

Лабораторная Работа №3

Измерение и тестирование пропускной способности сети.
Воспроизводимый эксперимент.

Козлов В.П.

Российский университет дружбы народов им. Патриса Лумумбы, Москва, Россия

- Козлов Всеволод Павлович
- НФИбд-02-22
- Российский университет дружбы народов
- [1132226428@pfur.ru]

Выполнение лабораторной работы

Основной целью работы является знакомство с инструментом для измерения пропускной способности сети в режиме реального времени — iPerf3, а также получение навыков проведения воспроизводимого эксперимента по измерению пропускной способности моделируемой сети в среде Mininet.

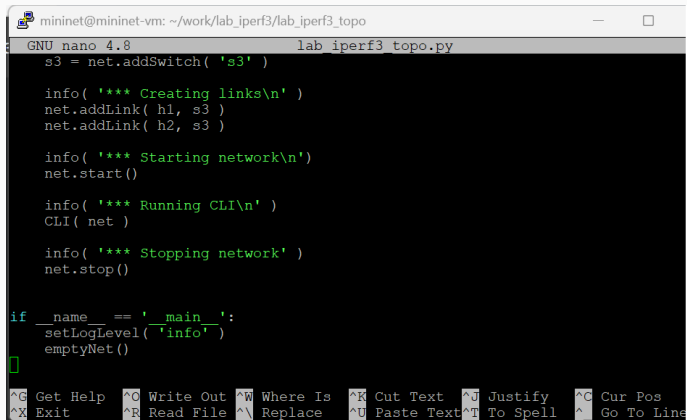
1. Воспроизвести посредством API Mininet эксперименты по измерению пропускной способности с помощью iPerf3.
2. Построить графики по проведённому эксперименту

Создал подкаталог с примером скрипта

```
mininet@mininet-vm:~$ cd ~/work/lab_iperf3
mininet@mininet-vm:~/work/lab_iperf3$ mkdir lab_iperf3_topo
mininet@mininet-vm:~/work/lab_iperf3$ cd ~/work/lab_iperf3/lab_iperf3_topo
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ cp ~/mininet/examples/empt
ynet.py ~/work/lab_iperf3/lab_iperf3_topo
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ mv emptynet.py lab_iperf3_
topo.py
```

Figure 1: Подкаталог с примером скрипта

Изучил содержание скрипта



```
mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo
GNU nano 4.8 lab_iperf3_topo.py
s3 = net.addSwitch( 's3' )

info( '*** Creating links\n' )
net.addLink( h1, s3 )
net.addLink( h2, s3 )

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

info( '*** Running CLI\n' )
CLI( net )

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

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

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

Figure 2: Содержание скрипта

Запустил скрипт

```
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ sudo python lab_iperf3_topo.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
*** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s3 ...
*** Waiting for switches to connect
s3
*** Running CLI
*** Starting CLI:
mininet>
```

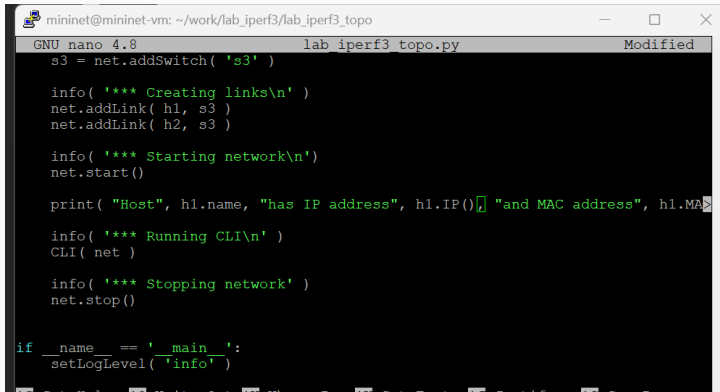
Figure 3: Запуск скрипта

Посмотрел элементы топологии и завершил работу mininet

```
mininet> net
h1 h1-eth0:s3-eth1
h2 h2-eth0:s3-eth2
s3 lo: s3-eth1:h1-eth0 s3-eth2:h2-eth0
c0
mininet> links
h1-eth0<->s3-eth1 (OK OK)
h2-eth0<->s3-eth2 (OK OK)
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=1692>
<Host h2: h2-eth0:10.0.0.2 pid=1696>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=1701>
<Controller c0: 127.0.0.1:6653 pid=1685>
mininet> exit
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s3
*** Stopping 2 hosts
h1 h2
```

Figure 4: Элементы топологии

Изменение, позволяющее вывести на экран информацию о хосте h1



```
mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo
GNU nano 4.8 lab_iperf3_topo.py Modified
s3 = net.addSwitch( 's3' )

info( '*** Creating links\n' )
net.addLink( h1, s3 )
net.addLink( h2, s3 )

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

print( "Host", h1.name, "has IP address", h1.IP(), "and MAC address", h1.MAC )

info( '*** Running CLI\n' )
CLI( net )

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

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

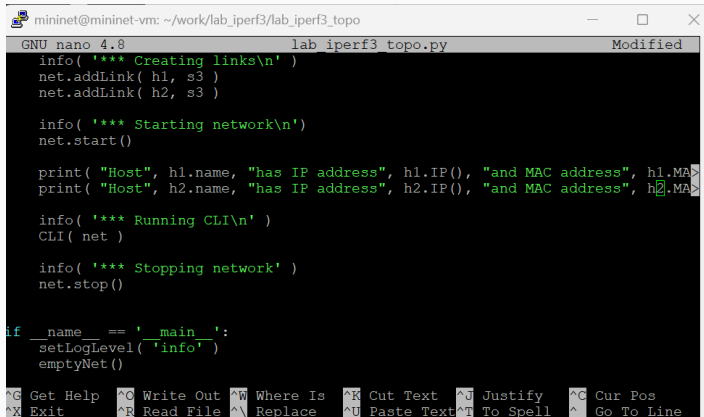
Figure 5: Вывд на экран информации о хосте h1

Запуск нового скрипта

```
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ sudo python lab_iperf3_topo.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
*** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s3 ...
*** Waiting for switches to connect
s3
Host h1 has IP address 10.0.0.1 and MAC address fe:0b:ea:c0:b6:33
*** Running CLI
*** Starting CLI:
mininet> █
```

Figure 6: Запуск нового скрипта

Изменил, чтобы выводилась информация об обоих хостах



```
mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo
GNU nano 4.8 lab_iperf3_topo.py Modified
info( '*** Creating links\n' )
net.addLink( h1, s3 )
net.addLink( h2, s3 )

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

print( "Host", h1.name, "has IP address", h1.IP(), "and MAC address", h1.MAC )
print( "Host", h2.name, "has IP address", h2.IP(), "and MAC address", h2.MAC )

info( '*** Running CLI\n' )
CLI( net )

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

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

Figure 7: Информация об обоих хостах

Запуск нового скрипта

```
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ sudo python lab_iperf3_topo.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
*** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s3 ...
*** Waiting for switches to connect
s3
Host h1 has IP address 10.0.0.1 and MAC address 12:39:1f:b5:da:e3
Host h2 has IP address 10.0.0.2 and MAC address 6a:5f:45:7c:57:3a
*** Running CLI
*** Starting CLI:
mininet>
```

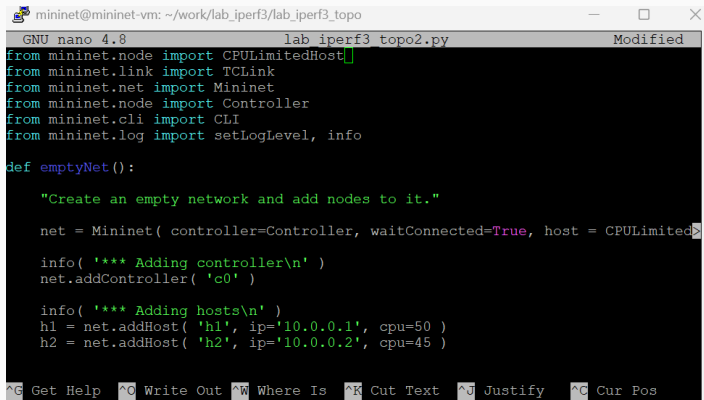
Figure 8: Запуск нового скрипта

Сделал копию скрипта lab_iperf3_topo.py

```
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ cp lab_iperf3_topo.py lab_iperf3_topo2.py
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$
```

Figure 9: Копия скрипта lab_iperf3_topo.py

Импорт классов, изменил описание сети, размер CPU и тд



The screenshot shows a terminal window with the title bar "mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo". The window contains a nano 4.8 editor editing the file "lab_iperf3_topo2.py". The script content is as follows:

```
from mininet.node import CPULimitedHost
from mininet.link import TCLink
from mininet.net import Mininet
from mininet.node import Controller
from mininet.cli import CLI
from mininet.log import setLogLevel, info

def emptyNet():

    "Create an empty network and add nodes to it."

    net = Mininet( controller=Controller, waitConnected=True, host = CPULimitedHost )

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

    info( '*** Adding hosts\n' )
    h1 = net.addHost( 'h1', ip='10.0.0.1', cpu=50 )
    h2 = net.addHost( 'h2', ip='10.0.0.2', cpu=45 )
```

The bottom of the terminal shows the nano editor's status bar with various keyboard shortcuts: ^G Get Help, ^O Write Out, ^W Where Is, ^K Cut Text, ^J Justify, ^C Cur Pos.

Figure 10: Импорт классов, описание сети, размер CPU и тд

Запуск нового скрипта

```
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ sudo python lab_iperf3_topo2.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
(10.00Mbit 5ms delay 10.00000% loss) (10.00Mbit 5ms delay 10.00000% loss) *** Starting network
*** Configuring hosts
h1 (cfs 5000000/100000us) h2 (cfs 4500000/100000us)
*** Starting controller
c0
*** Starting 1 switches
s3 (10.00Mbit 5ms delay 10.00000% loss) ... (10.00Mbit 5ms delay 10.00000% loss)
*** Waiting for switches to connect
s3
Host h1 has IP address 10.0.0.1 and MAC address 7e:1c:ae:11:5f:02
Host h2 has IP address 10.0.0.2 and MAC address 66:75:b6:d4:61:5b
*** Running CLI
*** Starting CLI:
mininet> █
```

Figure 11: Запуск нового скрипта

В `lab_iperf3_topo2.py`: - параметры линков: Отображаются характеристики (10.00Mbit 5ms delay 10.000000% loss) - конфигурация хостов: Подробная информация о CFS (Completely Fair Scheduler) Вывод: `lab_iperf3_topo2.py` показывает более детальную информацию о параметрах сети и планировщике, в то время как `lab_iperf3_topo.py` вывод упрощен

Сделал копию скрипта lab_iperf3_topo2.py и поместил его в подкаталог iperf

```
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ cp lab_iperf3_topo2.py lab_iperf3.py
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ mkdir -p ~/work/lab_iperf3/iperf3
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ mv ~/work/lab_iperf3/lab_iperf3_topo/lab_iperf3.py ~/work/lab_iperf3/iperf3
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ cd ~/work/lab_iperf3/iperf3
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ ls -l
total 4
-rwxrwxr-x 1 mininet mininet 1346 Oct 11 01:56 lab_iperf3.py
mininet@mininet-vm:~/work/lab_iperf3/iperf3$
```

Figure 12: Копия скрипта lab_iperf3_topo2.py

Далее будут внесены след. изменения: Хосты: Без ограничений CPU Каналы: 100 Мбит/с, задержка 75 мс, без потерь, без ограничителей iPerf3 тест: h2 запускает сервер iPerf3 h1 через 10 сек запускает клиент iPerf3 с сохранением в JSON

```
info( '*** Adding hosts\n' )
h1 = net.addHost( 'h1', ip='10.0.0.1', cpu=1.0 )
h2 = net.addHost( 'h2', ip='10.0.0.2', cpu=1.0 )

info( '*** Adding switch\n' )
s3 = net.addSwitch( 's3' )

info( '*** Creating links\n' )
net.addLink( h1, s3, bw=100, delay='75ms', loss=0, use_htb=False )
net.addLink( h2, s3, bw=100, delay='75ms', loss=0, use_htb=False )

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

info( '*** Traffic generation\n' )
h2.cmdPrint( 'iperf3 -s -D -1' )
time.sleep(10)
h1.cmdPrint( 'iperf3 -c', h2.IP(), '-J > iperf_result.json' )
```

Figure 13: Изменения в коде

Запуск скрипта

```
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ sudo python lab_iperf3.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
(100.00Mbit 75ms delay 0.00000% loss) (100.00Mbit 75ms delay 0.00000% loss) (100.00Mbit 75ms delay 0.00000% loss) (100.00Mbit 75ms delay 0.00000% loss) *** Starting network
*** Configuring hosts
h1 (cfs 100000/100000us) h2 (cfs 100000/100000us)
*** Starting controller
c0
*** Starting 1 switches
s3 (100.00Mbit 75ms delay 0.00000% loss) (100.00Mbit 75ms delay 0.00000% loss) ..(100.00Mbit 75ms delay 0.00000% loss) (100.00Mbit 75ms delay 0.00000% loss)
*** Waiting for switches to connect
s3
*** Traffic generation
*** h2 : ('iperf3 -s -D -l',)
*** h1 : ('iperf3 -c', '10.0.0.2', '-J > iperf_result.json')
Host h1 has IP address 10.0.0.1 and MAC address 32:43:31:01:96:8f
Host h2 has IP address 10.0.0.2 and MAC address fa:b2:c2:94:1a:b3
*** Running CLI
*** Starting CLI:
```

Figure 14: Запуск скрипта

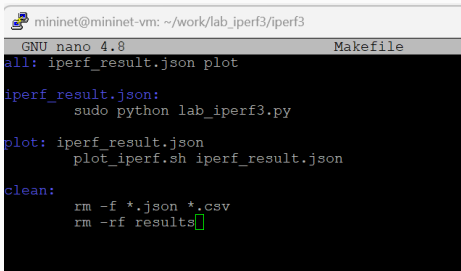
Характеристики каналов: - пропускная способность: 100 Мбит/с - задержка: 75 мс - потери: 0% (без потерь) Конфигурация хостов: - h1 и h2 - оба настроены с параметрами CFS (без ограничений CPU) Автоматический тест iperf3: - h2 запущен как сервер: iperf3 -s -D -1 - h1 запущен как клиент к 10.0.0.2 с сохранением в JSON

Построил графики из получившегося JSON-файла

```
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ plot_iperf.sh iperf_result.json
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ cd results
mininet@mininet-vm:~/work/lab_iperf3/iperf3/results$ ls
1.dat      cwnd.pdf  retransmits.pdf  RTT_Var.pdf
bytes.pdf  MTU.pdf   RTT.pdf          throughput.pdf
mininet@mininet-vm:~/work/lab_iperf3/iperf3/results$
```

Figure 15: Графики из получившегося JSON-файла

Создал Makefile

A screenshot of a terminal window. The top bar shows the user 'mininet' on a 'mininet-vm' machine, with the current directory being '~/work/lab_iperf3/iperf3'. Below the bar, the text 'GNU nano 4.8' is on the left and 'Makefile' is on the right. The main content of the terminal is a Makefile with the following rules:

```
all: iperf_result.json plot

iperf_result.json:
    sudo python lab_iperf3.py

plot: iperf_result.json
    plot_iperf.sh iperf_result.json

clean:
    rm -f *.json *.csv
    rm -rf results
```

The cursor is at the end of the 'rm -rf results' line.

Figure 16: Makefile

Проверка Makefile, все работает успешно

```
mininet@mininet-vm:~/work/lab_iperf3/iperf3$make
make: command not found
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ make
sudo python lab_iperf3.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
(100.00Mbit 75ms delay 0.00000% loss) (100.00Mbit 75ms delay 0.00000% loss) (100.00Mbit 75ms delay 0.00000% loss) (100.00Mbit 75ms delay 0.00000% loss) *** Starting network
*** Configuring hosts
h1 (cfs 100000/100000us) h2 (cfs 100000/100000us)
*** Starting controller
c0
*** Starting 1 switches
s3 (100.00Mbit 75ms delay 0.00000% loss) (100.00Mbit 75ms delay 0.00000% loss) ..(100.00Mbit 75ms delay 0.00000% loss) (100.00Mbit 75ms delay 0.00000% loss)
*** Waiting for switches to connect
s3
*** Traffic generation
*** h2 : ('iperf3 -s -D -l',)
*** h1 : ('iperf3 -c', '10.0.0.2', '-J > iperf_result.json')
Host h1 has IP address 10.0.0.1 and MAC address 72:eb:6e:1f:ac:5f
Host h2 has IP address 10.0.0.2 and MAC address da:0f:0e:6e:5f:56
*** Running CLI
*** Starting CLI:
mininet> █
```

Figure 17: Проверка Makefile

Ознакомился с инструментом для измерения пропускной способности сети в режиме реального времени — iPerf3, а также получил навыки проведения воспроизводимого эксперимента по измерению пропускной способности моделируемой сети в среде Mininet.