

Лабораторная работа №3

Моделирование сетей передачи данных

Коннова Т. А.

2025

Российский университет дружбы народов имени Патриса Лумумбы, Москва, Россия

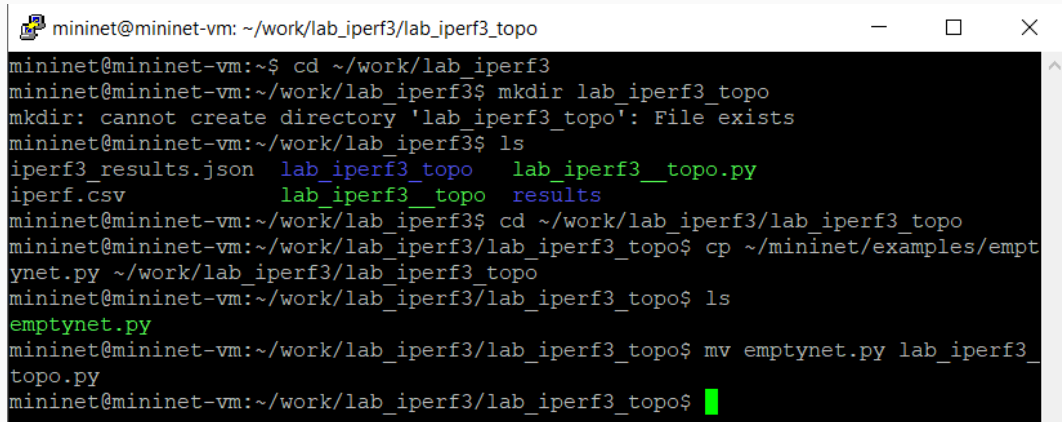
- Коннова Татьяна Алексеевна
- Студентка группы НПИбд-01-22
- Студ. билет 1132221814
- Российский университет дружбы народов имени Патриса Лумумбы



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

Выполнение лабораторной работы

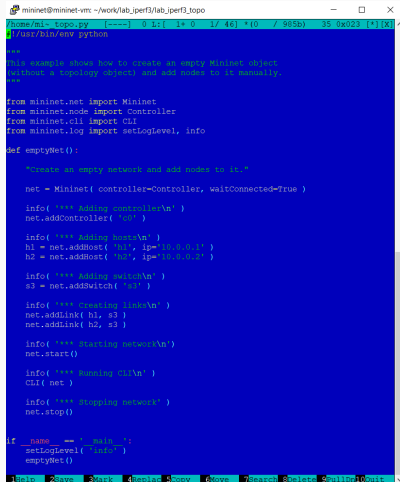
Создание простейшей топологии сети



```
mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo
mininet@mininet-vm:~$ cd ~/work/lab_iperf3
mininet@mininet-vm:~/work/lab_iperf3$ mkdir lab_iperf3_topo
mkdir: cannot create directory 'lab_iperf3_topo': File exists
mininet@mininet-vm:~/work/lab_iperf3$ ls
iperf3_results.json  lab_iperf3_topo    lab_iperf3_topo.py
iperf.csv            lab_iperf3_topo    results
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/emptynet.py ~/work/lab_iperf3/lab_iperf3_topo
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ ls
emptynet.py
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ mv emptynet.py lab_iperf3_topo.py
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$
```

Рис. 1: Создание подкаталога, копирование файла с примером скрипта (описывающего стандартную простую топологию сети mininet)

Создание простейшей топологии сети



```
mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo
/home/mininet/lab_iperf3_topo.py [----] 0 L: [ 1* 0 1/ 46] *{0 / 965b} 35 0x023 [*]{X}
~/usr/bin/env python

***
This example shows how to create an empty Mininet object
(without a topology object) and add nodes to it manually.
***

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 )

    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' )
    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\n' )
    net.stop()

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

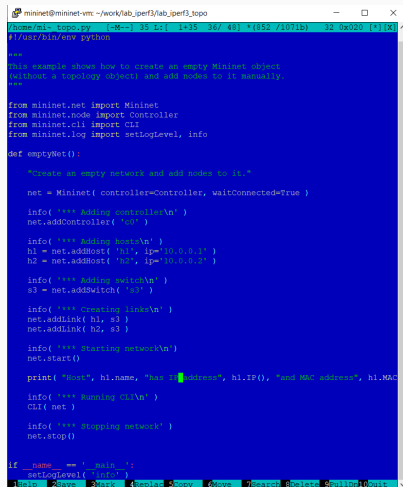
Рис. 2: Открытие файла lab_iperf3_topo.py

Создание простейшей топологии сети

```
mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo
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> 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=823>
<Host h2: h2-eth0:10.0.0.2 pid=827>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=832>
<Controller c0: 127.0.0.1:6653 pid=816>
mininet> exit
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s3
*** Stopping 2 hosts
h1 h2
*** Done
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$
```

Рис. 3: Запуск скрипта создания топологии и дальнейший просмотр элементов

Внесение изменений в скрипт



```
mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo
/home/ml_topo.py [-M--] 35 L: [ 1+35 36/ 48] *(852 /1071b) 32 0x020 [*][X]
#!/usr/bin/env python

"""
This example shows how to create an empty Mininet object
(without a topology object) and add nodes to it manually.
"""

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 )

    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" )
    s3 = net.addSwitch( 's3' )

    info( "**** Creating links\n" )
    net.addlink( h1, s3 )
    net.addlink( h2, s3 )

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

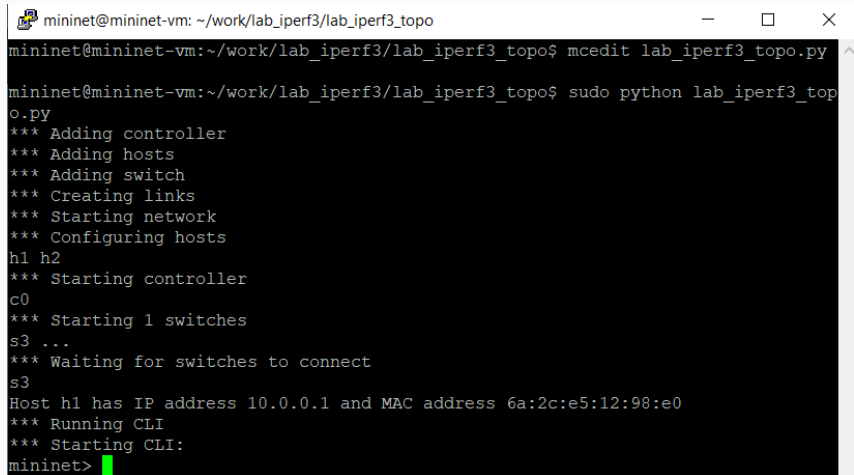
    print( "Host", h1.name, "has IP address", h1.IP(), "and MAC address", h1.MAC() )

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

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

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

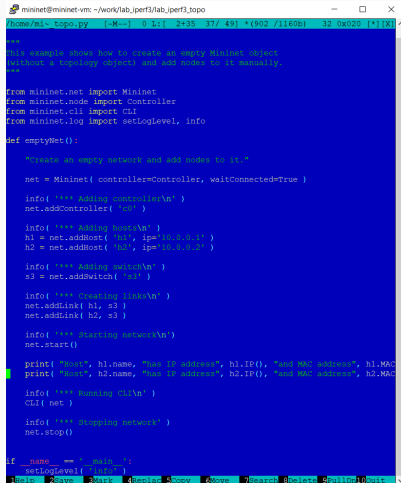
Рис. 4: Внесение изменения в скрипт, позволяющего вывести на экран информацию о хосте h1 (имя, IP-адрес, MAC-адрес)

A terminal window titled 'mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo'. The window shows the execution of a script 'lab_iperf3_topo.py' using 'mcedit' and 'python'. The script output includes steps like 'Adding controller', 'Adding hosts', 'Adding switch', 'Creating links', 'Starting network', and 'Configuring hosts'. It then lists hosts 'h1 h2' and starts the controller 'c0'. It also shows 'Starting 1 switches' and 'Waiting for switches to connect'. Finally, it displays the IP and MAC addresses for host 'h1' and starts the CLI.

```
mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ mcedit lab_iperf3_topo.py
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 6a:2c:e5:12:98:e0
*** Running CLI
*** Starting CLI:
mininet>
```

Рис. 5: Проверка корректности отработки скрипта

Внесение изменений в скрипт



```
mininet@mininet-vm: ~/work/lab_perf3/lab_perf3_topo
/home/mininet/topo.py [-M--] 0 L: [ 2+35 37/ 49] *(902 /1160b) 32 0x020 [*](X)
***
This example shows how to create an empty Mininet object
(without a topology object) and add nodes to it manually.
***

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 )

    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" )
    s3 = net.addSwitch( 's3' )

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

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

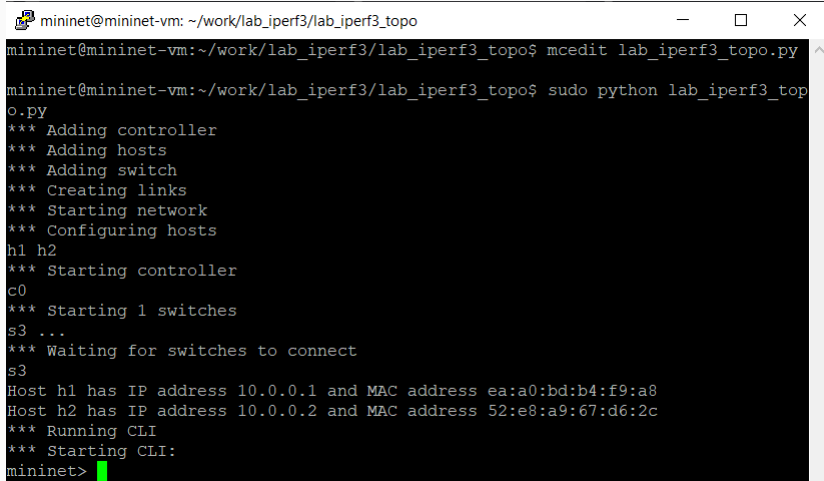
    print( "Host", h1.name, "has IP address", h1.IP(), "and MAC address", h1.MAC )
    print( "Host", h2.name, "has IP address", h2.IP(), "and MAC address", h2.MAC )

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

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

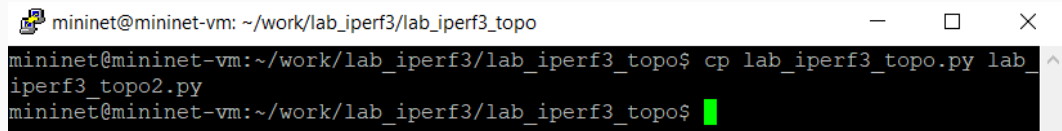
if __name__ == '__main__':
    setLogLevel( 'info' )
```

Рис. 6: Внесение изменения в скрипт, позволяющего вывести на экран информацию о двух хостах (имя, IP-адрес, MAC-адрес)



```
mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ mcedit lab_iperf3_topo.py ^
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 ea:a0:bd:b4:f9:a8
Host h2 has IP address 10.0.0.2 and MAC address 52:e8:a9:67:d6:2c
*** Running CLI
*** Starting CLI:
mininet> 
```

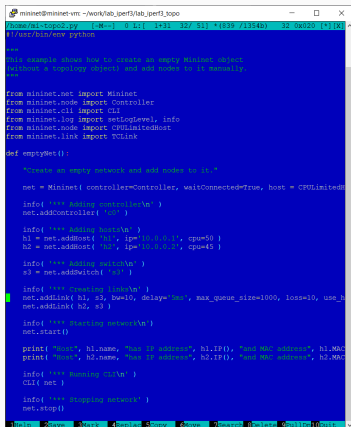
Рис. 7: Проверка корректности отработки скрипта

A terminal window with a title bar containing a file icon, the text 'mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo', and standard window controls (minimize, maximize, close). The terminal has a black background with white text. It shows a prompt 'mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo\$' followed by the command 'cp lab_iperf3_topo.py lab_iperf3_topo2.py'. The command is executed, and the next line shows the prompt again with a green cursor. A vertical scrollbar is visible on the right side of the terminal window.

```
mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo
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$
```

Рис. 8: Создание копии скрипта lab_iperf3_topo.py

Добавление в скрипт настроек параметров производительности



```
mininet@mininet-vm ~/work/lab_iperf3/lab_iperf3_topo
/home/mininet$ python
#!/usr/bin/env python

"""
This example shows how to create an empty Mininet object
without a topology object and add nodes to it manually.
"""

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

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 )

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

    info( '*** Creating links\n' )
    net.addLink( h1, s3, bw=10, delay='5ms', max_queue_size=1000, loss=10, use_h
    net.addLink( h2, s3 )

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

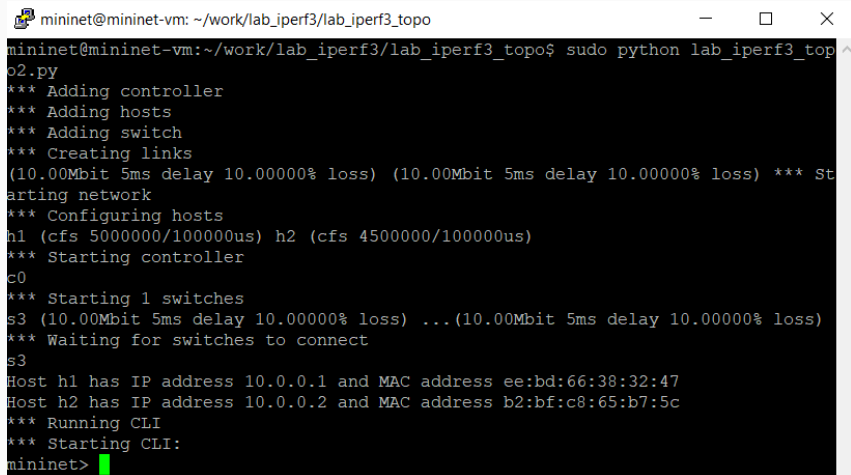
    print( "Host", h1.name, "has IP address", h1.IP(), "and MAC address", h1.MAC
    print( "Host", h2.name, "has IP address", h2.IP(), "and MAC address", h2.MAC

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

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

Рис. 9: Изменение скрипта lab_iperf3_topo2.py: добавление ипорта классов, изменение строки описания сети, изменение функции задания параметров виртуального хоста h1 и h2, изменение функции параметров соединения между хостом h1 и коммутатором s3

Добавление в скрипт настроек параметров производительности

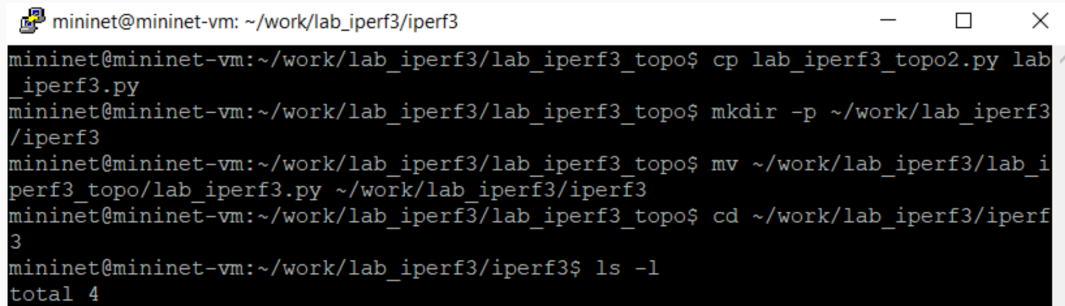


A terminal window titled "mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo" with standard window controls. The terminal displays the output of running "sudo python lab_iperf3_topo2.py". The output shows the Mininet environment setup process, including adding a controller, hosts, and a switch, creating links with specific performance parameters (10.00Mbit, 5ms delay, 10.00000% loss), starting the network, configuring hosts h1 and h2 with their respective IP and MAC addresses, starting the controller c0 and switch s3, and finally running the CLI.

```
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 ee:bd:66:38:32:47
Host h2 has IP address 10.0.0.2 and MAC address b2:bf:c8:65:b7:5c
*** Running CLI
*** Starting CLI:
mininet> █
```

Рис. 10: Запуск скрипта lab_iperf3_topo2.py на отработку

Добавление в скрипт настроек параметров производительности

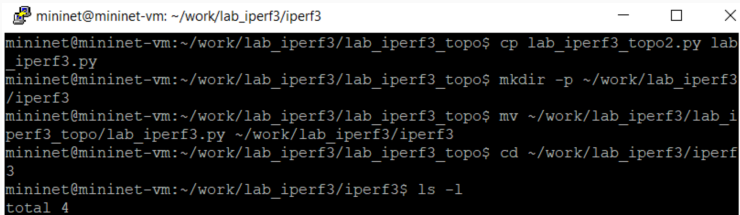


A terminal window titled "mininet@mininet-vm: ~/work/lab_iperf3/iperf3" with standard window controls. The terminal shows a series of commands and their outputs:

```
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
```

Рис. 11: Создание копии скрипта lab_iperf3_topo2.py и его дальнейшее помещение в подкаталог iperf

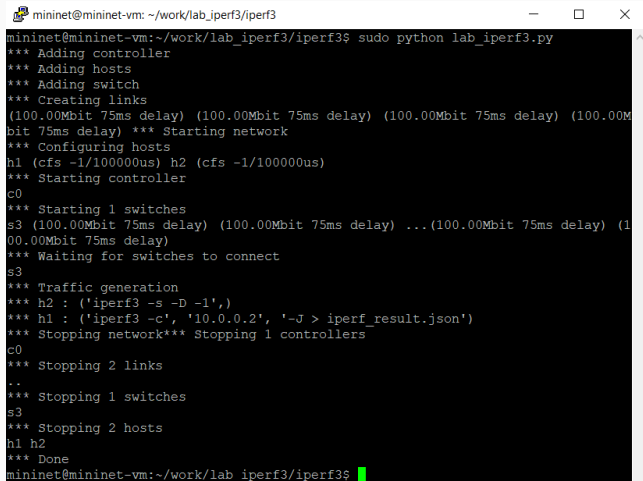
Добавление в скрипт настроек параметров производительности



```
mininet@mininet-vm: ~/work/lab_iperf3/iperf3
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
```

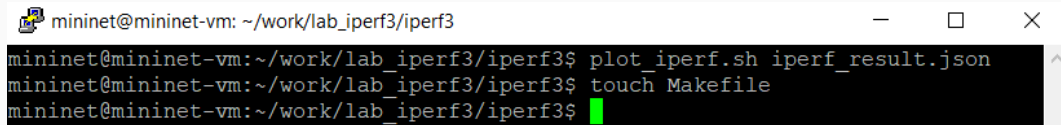
Рис. 12: Добавление в скрипт lab_iperf3.py записи об импорте time; снятие ограничений по использованию ресурсов процессора; добавление кода, чтобы каналы между хостами и коммутатором были по 100 Мбит/с с задержкой 75 мс, без потерь

Добавление в скрипт настроек параметров производительности



```
mininet@mininet-vm: ~/work/lab_iperf3/iperf3
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ sudo python lab_iperf3.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
(100.00Mbit 75ms delay) (100.00Mbit 75ms delay) (100.00Mbit 75ms delay) (100.00M
bit 75ms delay) *** Starting network
*** Configuring hosts
h1 (cfs -l/1000000us) h2 (cfs -l/1000000us)
*** Starting controller
c0
*** Starting 1 switches
s3 (100.00Mbit 75ms delay) (100.00Mbit 75ms delay) ... (100.00Mbit 75ms delay) (1
00.00Mbit 75ms delay)
*** Waiting for switches to connect
s3
*** Traffic generation
*** h2 : ('iperf3 -s -D -l',)
*** h1 : ('iperf3 -c', '10.0.0.2', '-J > iperf_result.json')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s3
*** Stopping 2 hosts
h1 h2
*** Done
mininet@mininet-vm:~/work/lab_iperf3/iperf3$
```

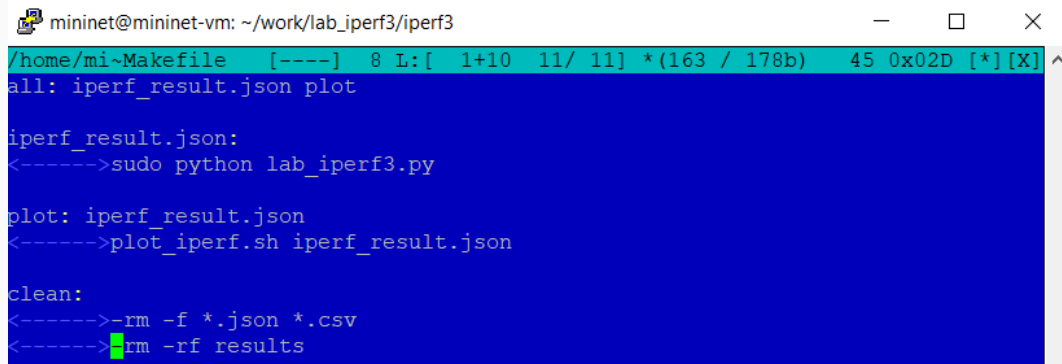
Рис. 14: Запуск скрипта lab_iperf3.py на отработку



```
mininet@mininet-vm: ~/work/lab_iperf3/iperf3
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ plot_iperf.sh iperf_result.json
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ touch Makefile
mininet@mininet-vm:~/work/lab_iperf3/iperf3$
```

Рис. 15: Построение графиков и создание Makefile для проведения всего эксперимента

Построение графиков по проводимому эксперименту



```
mininet@mininet-vm: ~/work/lab_iperf3/iperf3
/home/mi~Makefile  [----]  8 L:[  1+10  11/ 11] *(163 / 178b)  45 0x02D [*][X] ^
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
```

Рис. 16: Добавление скрипта в Makefile

Построение графиков по проводимому эксперименту

```
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ make Clean
rm -f *.json *.csv
rm -rf results
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) (100.00Mbit 75ms delay) (100.00Mbit 75ms delay) (100.00M
bit 75ms delay) *** Starting network
*** Configuring hosts
h1 (cfs -l/1000000us) h2 (cfs -l/1000000us)
*** Starting controller
c0
*** Starting 1 switches
s3 (100.00Mbit 75ms delay) (100.00Mbit 75ms delay) ... (100.00Mbit 75ms delay) (1
00.00Mbit 75ms delay)
*** Waiting for switches to connect
s3
*** Traffic generation
*** h2 : ('iperf3 -s -D -1',)
*** h1 : ('iperf3 -c', '10.0.0.2', '-J > iperf_result.json')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s3
*** Stopping 2 hosts
h1 h2
*** Done
plot_iperf.sh iperf_result.json
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ mcedit Makefile
mininet@mininet-vm:~/work/lab_iperf3/iperf3$
```

Рис. 17: Проверка корректности отработки Makefile

Вывод

- В ходе выполнения лабораторной работы познакомились с инструментом для измерения пропускной способности сети в режиме реального времени — iPerf3, а также получили навыки проведения воспроизводимого эксперимента по измерению пропускной способности моделируемой сети в среде Mininet

Список литературы. Библиография

[1] Mininet: <https://mininet.org/>