

Лабораторная работа №4

Эмуляция и измерение задержек в глобальных сетях

Хрусталев В.Н.

Российский университет дружбы народов, Москва, Россия

Информация

- Хрусталев Влад Николаевич
- студент
- Российский университет дружбы народов
- 1132222011@pfur.ru

Цель работы

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.

Выполнение лабораторной работы

Исправление прав запуска X-соединения

```
Last login: Sat Oct 25 10:51:51 2025 from 192.168.56.1
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  0f656a238b0a277ac162ec35e2a18e23
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth list $DISPLAY
mininet-vm/ unix:10  MIT-MAGIC-COOKIE-1  0f656a238b0a277ac162ec35e2a18e23
root@mininet-vm:~# logout
mininet@mininet-vm:~$ |
```

Рис. 1: Исправление прав запуска X-соединения

Простейшая топология

```
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: Простейшая топология

ifconfig на хостах h1 и h2

```
X "host: h1"@mininet-vm
root@mininet-vm:/home/mininet# ifconfig
\11-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 0e:c3:2d:36:4d:d3 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 1036 bytes 270956 (270.9 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 1036 bytes 270956 (270.9 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

```
X "host: h2"@mininet-vm
root@mininet-vm:/home/mininet# ifconfig
\12-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:2a:5d:c5:ab:c5 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 1406 bytes 307892 (307.8 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 1406 bytes 307892 (307.8 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

Проверка подключения между хостами

```
X "host: h1"@mininet-vm
      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 1036 bytes 270956 (270.9 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 1036 bytes 270956 (270.9 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 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=4.50 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.466 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.092 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.092 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.101 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.109 ms

... 10.0.0.2 ping statistics ...
6 packets transmitted, 6 received, 0% packet loss, time 5096ms
rtt min/avg/max/mdev = 0.092/0.893/4.498/1.617 ms
root@mininet-vm:/home/mininet#
```

```
X "host: h2"@mininet-vm
      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 1406 bytes 307892 (307.8 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 1406 bytes 307892 (307.8 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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=3.27 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.085 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.122 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.085 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.113 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.114 ms

... 10.0.0.1 ping statistics ...
6 packets transmitted, 6 received, 0% packet loss, time 5078ms
rtt min/avg/max/mdev = 0.085/0.630/3.265/1.178 ms
root@mininet-vm:/home/mininet#
```

Добавление задержки в 100мс

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

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 100.473/101.079/101.771/0.494 ms
root@mininet-vm:/home/mininet# █
```

Рис. 5: Добавление задержки в 100мс

Двунаправленная задержка соединения

```
X "host: h1"@mininet-vm
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=202 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
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 5009ms
rtt min/avg/max/mdev = 200.758/201.321/202.437/0.561 ms
root@mininet-vm:/home/mininet# █
```

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

Изменение задержки на 50мс

```
"host: h1"@mininet-vm
root@mininet-vm:/home/mininet# sudo tc qdisc change dev h1-eth0 root netem delay 50
ms
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=110 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=105 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 5010ms
rtt min/avg/max/mdev = 101.185/103.368/109.648/3.065 ms
root@mininet-vm:/home/mininet#
```

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

Восстановление исходных значений задержки

The image shows two terminal windows side-by-side. Both windows have a title bar "host: h1" and "host: h2" respectively, followed by "@mininet-vm". The windows are running on a Linux system with a root shell.

The top window (host: h1) displays the following command and its output:

```
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=2.40 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.717 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.288 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.090 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.170 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.083 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5095ms
rtt min/avg/max/mdev = 0.083/0.624/2.397/0.821 ms
root@mininet-vm:/home/mininet#
```

The bottom window (host: h2) displays the following command and its output:

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

Рис. 8: Восстановление исходных значений задержки

Добавление значения дрожания задержки в интерфейс подключения

```
X "host: h1"@"mininet-vm
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
Error: Invalid qdisc name.
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms
10ms
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=101 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=97.7 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=94.7 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=107 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 5012ms
rtt min/avg/max/mdev = 94.729/102.178/111.134/5.498 ms
root@mininet-vm:/home/mininet# █
```

Рис. 9: Добавление значения дрожания задержки в интерфейс подключения

Добавление значения корреляции для джиттера и задержки в интерфейс подключения

```
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms
10ms 25%
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=103 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=103 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=105 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=106 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 5009ms
rtt min/avg/max/mdev = 102.915/105.032/108.211/1.788 ms
root@mininet-vm:/home/mininet# █
```

Рис. 10: Добавление значения корреляции для джиттера и задержки в интерфейс подключения

Распределение задержки в интерфейсе подключения

```
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 -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=100 ms  
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=77.0 ms  
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=81.4 ms  
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=77.2 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=100 ms  
  
--- 10.0.0.2 ping statistics ---  
6 packets transmitted, 6 received, 0% packet loss, time 5011ms  
rtt min/avg/max/mdev = 77.024/89.730/102.369/11.292 ms  
root@mininet-vm:/home/mininet# ping -c 10 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=92.6 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=92.9 ms  
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=87.6 ms  
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=90.7 ms  
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=94.1 ms  
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=123 ms  
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=73.4 ms  
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=81.4 ms  
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=85.8 ms  
  
--- 10.0.0.2 ping statistics ---  
10 packets transmitted, 10 received, 0% packet loss, time 9012ms  
rtt min/avg/max/mdev = 73.363/92.244/123.105/12.544 ms  
root@mininet-vm:/home/mininet# █
```

Рис. 11: Распределение задержки в интерфейсе подключения

Подготовка к производимому эксперименту

```
mininet@mininet-vm:~$ sudo apt-get update
Hit:1 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Reading package lists... Done
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 362 not upgraded.
mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_i/expname
mininet@mininet-vm:~$ cd work/lab_netem_i/
mininet@mininet-vm:~/work/lab_netem_i$
```

Рис. 12: Подготовка к производимому эксперименту

Листинг lab_netem_i.py

```
GNU nano 4.8          lab_netem_i.py          Modified
#!/usr/bin/env python

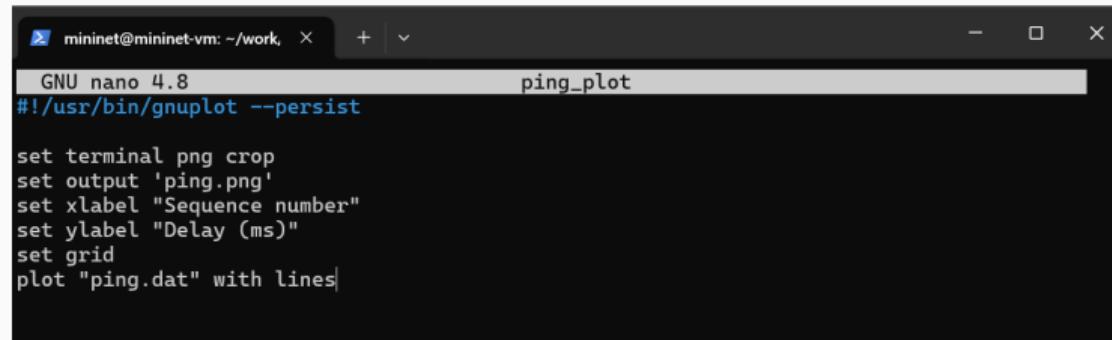
"""
Simple experiment.
Output: ping.dat
"""

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

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

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

Листинг ping_plot



The screenshot shows a terminal window titled "ping_plot" running on a "mininet@mininet-vm" host. The window contains a script written in gnuplot. The script sets up a terminal to output a PNG file named "ping.png", with the x-axis labeled "Sequence number" and the y-axis labeled "Delay (ms)". It enables a grid and plots the data from a file named "ping.dat" using lines.

```
GNU nano 4.8                               ping_plot
#!/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|
```

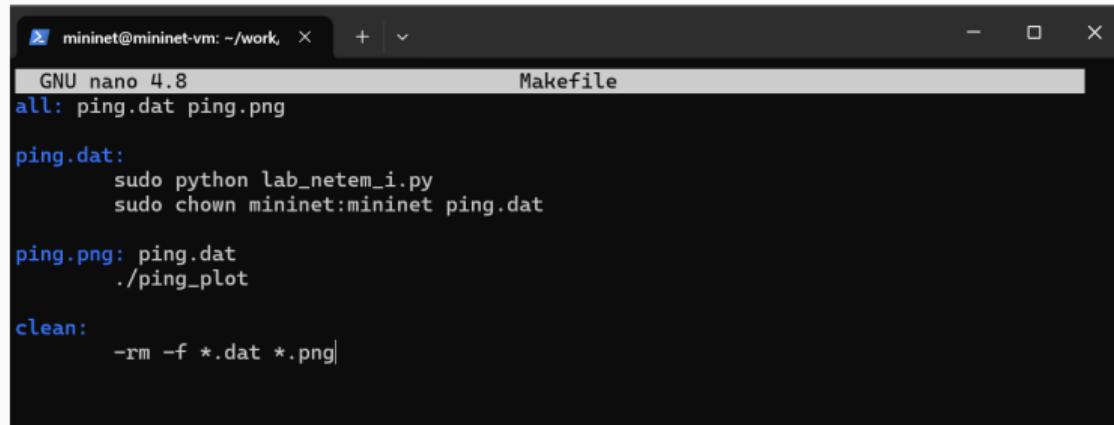
Рис. 14: Листинг ping_plot

Выдача прав выполнения для 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$ |
```

Рис. 15: Выдача прав выполнения для ping_plot

Листинг Makefile



The screenshot shows a terminal window titled "mininet@mininet-vm: ~/work" with a file named "Makefile". The terminal has a dark background and light-colored text. The "Makefile" content is as follows:

```
GNU nano 4.8                                     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|
```

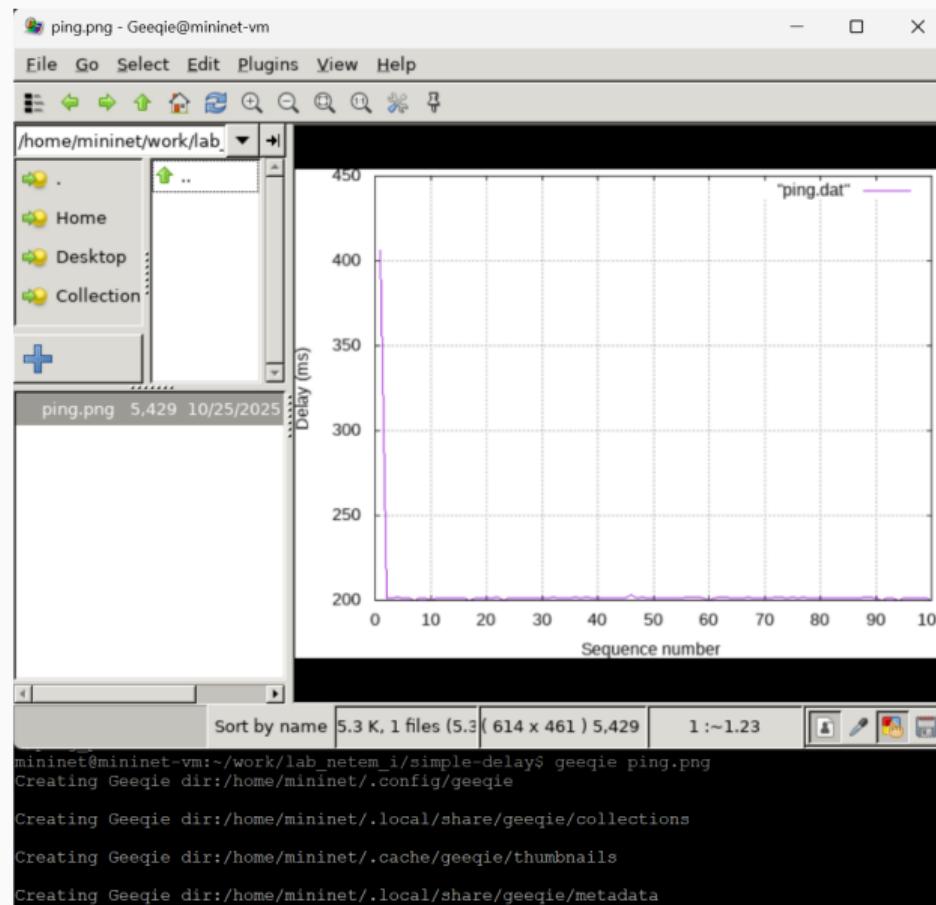
Рис. 16: Листинг Makefile

Запуск эксперимента

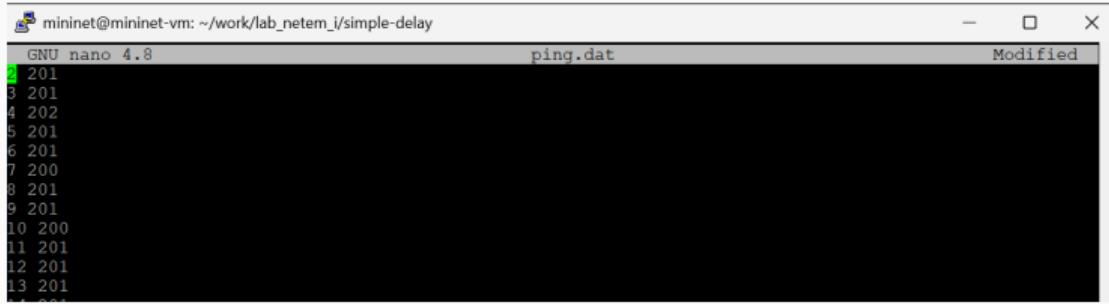
```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make
sudo python lab_netem_i.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
*** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
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
```

Рис. 17: Запуск эксперимента

Просмотр графика



Удаление первой строчки из файла ping.dat

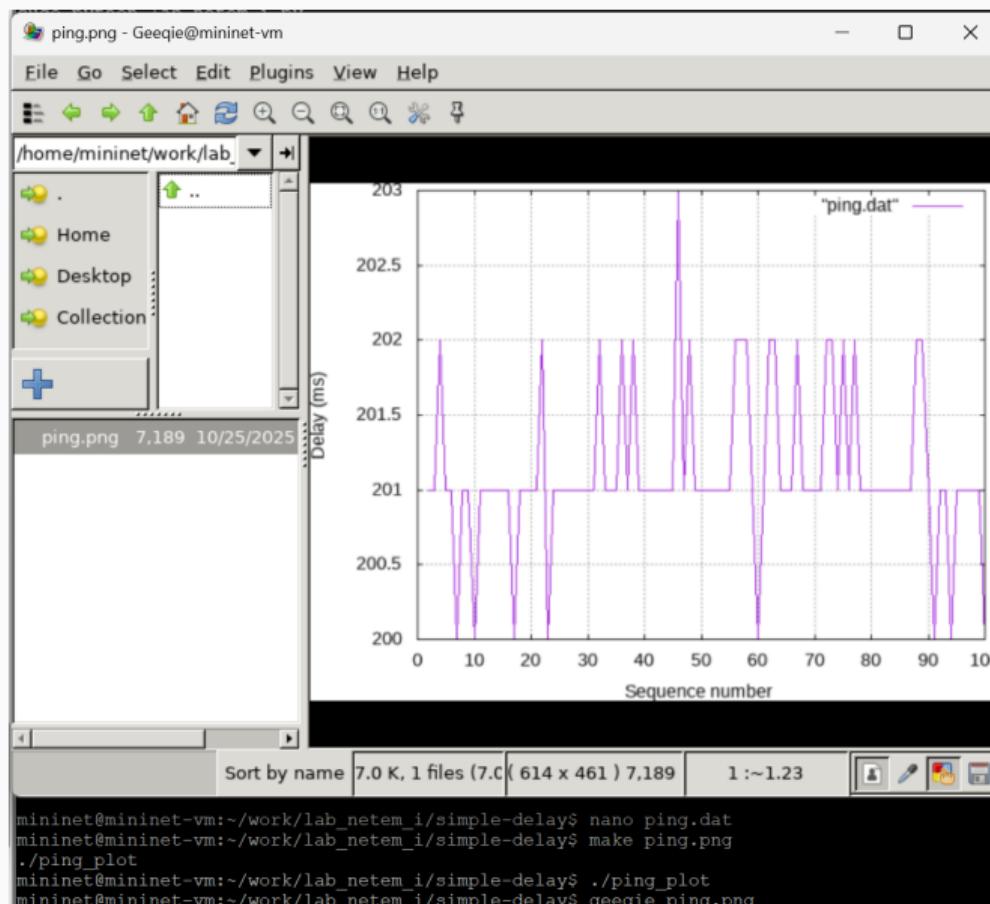


The screenshot shows a terminal window titled "mininet@mininet-vm: ~/work/lab_netem_i/simple-delay". Inside the terminal, the command "nano ping.dat" is run, opening a text editor window. The file "ping.dat" contains the following content:

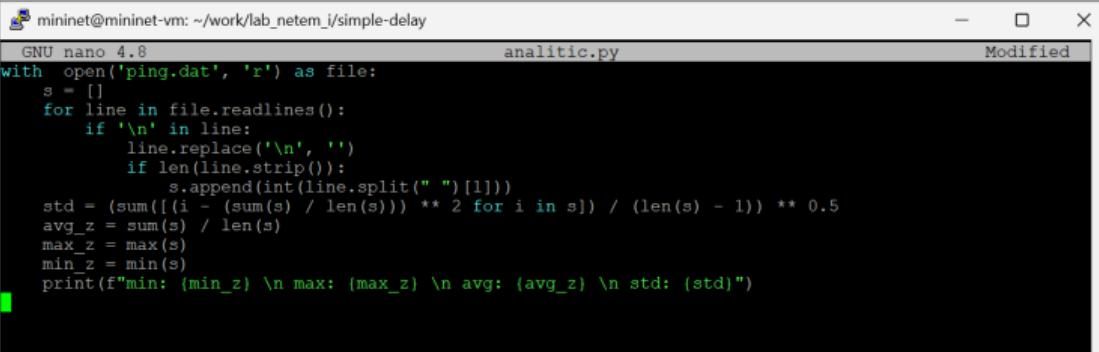
```
GNU nano 4.8          ping.dat          Modified
201
3 201
4 202
5 201
6 201
7 200
8 201
9 201
10 200
11 201
12 201
13 201
14 201
```

Рис. 19: Удаление первой строчки из файла ping.dat

Просмотр графика после удаления первой строки



Разработка скрипта для вычисления статистических данных



The screenshot shows a terminal window titled "mininet@mininet-vm: ~/work/lab_netem_i/simple-delay". The window contains a code editor for "analytic.py" which is a Python script. The script reads data from a file named "ping.dat" and calculates statistical values: minimum, maximum, average, and standard deviation. The code uses a for loop to read lines from the file, strip them, and split them by whitespace to extract numerical values. It then calculates the average and standard deviation of these values.

```
GNU nano 4.8                                analytic.py                                         Modified
with open('ping.dat', 'r') as file:
    s = []
    for line in file.readlines():
        if '\n' in line:
            line.replace('\n', '')
        if len(line.strip()):
            s.append(int(line.split(" ")[1]))
    std = (sum([(i - (sum(s) / len(s))) ** 2 for i in s]) / (len(s) - 1)) ** 0.5
    avg_z = sum(s) / len(s)
    max_z = max(s)
    min_z = min(s)
    print(f"min: {min_z} \n max: {max_z} \n avg: {avg_z} \n std: {std}")
```

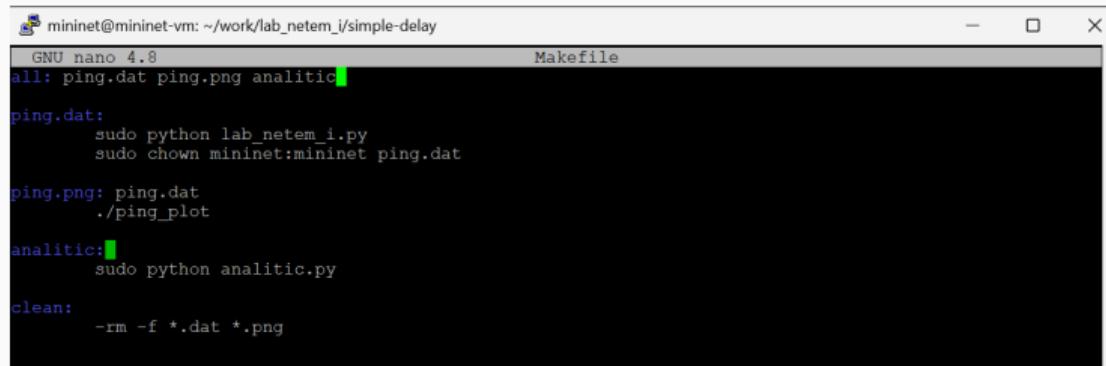
Рис. 21: Разработка скрипта для вычисления статистических данных

Тестирование разработанного скрипта

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nano analitic.py
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ sudo python analitic.py
min: 200
max: 203
avg: 201.12121212121212
std: 0.539703759874809
```

Рис. 22: Тестирование разработанного скрипта

Добавление правила запуска скрипта в Makefile



GNU nano 4.8 Makefile

```
all: ping.dat ping.png analitic

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

ping.png: ping.dat
    ./ping_plot

analitic:
    sudo python analitic.py

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

Рис. 23: Добавление правила запуска скрипта в Makefile

Очистка результатов эксперимента

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make clean
rm -f *.dat *.png
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ █
```

Рис. 24: Очистка результатов эксперимента

Листинг - Воспроизводимый эксперимент по изменению задержки

```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
GNU nano 4.8                               lab netem i.py
Modified

#!/usr/bin/env python

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

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

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

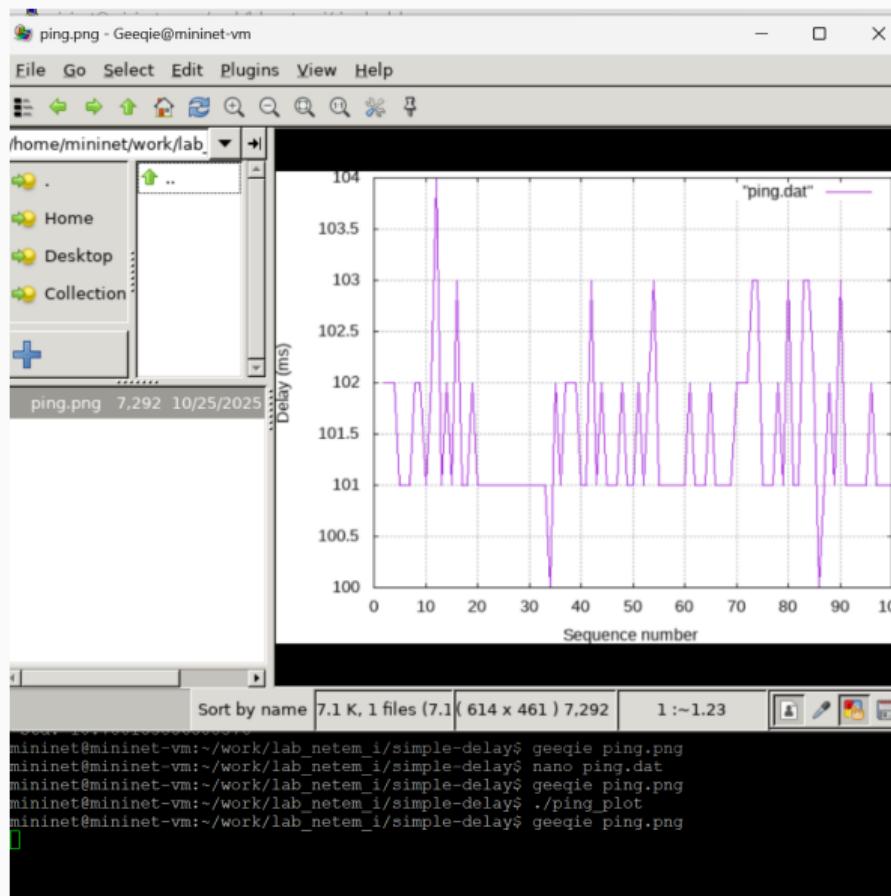
if __name__ == '__main__':
    setLogLevel('info')
    emptyNet()
```

Вывод - Воспроизводимый эксперимент по изменению задержки

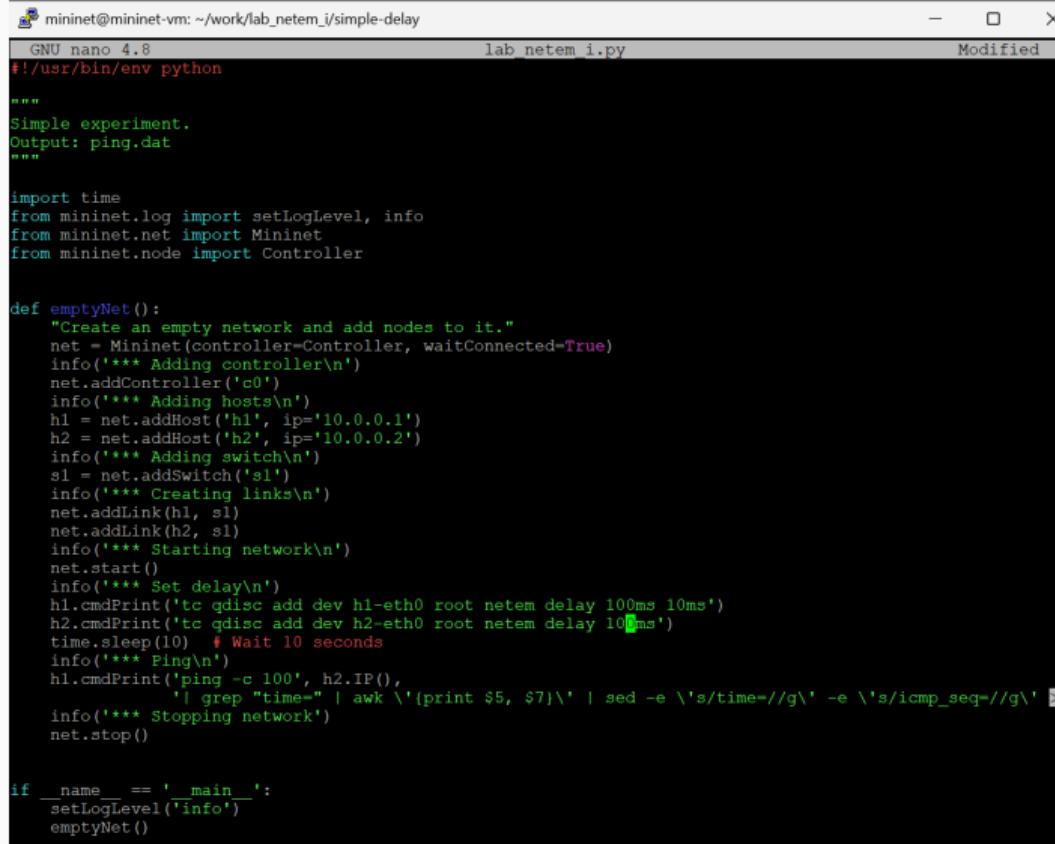
```
mininet@mininet-vm:~/work/lab_neterm_i/simple-delay$ nano lab_neterm_i.py
mininet@mininet-vm:~/work/lab_neterm_i/simple-delay$ make
udo python lab_neterm_i.py
** Adding controller
** Adding hosts
** Adding switch
** Creating links
** Starting network
** Configuring hosts
l h2
** Starting controller
:0
** Starting 1 switches
l ...
** Waiting for switches to connect
l
** 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, $7)\' | sed -e \'s/time=/g\' -e \'s/icmp_seq=/g\' > ping.dat')
** Stopping network*** Stopping 1 controllers
:0
** Stopping 2 links
.
** Stopping 1 switches
l
** Stopping 2 hosts
l h2
** Done
udo chown mininet:mininet ping.dat
/ping_plot
udo python analitic.py
in: 100
max: 209
avg: 102.52
std: 10.780153856508376
mininet@mininet-vm:~/work/lab_neterm_i/simple-delay$
```

Рис. 26: ВЫВОД|Воспроизводимый эксперимент по изменению задержки

График - Воспроизводимый эксперимент по изменению задержки



Листинг - Воспроизводимый эксперимент по изменению джиттера



The screenshot shows a terminal window titled "mininet@mininet-vm: ~/work/lab_netem_i/simple-delay". The window contains a Python script named "lab netem i.py". The script uses the mininet library to create a simple network with two hosts (h1, h2) and one switch (s1). It adds links between h1 and s1, and between h2 and s1. Then, it starts the network and sets traffic classes (qdisc) on both hosts to have a root queue with a delay of 100ms and a burst of 10ms. Finally, it performs a ping from h1 to h2 and stops the network.

```
GNU nano 4.8          lab netem i.py          Modified
#!/usr/bin/env python

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

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

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')
    h2.cmdPrint('tc qdisc add dev h2-eth0 root netem delay 10ms')
    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()

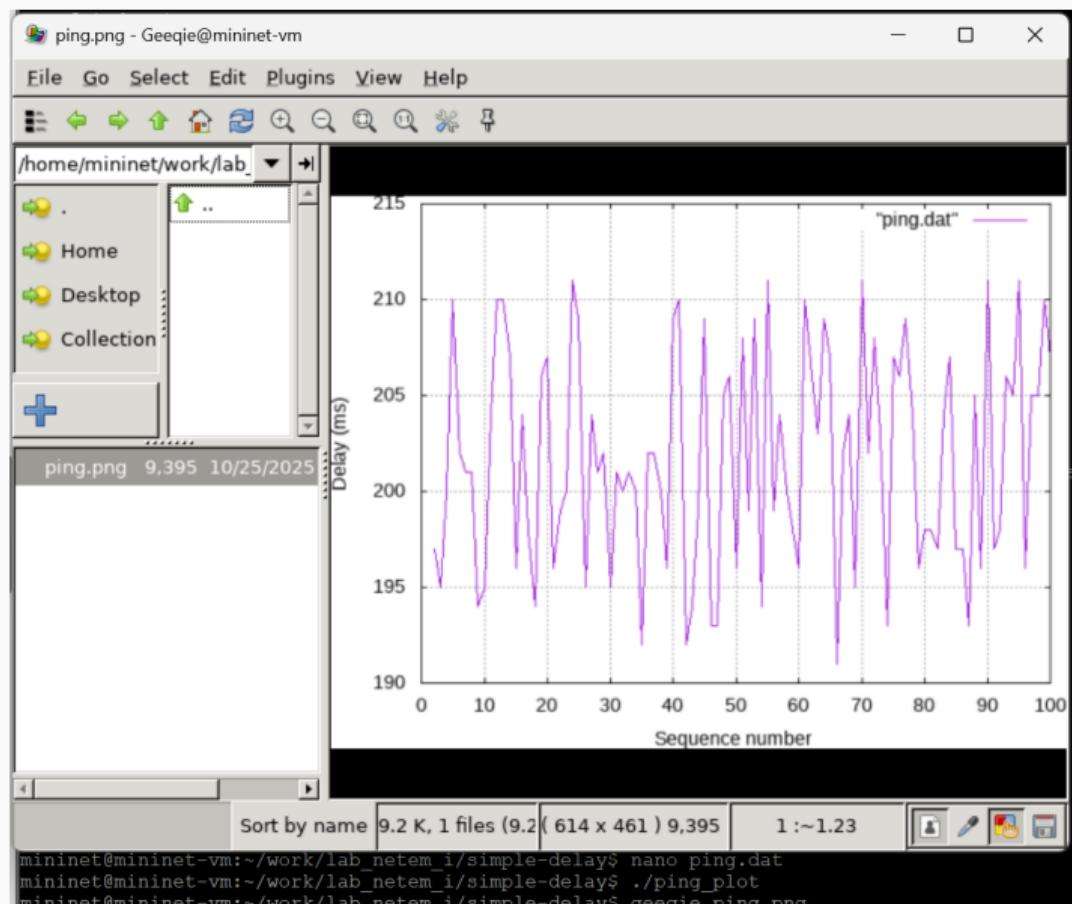
if __name__ == '__main__':
    setLogLevel('info')
    emptyNet()
```

Вывод - Воспроизводимый эксперимент по изменению джиттера

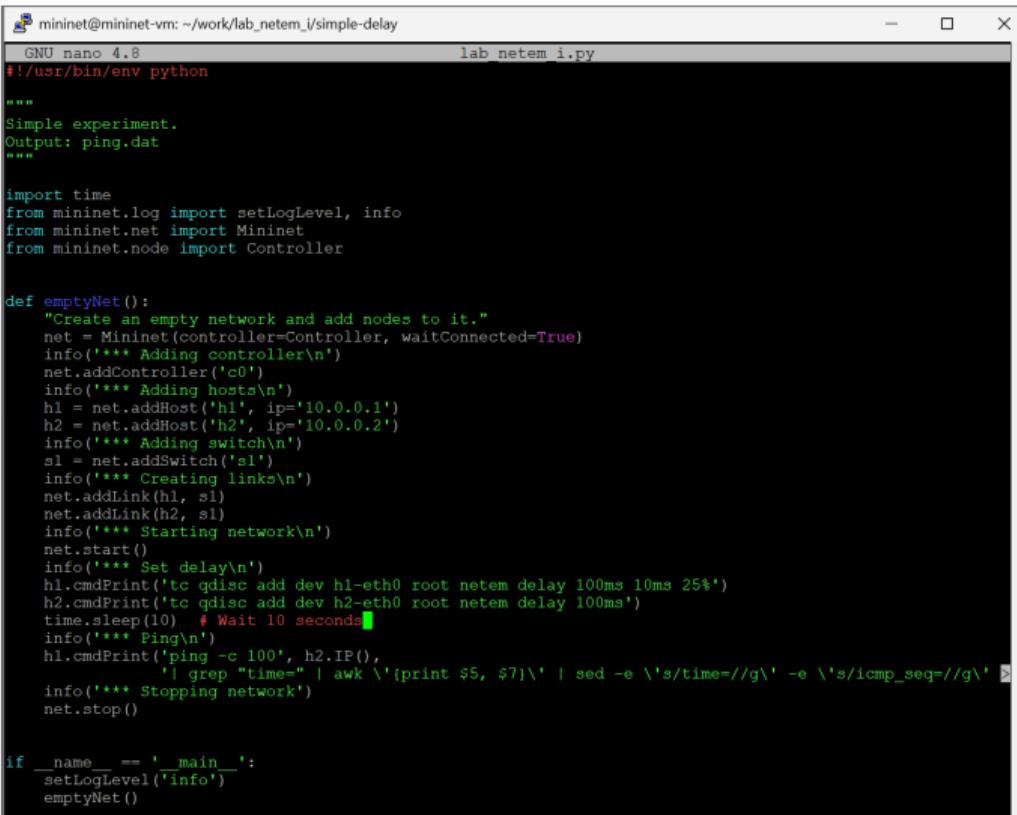
```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make clean
rm -f *.dat *.png
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nano lab_netem_i.py
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make
sudo python lab_netem_i.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** 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 10ms',)
*** 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, $7}\' | sed -e \'s/time=/g\' -e
\'s/icmp_seq=/g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
...
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
sudo python analitic.py
min: 191
max: 406
avg: 203.76
std: 21.194205421160365
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$
```

Рис. 29: ВЫВОД|Воспроизводимый эксперимент по изменению джиттера

График - Воспроизводимый эксперимент по изменению джиттера



Листинг - Воспроизводимый эксперимент по изменению значения корреляции для джиттера и задержки



```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
GNU nano 4.8                                     lab_netem_i.py

#!/usr/bin/env python

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

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

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

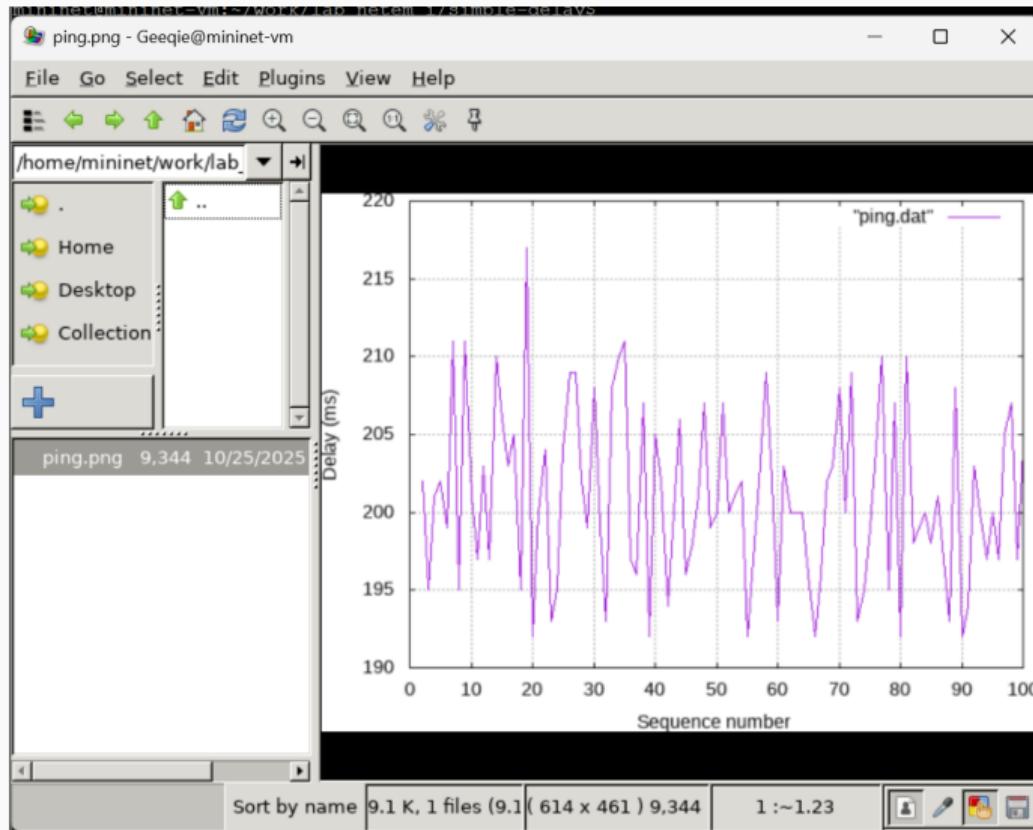
if __name__ == '__main__':
    setLogLevel('info')
    emptyNet()
```

Вывод - Воспроизводимый эксперимент по изменению значения корреляции для джиттера и задержки

```
mininet@mininet-vm:~/work/lab_neterm_i/simple-delay$ nano lab_neterm_i.py
mininet@mininet-vm:~/work/lab_neterm_i/simple-delay$ make
sudo python lab_neterm_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 10ms 25%',)
*** 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
sudo python analitic.py
min: 192
max: 403
avg: 202.98
std: 20.96075216477529
```

Рис. 32: ВЫВОД| Воспроизводимый эксперимент по изменению значения корреляции для джиттера и 35/42

График - Воспроизводимый эксперимент по изменению значения корреляции для джиттера и задержки



Листинг - Воспроизводимый эксперимент по изменению распределения времени задержки

```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
GNU nano 4.8                               lab netem_i.py                         Modified
#!/usr/bin/env python

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

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

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

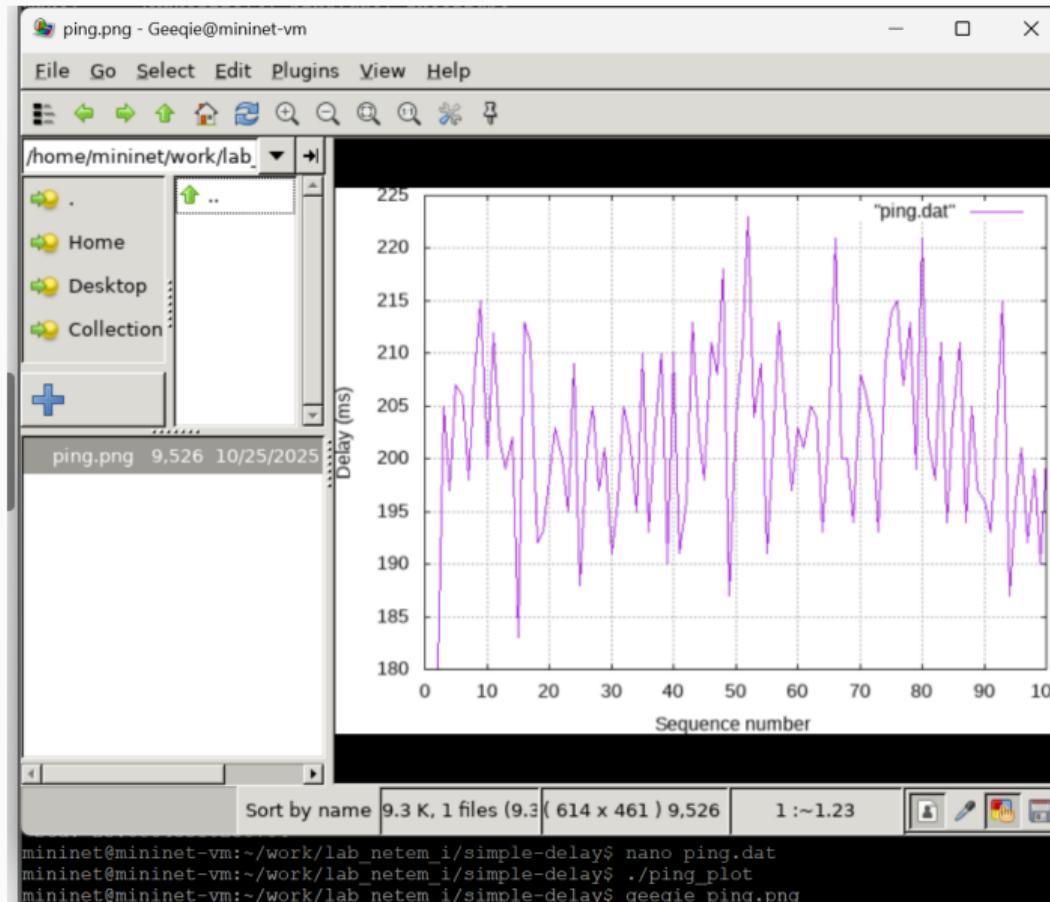
if __name__ == '__main__':
    setLogLevel('info')
    emptyNet()
```

Вывод - Воспроизводимый эксперимент по изменению распределения времени задержки

```
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 10ms 25% distribution normal',)
*** 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
sudo python analitic.py
min: 180
max: 417
avg: 204.21
std: 23.08845310250704
```

Рис. 35: ВЫВОД|Воспроизводимый эксперимент по изменению распределения времени задержки в эмулируемой глобальной сети

График - Воспроизводимый эксперимент по изменению распределения времени задержки



Создание папок под эксперименты

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ geeqie ping.png
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ cd ..
mininet@mininet-vm:~/work/lab_netem_i$ cp -R simple-delay correlation-delay
mininet@mininet-vm:~/work/lab_netem_i$ cp -R simple-delay jitter-delay
mininet@mininet-vm:~/work/lab_netem_i$ cp -R simple-delay change-delay
mininet@mininet-vm:~/work/lab_netem_i$ cp -R simple-delay change-jitter
mininet@mininet-vm:~/work/lab_netem_i$ cp -R simple-delay change-jitter-delay
mininet@mininet-vm:~/work/lab_netem_i$
```

Рис. 37: Создание папок под эксперименты

Выводы

Выводы

В результате выполнения данной лабораторной работы я познакомился с NETEM – инструментом для тестирования производительности приложений в виртуальной сети, а также получил навыки проведения интерактивного и воспроизводимого экспериментов по измерению задержки и её дрожания (jitter) в моделируемой сети в среде Mininet.

Список литературы

Список литературы

1. Mininet [Электронный ресурс]. Mininet Project Contributors. URL: <http://mininet.org/> (дата обращения: 06.10.2025).

