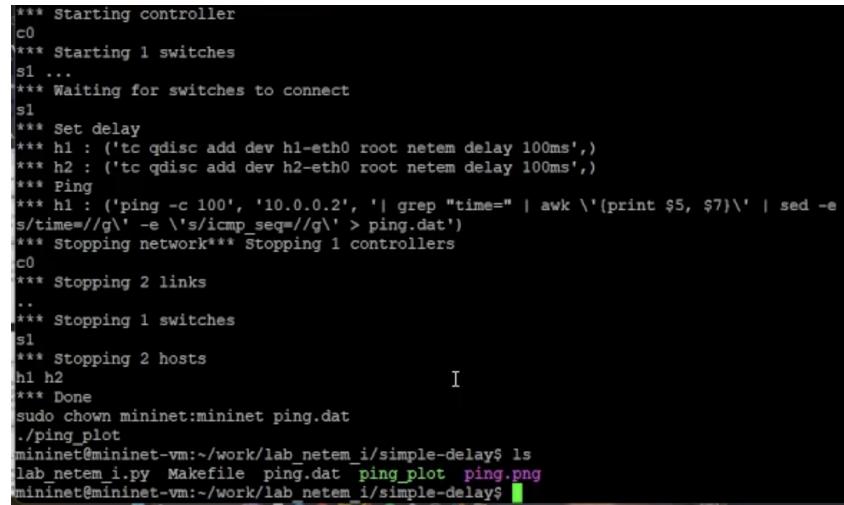
написали makefile



```
/home/mininet-lay/Makefile [-M--] 8 L:[ 1+10 11/ 12] *(141 / 164b) 32 0x020 [*]
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.
```

Рис. 2.17: несложно

Протестировали



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

Рис. 2.18: test

Посмотрели график

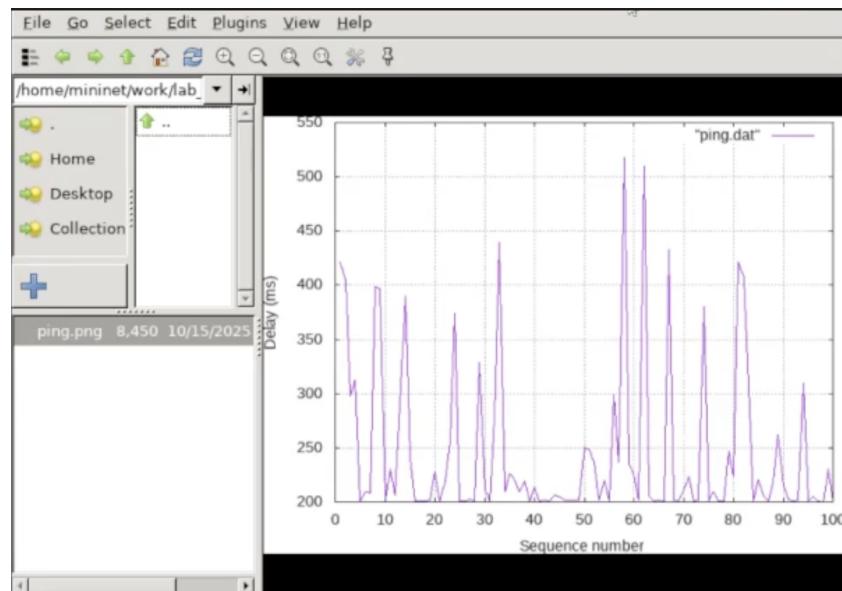


Рис. 2.19: красивый

Смотрим график при убранном из входных первого пакета

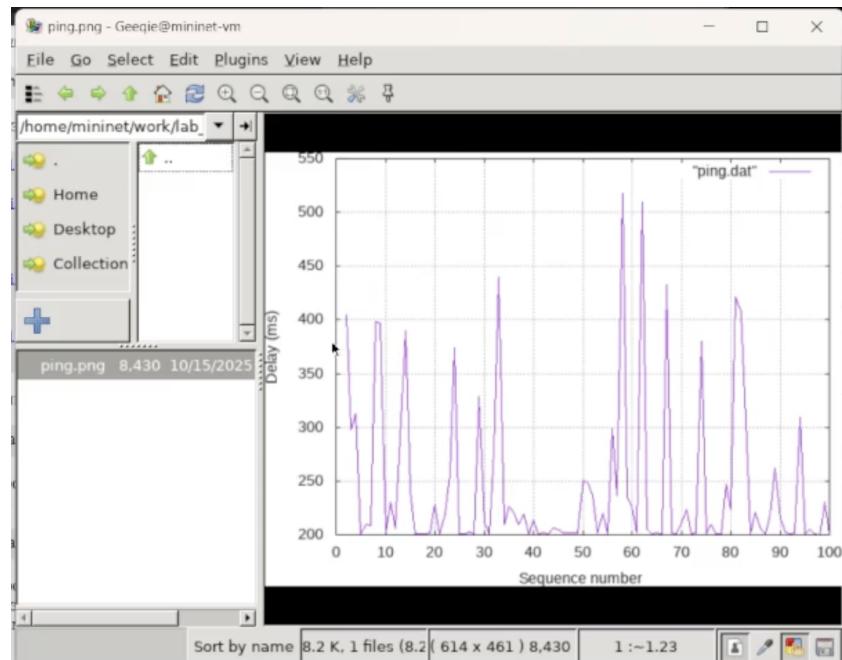


Рис. 2.20: видим отступ в начале

Скрипт для подсчёта метрик rtt

```
GNU nano 4.0 rtt.py Modified
import math

with open("ping.dat", "r") as file:
    values = []
    for line in file:
        parts = line.split()
        if parts:
            try:
                values.append(float(parts[-1]))
            except:
                continue

if values:
    n = len(values)
    avg = sum(values) / n
    std = math.sqrt(sum((x - avg) ** 2 for x in values) / n)
    print(f"min/max/avg/std={min(values):.3f}/{max(values):.3f}/{avg:.3f}/{std:.3f}")
else:
    print("No data found")
```

^G Get Help ^C Write Out ^W Where Is ^K Cut Text ^J Justify ^O Cur Pos
^X Exit ^R Read File ^Y Replace ^U Paste Text ^T To Spell ^L Go To Line

Рис. 2.21: на питоне

Обновляем makefile

```
GNU nano 4.0 Makefile Modified
all: ping.dat ping.png

ping.dat:
    sudo python lab_netem_i.py
    sudo python rtt.py
    sudo chown mininet:mininet ping.dat

ping.png: ping.dat
    ./ping_plot

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

^G Get Help ^C Write Out ^W Where Is ^K Cut Text ^J Justify ^O Cur Pos
^X Exit ^R Read File ^Y Replace ^U Paste Text ^T To Spell ^L Go To Line

Рис. 2.22: добавили rtt.py

Протестили

```

*** 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, $7)\' | sed -e \'s/time//g\' -e \'s/icmp_seq//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
...
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo python rtt.py
min/max/avg/std=200.000/580.000/232.300/70.663
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab netem i/simple-delay$ 

```

Рис. 2.23: test

Сделали папки для СР

```

-rw-rw-r-- 1 mininet mininet 1283 Oct 15 14:49 lab_netem_i.py
-rw-rw-r-- 1 mininet mininet 184 Oct 15 15:13 Makefile
-rwxrwxr-x 1 mininet mininet 170 Oct 15 14:54 ping_plot
-rw-rw-r-- 1 mininet mininet 482 Oct 15 15:13 rtt.py
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nano lab_netem_i.py
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ cd ..
mininet@mininet-vm:~/work/lab_netem_i$ ls
simple-delay
mininet@mininet-vm:~/work/lab_netem_i$ cd simple-delay/
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ ls
lab netem_i.py Makefile ping_plot rtt.py
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ cd ..
mininet@mininet-vm:~/work/lab_netem_i$ ls
simple-delay
mininet@mininet-vm:~/work/lab_netem_i$ mkdir change-delay
mininet@mininet-vm:~/work/lab_netem_i$ mkdir jitter-delay
mininet@mininet-vm:~/work/lab_netem_i$ mkdir correlation-delay
mininet@mininet-vm:~/work/lab_netem_i$ mkdir distribution-delay
mininet@mininet-vm:~/work/lab_netem_i$ cp simple-delay/* jitter-delay/
mininet@mininet-vm:~/work/lab_netem_i$ ls jitter-delay/
lab netem_i.py Makefile ping_plot rtt.py
mininet@mininet-vm:~/work/lab_netem_i$ cp simple-delay/* correlation-delay/
mininet@mininet-vm:~/work/lab_netem_i$ cp simple-delay/* distribution-delay/
mininet@mininet-vm:~/work/lab_netem_i$ ls
change-delay correlation-delay distribution-delay jitter-delay simple-delay
mininet@mininet-vm:~/work/lab netem i$ 

```

Рис. 2.24: 4шт

Для демонстрации изменения задержки в дисц.очередей - делаем два прогона ping'a

```

GNU nano 4.0                               lab netem i.py                         Modified
    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 50', h2.IP(), '| grep "time=' | awk \'{print $5, $7}\' | sed -e \'s/time=/g\' -e \'s/icmp_seq=/g\' >> ping.dat' )

    h1.cmdPrint( 'tc qdisc change dev h1-eth0 root netem delay 100ms' )
    h2.cmdPrint( 'tc qdisc change dev h2-eth0 root netem delay 100ms' )

    h1.cmdPrint( 'ping -c 50', h2.IP(), '| grep "time=' | awk \'{print $5, $7}\' | sed -e \'s/time=/g\' -e \'s/icmp_seq=/g\' >> ping.dat' )

```

Get Help ^{^G} Write Out ^{^W} Where Is ^{^W} Cut Text ^{^K} Justify ^{^J} Cur Pos
X Exit ^{^X} Read File ^{^R} Replace ^{^R} Paste Text ^{^V} To Spell ^{^T} Go To Line ^{^G}

Рис. 2.25: сделали

запуск

```

*** 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 50', '10.0.0.2', '| grep "time=' | awk \'{print $5, $7}\' | sed -e \'s/time=/g\' -e \'s/icmp_seq=/g\' >> ping.dat')
*** h1 : ('tc qdisc change dev h1-eth0 root netem delay 10ms',)
*** h2 : ('tc qdisc change dev h2-eth0 root netem delay 10ms',)
*** h1 : ('ping -c 50', '10.0.0.2', '| grep "time=' | awk \'{print $5, $7}\' | sed -e \'s/time=/g\' -e \'s/icmp_seq=/g\' >> ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
...
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo python rtt.py
min/max/avg/std=20.200/604.000/119.424/106.074
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab netem i/change-delay$ 

```

Рис. 2.26: change

график

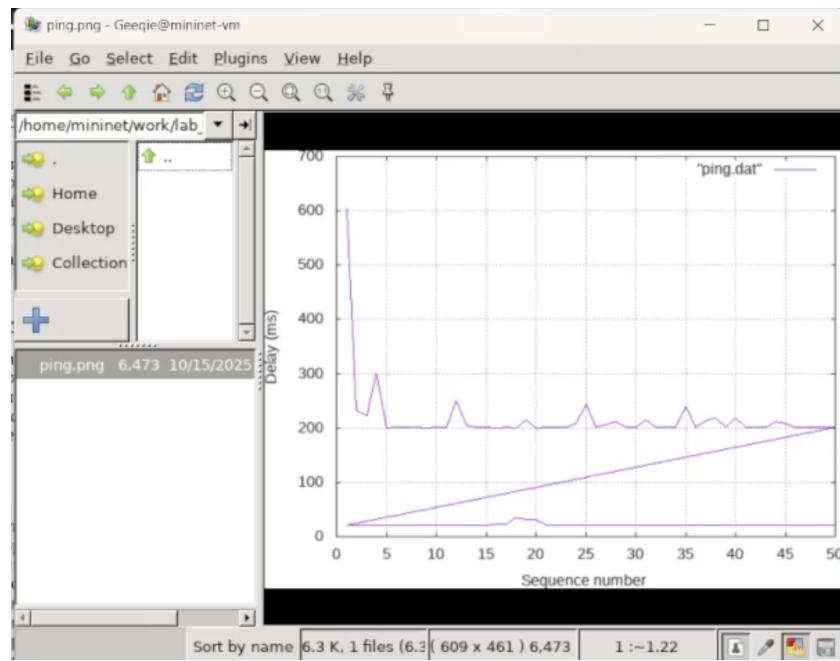


Рис. 2.27: change

Прогон с отклонением

```
GNU nano 4.0                               lab netem i.py                         Modified
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 30ms' )
h2.cmdPrint( 'tc qdisc add dev h2-eth0 root netem delay 100ms 40ms' )
time.sleep(10) # Wait 10 seconds

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

```

Save modified buffer? Yes No Cancel

Рис. 2.28: dev

запуск с отклонением

```

** 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 30ms',)
*** h2 : ('tc qdisc add dev h2-eth0 root netem delay 100ms 40ms',)
*** Ping
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "time=" | awk \'(print $5, $7)\' | sed -e \'s/time=/g\' -e \'s/icmp_seq=/g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
s1 h2
*** Done
sudo python rtt.py
min/max/avg/std=143.000/628.000/238.950/88.995
sudo chown mininet:mininet ping.dat  I
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/jitter-delay$ ls

```

Рис. 2.29: dev

график

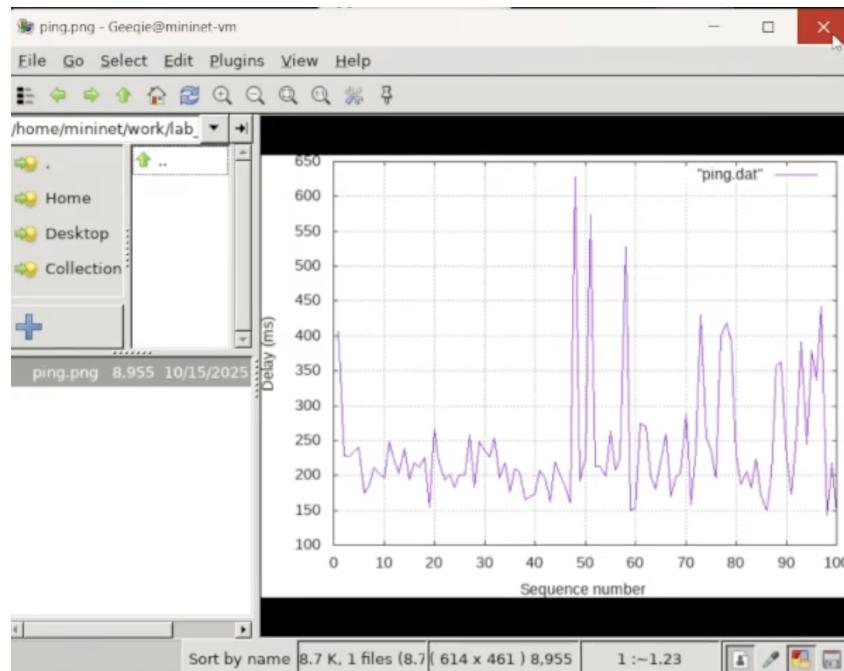


Рис. 2.30: dev

прогон с корреляцией

```

GNU nano 4.8                               lab_netem_i.py
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 35%')
h2.cmdPrint('tc qdisc add dev h2-eth0 root netem delay 100ms 20ms 35%')

time.sleep(10) # Wait 10 seconds

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

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

^G Get Help   ^C Write Out   ^W Where Is   ^X Cut Text   ^J Justify   ^O Cur Pos
^X Exit      ^R Read File   ^A Replace   ^I Paste Text  ^T To Spell   ^L Go To Line

```

Рис. 2.31: cor

запуск топологии

```

./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/jitter-delay$ geedie
mininet@mininet-vm:~/work/lab_netem_i/jitter-delay$ cd ..
mininet@mininet-vm:~/work/lab_netem_i$ ls
change-delay correlation-delay distribution-delay jitter-delay simple-delay
mininet@mininet-vm:~/work/lab_netem_i$ cd correlation-delay/
mininet@mininet-vm:~/work/lab_netem_i/correlation-delay$ nano 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 20ms 35%',)
*** h2 : ('tc qdisc add dev h2-eth0 root netem delay 100ms 20ms 35%',)

```

Рис. 2.32: cor

plot

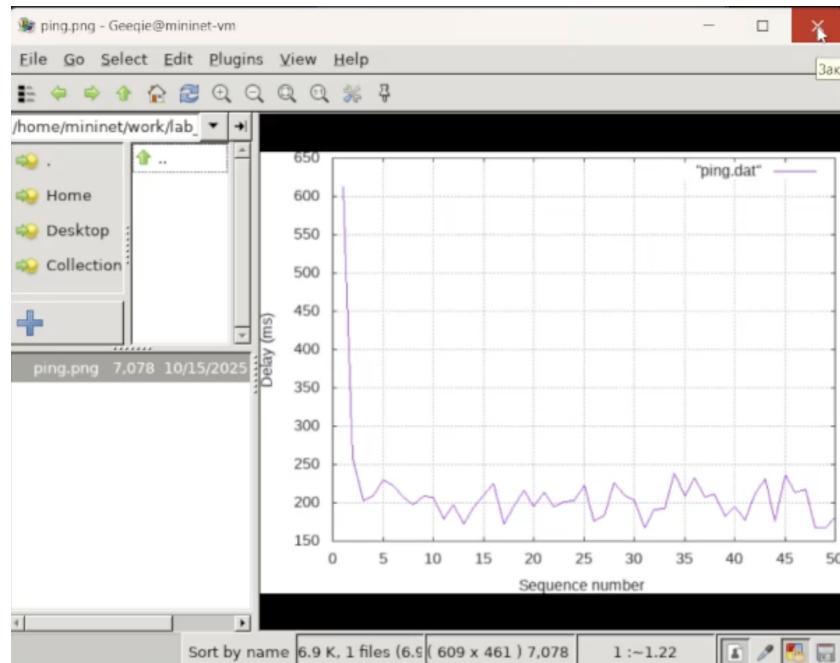


Рис. 2.33: cor

Прогон с паретовским распределением

```

GNU nano 4.0                               lab netem i.py                         Modified
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 distribution pareto' )
h2.cmdPrint( 'tc qdisc add dev h2-eth0 root netem delay 100ms distribution pareto' )

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/\"/./g' )

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

if __name__ == '__main__':

```

Рис. 2.34: pareto

execution

```

*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
*** Set delay
*** h1 : ('tc qdisc add dev h1-eth0 root netem delay 100ms 100ms 25% distribution pareto',
  )
*** h2 : ('tc qdisc add dev h2-eth0 root netem delay 100ms 100ms 25% distribution pareto',
  )
*** Ping
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "time=' | awk \'(print $5, $7)\' | sed -e \'s/time=/g\' -e \'s/icmp_seq=/g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
...
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo python rtt.py
min/max/avg/std=73.800/1115.000/203.003/143.205
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/distribution-delay$ cd distribution-delay/

```

Рис. 2.35: pareto

plot

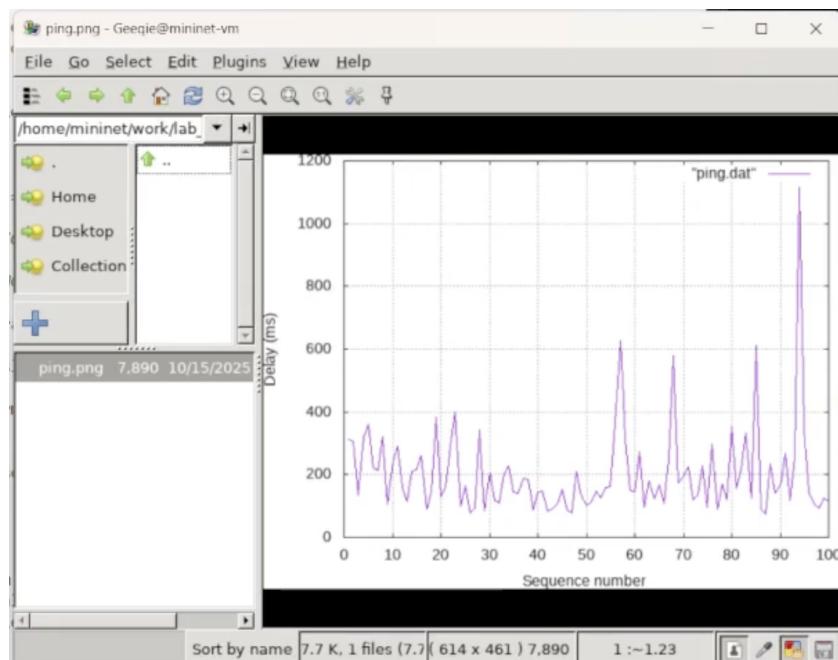


Рис. 2.36: pareto

3 Выводы

Познакомились с NETEM, а также получили навыки проведения интерактивного и воспроизводимого экспериментов по измерению задержки и её дрожания (jitter) в моделируемой сети в среде Mininet.

Список литературы