



# Лабораторная работа N°4

по дисциплине Моделирование сетей передачи данных

# Ход работы

The screenshot displays a terminal window within a Mininet VM environment. The user is at the root prompt and has executed the `ifconfig` command to show the configuration of the `h2-eth0` and `lo` interfaces. The `h2-eth0` interface is configured with IP `10.0.0.2`, netmask `255.0.0.0`, and broadcast `10.255.0.0`. The `lo` interface is configured with IP `127.0.0.1` and netmask `255.0.0.0`. Both interfaces have a MTU of 65536 and are in the `UP` state. The terminal also shows the output of a `ping -c 6 10.0.0.2` command, which successfully reaches the destination with 0% packet loss and a round-trip time of approximately 1.029 ms. The terminal window is titled `mininet@mininet-vm` and is part of a larger desktop environment with a taskbar at the bottom.

```
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 10.255.0.0
    ether fe:fe:8d:be:6d:9b 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 1535 bytes 345088 (345.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1535 bytes 345088 (345.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

```
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=1.03 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.094 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.019 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.023 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.026 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.018 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 510ms
rtt min/avg/max/mdev = 0.018/0.201/1.029/0.371 ms
root@mininet-vm:/home/mininet#
```



## Ход работы

```
-- 10.0.0.2 ping statistics ---
  packets transmitted, 6 received, 0% packet loss, time 5009ms
  tt min/avg/max/mdev = 100.035/100.294/100.602/0.186 ms
oot@mininet-vm:/home/mininet# ping -c 6 10.0.0.2
ING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
4 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=201 ms
4 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=201 ms
4 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=200 ms
4 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=200 ms
4 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=201 ms
4 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=201 ms
```

```
..
```

# Ход работы

```
root@mininet-virtual-machine:~# sudo tc qdisc change dev h1-eth0 root netem delay 50ms
root@mininet-virtual-machine:~# 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=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 5010ms
rtt min/avg/max/mdev = 100.396/100.727/101.053/0.217 ms
root@mininet-virtual-machine:~# sudo tc qdisc del dev h1-eth0 root netem
root@mininet-virtual-machine:~# 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=0.410 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.205 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.066 ms
```

Для получателя h2:

```
sudo tc qdisc del dev h2-eth0 root netem
```

2. Проверьте, что соединение между хостом h1 и хостом h2 не имеет явно установленной задержки, используя команду `ping` с параметром `-c 6` с терминала хоста h1. Укажите в отчете минимальное, среднее, максимальное и стандартное отклонение времени приёма-передачи (RTT).

#### 4.4.2.4. Добавление значения дрожания задержки в интерфейс подключения к эмулируемой глобальной сети

В сетях нет постоянной задержки. Она может варьироваться в зависимости от других потоков трафика, конкурирующих за тот же путь. Джиттер (jitter) — это изменение времени задержки. Параметры задержки описываются в терминах теории вероятностей средним значением  $\mu$ , стандартным отклонением  $\sigma$  и корреляцией. По умолчанию NETEM использует равномерное распределение,

так что задержка находится в пределах  $\mu \pm \sigma$ . По отношению между последовательными псевдослучайными числами.

1. При необходимости восстановите конфигурацию на узлах h1 и h2:

```
sudo tc qdisc del dev h1-eth0 root netem
```

```
sudo tc qdisc del dev h2-eth0 root netem
```

# Ход работы

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 1
00ms 20ms distribution normal
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=257 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=112 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=135 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=93.5 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=82.2 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=80.4 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=123 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=101 ms

--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9014ms
rtt min/avg/max/mdev = 80.379/119.694/257.188/48.558 ms
root@mininet-vm:/home/mininet# clear
```

нормальное.

1. При необходимости восстановите конфигурацию интерфейсов по умолчанию на узлах h1 и h2.
2. Задайте нормальное распределение задержки на узле h1 в эмулируемой сети:

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

3. Убедитесь, что все пакеты, покидающие хост h1 на интерфейсе h1-eth0, будут иметь время задержки, которое распределено в диапазоне  $100 \text{ ms} \pm 20 \text{ ms}$ . Используйте для этого команду `ping` на терминале хоста h1 с параметром `-c 10`.
4. Восстановите конфигурацию интерфейса по умолчанию на узле h1.
5. Завершите работу mininet в интерактивном режиме, введя в интерфейс mininet:

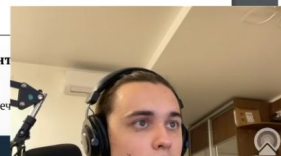
```
mininet> exit
```

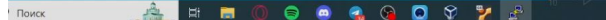
36

## 4.4.3. Воспроизведение эксперимента

### 4.4.3.1. Предварительная подготовка

1. Обновите репозитории программного обеспечения





#### 4.4.3.1. Предварительная подготовка

#### 4.4.3.1. Предварительная подготовка

- ```
1 sudo apt-get update
```

2. Установите пакет `geeqie` — понадобится для просмотра файлов `png`:

- ```
sudo apt install geeqie
```

3. Для каждого воспроизводимого эксперимента `exname` создайте свой каталог, в котором будут размещаться файлы эксперимента:

- ```
1 mkdir -p ~/work/lab_netem_i/expname
```

- Здесь `expname` может принимать значения `simple-delay`, `change-delay`, `jitter-delay`, `correlation-delay` и т.п.
4. Для каждого случая создайте скрипт для проведения эксперимента `lab_netem_1.py` и скрипт для визуализации результатов `ping_plot1`.

#### 4.4.3.2. Добавление задержки для интерфейса, подключающегося к эмулируемой глобальной сети

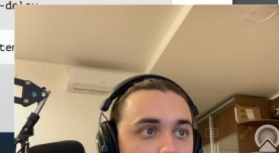
С помощью API Mininet воспроизведите эксперимент по добавлению задержки для интерфейса хоста, подключающегося к эмулируемой глобальной сети.

1. В виртуальной среде mininet в своём рабочем каталоге с проектами создайте каталог `simple-delay` и перейдите в него:

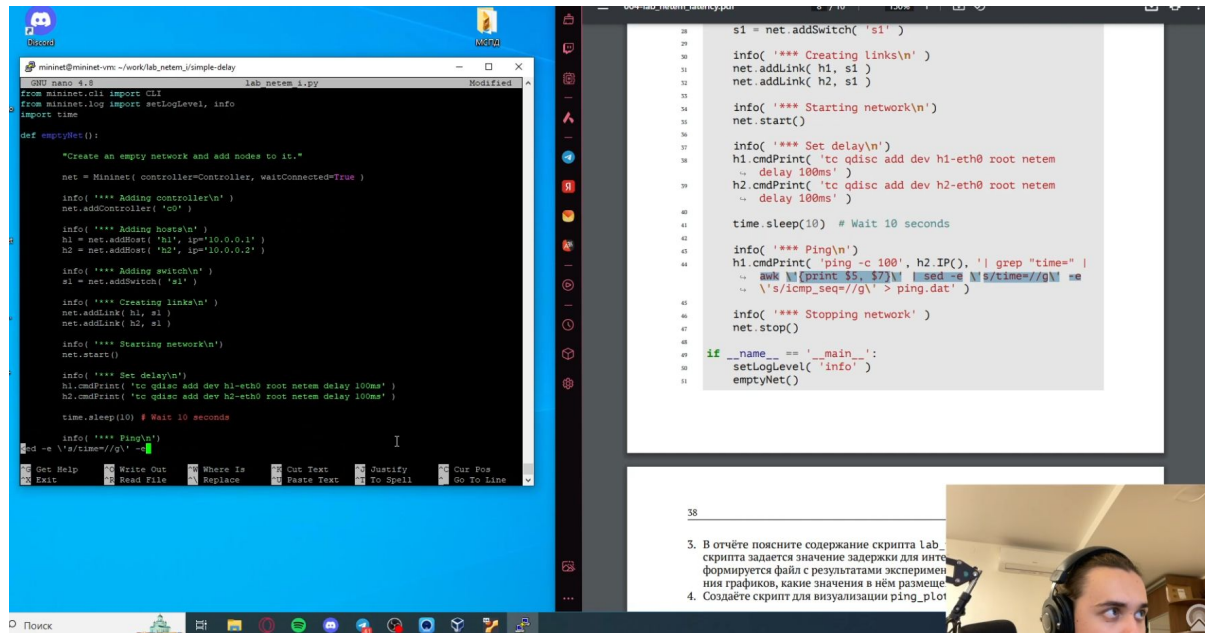
- ```
1 mkdir -p ~/work/lab_netem_i/simple-delay
2 cd ~/work/lab_netem_i/simple-delay
```

2. Создаёте скрипт для эксперимента `lab_neter`

- ```
1 #!/usr/bin/env python
2
3 """
4 Simple experiment.
5 Output: ping.dat
```



# Ход работы



```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
GNU nano 4.8 lab_netem_i.py
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.cmd( 'ping -c 100' )
    h2.cmd( 'ping -c 100' )

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

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

```
38 s1 = net.addSwitch( 's1' )
39
40 info( '*** Creating links\n' )
41 net.addLink( h1, s1 )
42 net.addLink( h2, s1 )
43
44 info( '*** Starting network\n' )
45 net.start()
46
47 info( '*** Set delay\n' )
48 h1.cmdPrint( 'tc qdisc add dev h1-eth0 root netem
49   delay 100ms' )
50 h2.cmdPrint( 'tc qdisc add dev h2-eth0 root netem
51   delay 100ms' )
52
53 time.sleep(10) # Wait 10 seconds
54
55 info( '*** Ping\n' )
56 h1.cmdPrint( 'ping -c 100', h2.IP(). ' | grep "time=" |
57   awk {print $5, $7} | sed -e s/time=//g | se
58   \sicmp_seq=//g\ > ping.dat' )
59
60 info( '*** Stopping network\n' )
61 net.stop()
62
63 if __name__ == '__main__':
64     setLogLevel( 'info' )
65     emptyNet()
```

38

3. В отчёте поясните содержание скрипта lab\_...  
скрипта задается значение задержки для инте...  
формируется файл с результатами эксперимен...  
ния графиков, какие значения в нём разме...  
4. Создайте скрипт для визуализации ping\_plot

# Ход работы

```
#!/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
```

ИСПД\_лаб4\_выполнение

38 Лабораторная работа №4

3. В отчёте поясните содержание скрипта lab\_netem\_i.py. В каких строках скрипта задается значение задержки для интерфейса хоста? Каким образом формируется файл с результатами эксперимента для последующего построения графиков, какие значения в нём размещены?

4. Создайте скрипт для визуализации ping\_plot результатов эксперимента:

```
#!/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
```

5. Задайте права доступа к файлу скрипта:

```
chmod +x ping_plot
```

6. Создайте Makefile для управления процессом проведения эксперимента:

```
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 *.p
```

7. Выполните эксперимент:

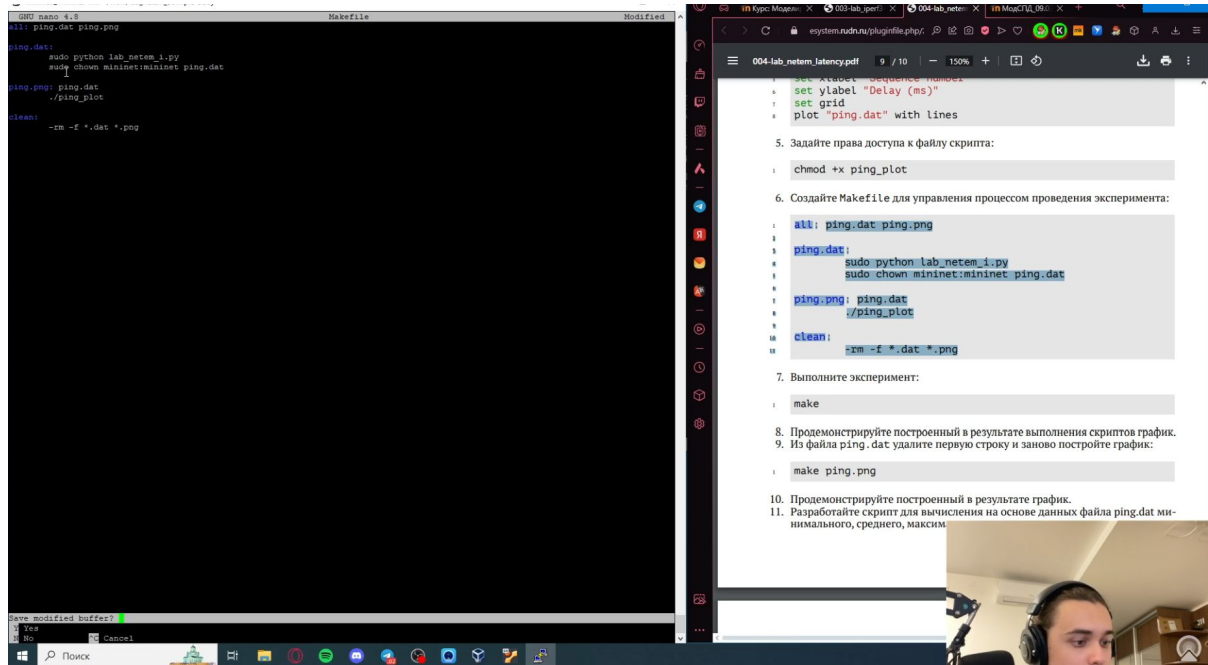
```
make
```

8. Продемонстрируйте построение

9. Изобразите график задержки



# Ход работы



The screenshot displays a computer screen with two main windows. On the left is a terminal window titled 'Makefile' showing the following content:

```
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
```

At the bottom of the terminal, a prompt asks 'Save modified buffers?' with 'Yes' and 'No' options.

On the right is a web browser window displaying a PDF document titled '004-lab\_netem\_latency.pdf'. The document contains a list of instructions for a lab experiment, including steps for setting permissions, creating a Makefile, and running the experiment. The instructions are as follows:

5. Задайте права доступа к файлу скрипта:  
`chmod +x ping_plot`
6. Создайте Makefile для управления процессом проведения эксперимента:  

```
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
```
7. Выполните эксперимент:  
`make`
8. Продемонстрируйте построенный в результате выполнения скриптов график.
9. Из файла ping.dat удалите первую строку и заново постройте график:  
`make ping.png`
10. Продемонстрируйте построенный в результате график.
11. Разработайте скрипт для вычисления на основе данных файла ping.dat минимального, среднего, максим

In the bottom right corner of the browser window, there is a small video feed showing a person wearing headphones.

# Ход работы

```
mininet@mininet-vml:/work/lab_netem_1/simple-delay$ make ping.png
sudo python lab_netem_1.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
*** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
SD
*** Starting 1 switches
h1 ...
*** Waiting for switches to connect
h1
*** 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', 'j grep "time=" | awk '{print $5, $7}' | sed -e 's/time=//g' -e 's/icmp_seq=//g' > ping.da
17)
h1
*** Stopping network*** Stopping 1 controllers
SD
*** Stopping 2 links
...
*** Stopping 1 switches
h1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chmod mininet:mininet ping.dat
./ping_plot
mininet@mininet-vml:/work/lab_netem_1/simple-delay$ ls
lab_netem_1.py Makefile ping.dat ping_plot ping.png
mininet@mininet-vml:/work/lab_netem_1/simple-delay$ nano ping.dat
```

```
make
```

8. Продемонстрируйте построенный в результате выполнения скриптов гр  
9. Из файла ping.dat удалите первую строку и заново постройте график

```
make ping.png
```

10. Продемонстрируйте построенный в результате график.  
11. Разработайте скрипт для вычисления на основе данных файла ping.da  
нимального, среднего, максимального и стандартного отклонения вре

Королюков А. В., Куликов Д. С. Математическая обработка данных

приёма-передачи. Де  
стрируйте работу ск  
файла.

12. Очистите каталог от

```
make clean
```

# Ход работы

The screenshot displays a network simulation environment. The main window shows a graph titled "ping.dat" with the y-axis labeled "Delay (ms)" ranging from 200 to 450 and the x-axis labeled "Sequence number" ranging from 0 to 100. The graph shows a sharp initial spike in delay followed by a stable, low delay. Below the graph is a terminal window showing commands and output for running a ping test and generating a plot. To the right, a web browser displays a document with instructions in Russian, including steps for running experiments and cleaning up. The bottom right corner shows a video feed of a person.

ping.png - Geeqie@mininet-vm

File Go Select Edit Plugins View Help

/home/mininet/work/lab...

ping.png 5.389 12/06/2024

Delay (ms)

Sequence number

"ping.dat"

Stopping a trace

h2

do chown mininet:mininet ping.dat

/ping\_plot

mininet@mininet-vm:~/work/lab\_netem\_1/simple-delay\$

mininet@mininet-vm:~/work/lab\_netem\_1/simple-delay\$ ls

lab\_netem\_1.py Makefile ping.dat ping\_plot ping.png

mininet@mininet-vm:~/work/lab\_netem\_1/simple-delay\$ make ping.dat

mininet@mininet-vm:~/work/lab\_netem\_1/simple-delay\$ ping.png

ping.png: command not found

mininet@mininet-vm:~/work/lab\_netem\_1/simple-delay\$ make

clean:

rm -f \*.dat \*.png

7. Выполните эксперимент:

make

8. Продемонстрируйте построенный в результате выполнения скрипта гр

9. Из файла ping.dat удалите первую строку и заново постройте график

make ping.png

10. Продемонстрируйте построенный в результате график.

11. Разработайте скрипт для вычисления на основе данных файла ping.dat минимального, среднего, максимального и стандартного отклонения вре

Королюкова А. В., Кузнецов Д. С. Моделирование сетей передачи данных

приёма-передачи. Добавьте правило запуска скрипта в Makefile. Продемонстрируйте работу скрипта с выводом значений на экран или в отдельный файл.

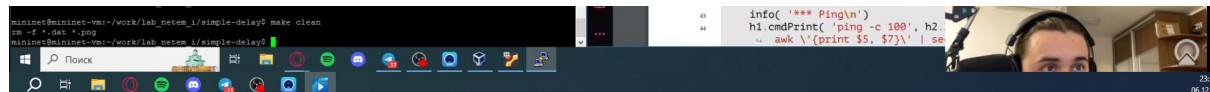
12. Очистите каталог от результатов проведения экспериментов:

make clean

4.4.3.3. Задание для самостоятельного решения

Самостоятельно реализуйте алгоритм вычисления задержки, джиттера, значения времени задержки. Вычислите минимальное, среднее, максимальное и стандартное отклонение времени приёма-передачи.

# Ход работы





## Вывод

Я ознакомился 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**.