

# Лабораторная работа № 4

Эмуляция и измерение задержек в глобальных сетях

Оразгелдиев Язгелди

2025-10-25

# Содержание I

Информация

Вводная часть

# Информация

► Оразгелдиев Язгелди

# Докладчик

- ▶ Оразгелдиев Язгелди
- ▶ 1032225075@pfur.ru

# Докладчик

- ▶ Оразгелдиев Язгелди
- ▶ 1032225075@pfur.ru
- ▶ <https://github.com/YazgeldiOrazgeldiyev>

# Вводная часть

# Цели

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.



# Задание

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

## Задание

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

## Задание

1. Задайте простейшую топологию, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8.
2. Проведите интерактивные эксперименты по добавлению/изменению задержки, джиттера, значения корреляции для джиттера и задержки, распределения времени задержки в эмулируемой глобальной сети.
3. Реализуйте воспроизводимый эксперимент по заданию значения задержки в эмулируемой глобальной сети. Постройте график.

## Задание

1. Задайте простейшую топологию, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8.
2. Проведите интерактивные эксперименты по добавлению/изменению задержки, джиттера, значения корреляции для джиттера и задержки, распределения времени задержки в эмулируемой глобальной сети.
3. Реализуйте воспроизводимый эксперимент по заданию значения задержки в эмулируемой глобальной сети. Постройте график.
4. Самостоятельно реализуйте воспроизводимые эксперименты по изменению задержки, джиттера, значения корреляции для джиттера и задержки, распределения времени задержки в эмулируемой глобальной сети. Постройте графики.

# Содержание исследования

```
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  71648ffa79f761f9b729315fc22cc37c
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth add mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  71648ffa7
9f761f9b729315fc22cc37c
root@mininet-vm:~# logout
```

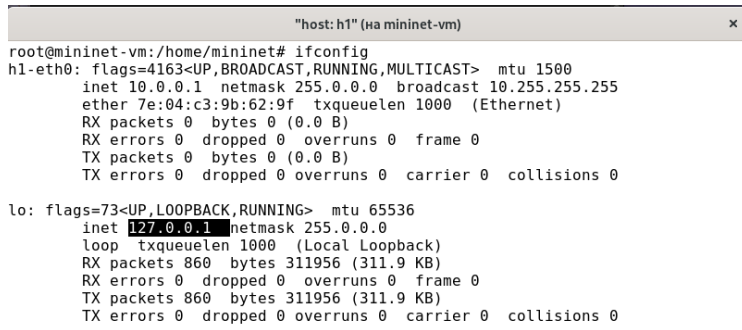
Рисунок 1: *Исправление права запуска X-соединения*

# Содержание исследования

```
mininet@mininet-vm:~$ sudo mn --topo=single,2 -x
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Running terms on localhost:10.0
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

Рисунок 2: Создание простейшей топологии

# Содержание исследования



```
root@mininet-vm:/home/mininet# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.0.0.1  netmask 255.0.0.0  broadcast 10.255.255.255
    ether 7e:04:c3:9b:62:9f  txqueuelen 1000  (Ethernet)
    RX packets 0  bytes 0 (0.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 0  bytes 0 (0.0 B)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    loop  txqueuelen 1000  (Local Loopback)
    RX packets 860  bytes 311956 (311.9 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 860  bytes 311956 (311.9 KB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Рисунок 3: Команда *ifconfig* на хосте *h1*

# Содержание исследования

```
"host: h2" (на mininet-vm) x
root@mininet-vm:/home/mininet# ifconfig
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.2 netmask 255.0.0.0 broadcast 10.255.255.255
    ether de:e7:91:c3:cb:54 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 1027 bytes 325680 (325.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1027 bytes 325680 (325.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Рисунок 4: Команда *ifconfig* на хосте *h2*



# Содержание исследования

```
root@mininet-vm:/home/mininet# ping -c 6 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.30 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.142 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.030 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.075 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.072 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.035 ms
```

Рисунок 5: Команда *ping*

# Содержание исследования

```
root@mininet-vm:/home/mininet# ping -c 6 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.813 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.052 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.042 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.082 ms
```

Рисунок 6: *Команда ping*

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

# Содержание исследования

```
root@mininet-vm:/home/mininet# ping -c 6 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=100 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=101 ms
```

Рисунок 7: Команда *ping*

# Содержание исследования

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

Рисунок 8: *Добавление задержки на хосте h2*

# Содержание исследования

```
root@mininet-vm:/home/mininet# ping -c 6 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=201 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=201 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=201 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=201 ms
```

Рисунок 9: Команда *ping*

## 1## Содержание исследования

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

Рисунок 10: Изменение задержки

# Содержание исследования

```
root@mininet-vm:/home/mininet# ping -c 6 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=102 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=102 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=100 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=102 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=100 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=100 ms
```

Рисунок 11: Команда *ping*

# Содержание исследования

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

Рисунок 12: *Удаление правил*

# Содержание исследования

```
root@mininet-vm:/home/mininet# ping -c 6 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.11 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.971 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.114 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.033 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.034 ms
```

Рисунок 13: Команда *ping*



# Содержание исследования

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

Рисунок 14: *Добавление задержки со случайным отклонением*

# Содержание исследования

```
root@mininet-vm:/home/mininet# ping -c 6 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=111 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=94.0 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=95.1 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=106 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=109 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=99.7 ms
```

Рисунок 15: Команда *ping*

# Содержание исследования

```
root@mininet-vm:/home/mininet# sudo tc qdisc del dev hl-eth0 root netem
root@mininet-vm:/home/mininet# ping -c 6 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.239 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.035 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.058 ms
^C
```

Рисунок 16: *Удаление правил*

# Содержание исследования

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

Рисунок 17: *Добавление задержки со случайным отклонением и корреляцией*

# Содержание исследования

```
root@mininet-vm:/home/mininet# ping -c 6 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=93.5 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=92.9 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=91.5 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=103 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=109 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=93.9 ms
```

Рисунок 18: Команда *ping*

# Содержание исследования

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

Рисунок 19: *Добавление нормального распределения*

# Содержание исследования

```
root@mininet-vm:/home/mininet# ping -c 6 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=79.7 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=93.3 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=135 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=121 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=75.8 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=133 ms
```

Рисунок 20: Команда *ping*

## Содержание исследования

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ sudo apt-get update
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [128 kB]
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [128 kB]
Fetched 383 kB in 1s (285 kB/s)
Reading package lists... Done
```

Рисунок 21: Обновление репозитория программного обеспечения



# Содержание исследования

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

Рисунок 22: Установка пакета *geeqie*

# Содержание исследования

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

Рисунок 23: *Создание каталога*

# Содержание исследования

```
mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_i/simple-delay  
mininet@mininet-vm:~$ cd ~/work/lab_netem_i/simple-delay  
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$
```

Рисунок 24: *Создание подкаталога*

# Содержание исследования

```
GNU nano 4.8 /home/mininet/work/lab_netem_i/simple-delay/lab_netem_i
#!/usr/bin/env python

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

def emptyNet():
    "Create an empty network and add nodes to it."
    net = Mininet( controller=Controller, waitConnected=True )

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

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

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

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

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

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

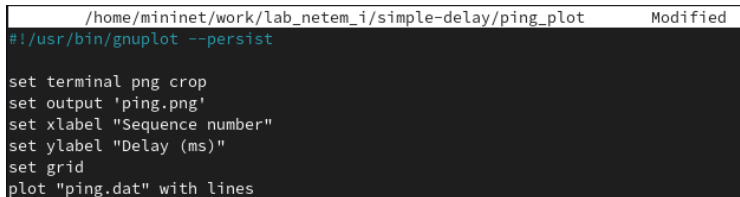
    time.sleep(10) # Wait 10 seconds

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

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

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

# Содержание исследования



```
/home/mininet/work/lab_netem_i/simple-delay/ping_plot Modified
#!/usr/bin/gnuplot --persist

set terminal png crop
set output 'ping.png'
set xlabel "Sequence number"
set ylabel "Delay (ms)"
set grid
plot "ping.dat" with lines
```

Рисунок 26: *Создание скрипта ping\_plot*

# Содержание исследования

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

Рисунок 27: *Изменение прав доступа*

# Содержание исследования

```
GNU nano 4.8 /home/mininet/work/lab_netem_i/simple-delay/Makefile
all: ping.dat ping.png

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

ping.png: ping.dat
    ./ping_plot

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

Рисунок 28: *Создание Makefile*

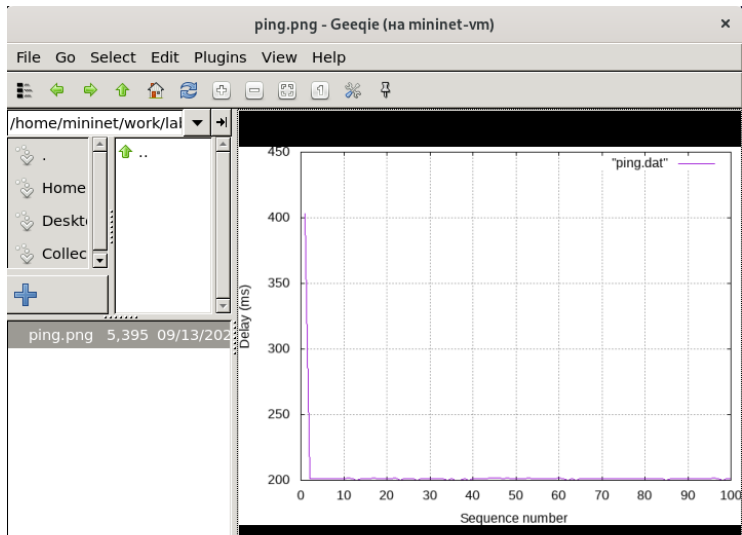
# Содержание исследования

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make
sudo python lab_netem_i.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
*** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
*** Set delay
*** h1 : ('tc qdisc add dev h1-eth0 root netem delay 100ms',)
*** h2 : ('tc qdisc add dev h2-eth0 root netem delay 100ms',)
*** Ping
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "time=" | awk \'{print $5, $7}\'' | sed -e \\'s/time=//g\' -e \\'s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$
```

Рисунок 29: *Выполнение эксперимента*



# Содержание исследования

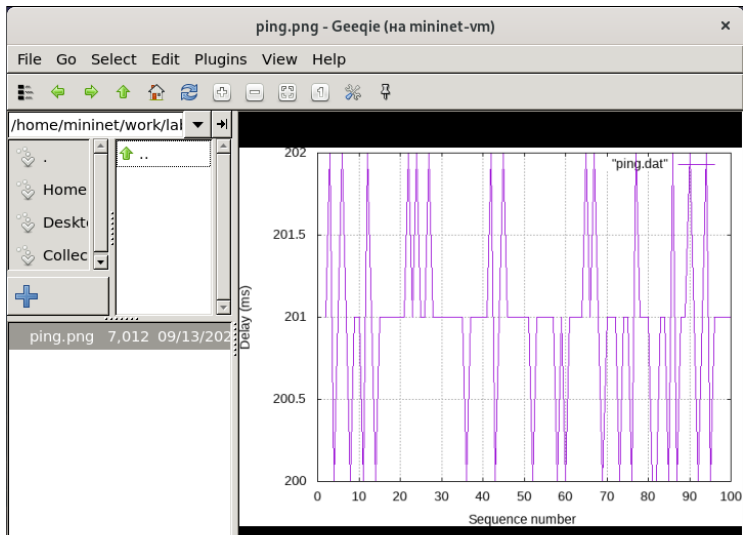


# Содержание исследования

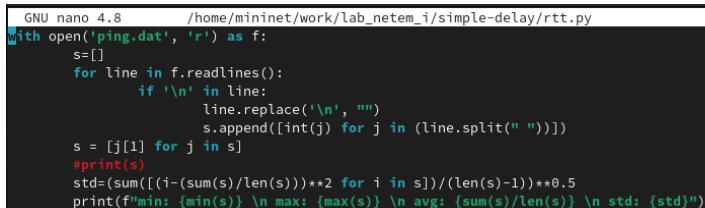
```
GNU nano 4.8 /home/mininet/work/lab_netem_i/simple-delay/ping.dat
2 201
3 202
4 200
5 201
6 202
7 201
8 200
9 201
10 201
11 200
12 202
13 201
14 200
15 201
16 201
17 201
18 201
19 201
```

Рисунок 31: Удаление строки

# Содержание исследования



## Содержание исследования



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

Рисунок 33: Скрипт для вычисления данных

## Содержание исследования

```
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
```

```
rtt:
```

```
    sudo python rtt.py
```

# Содержание исследования

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make rtt
sudo python rtt.py
min: 200
max: 202
avg: 200.95959595959596
std: 0.5699837433985384
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make clean
rm -f *.dat *.png
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ ls
lab_netem_i.py  Makefile  ping_plot  rtt.py
```

Рисунок 35: Результат работы скрипта

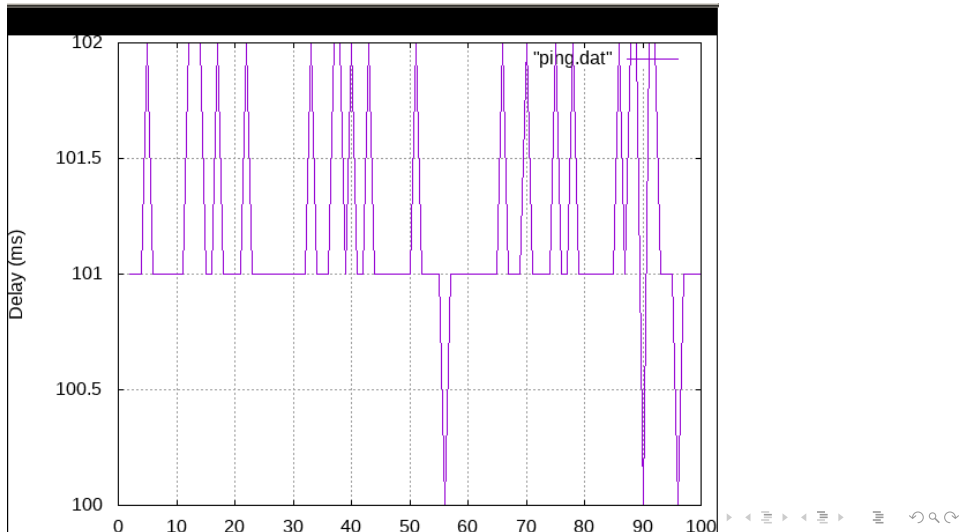
# Содержание исследования

```
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' )
```

Рисунок 36: Изменение файла *lab\_netem\_i.py*

# Содержание исследования





# Содержание исследования

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make rtt
sudo python rtt.py
min: 100
max: 102
avg: 101.181818181819
std: 0.45989431713313733
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$
```

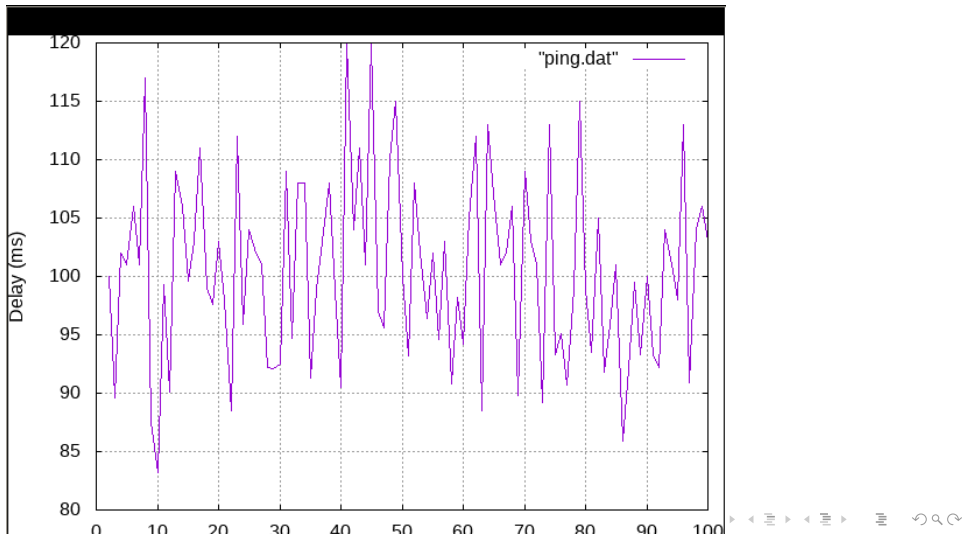
Рисунок 38: *Вычисленные значения*

# Содержание исследования

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

Рисунок 39: *Изменение файла lab\_netem\_i.py*

## Содержание исследования



# Содержание исследования

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

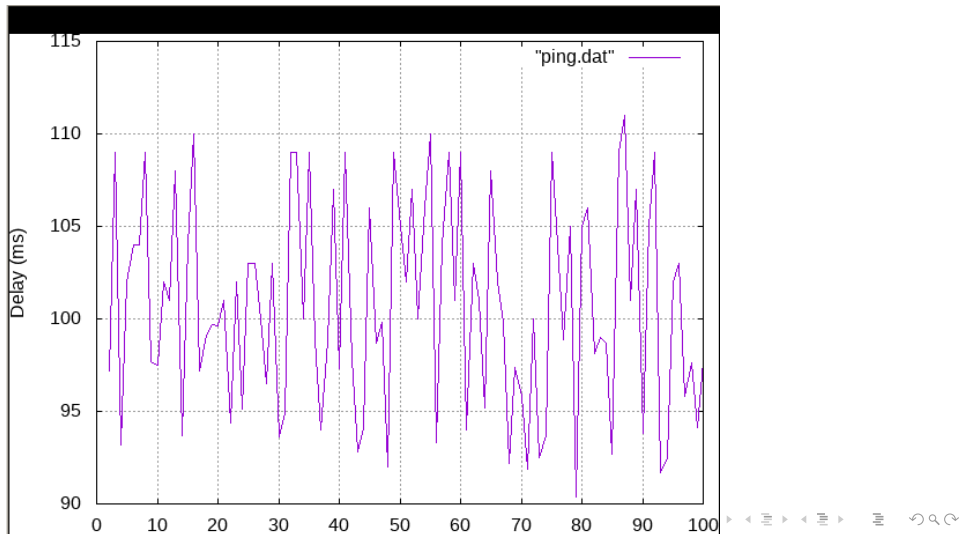
Рисунок 41: *Вычисленные значения*

# Содержание исследования

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

Рисунок 42: Изменение файла *lab\_netem\_i.py*

# Содержание исследования



# Содержание исследования

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

Рисунок 44: *Вычисленные значения*

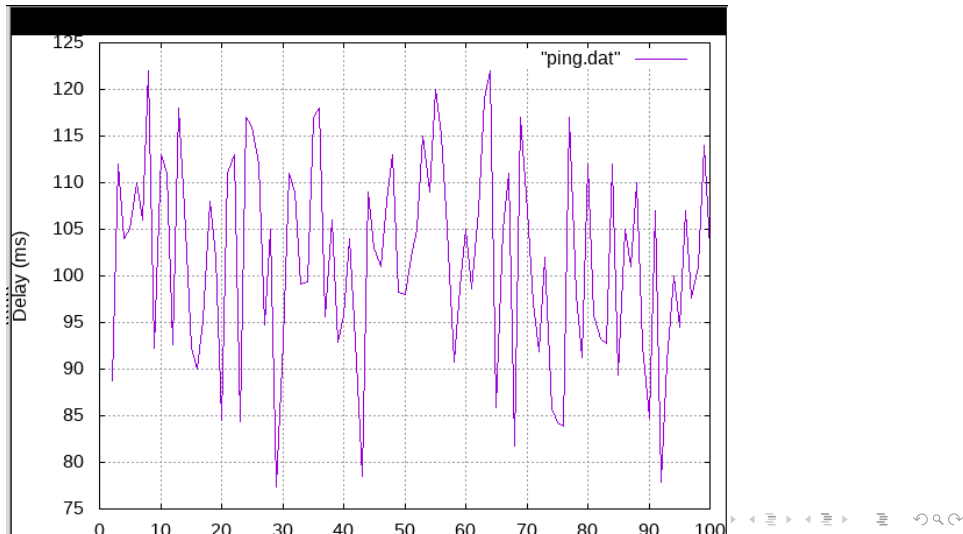
# Содержание исследования

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

Рисунок 45: Изменение файла *lab\_netem\_i.py*



# Содержание исследования



# Содержание исследования

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

Рисунок 47: *Вычисленные значения*

# Результаты

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