Моделирование сетей передачи данных

Лабораторная работа №3:Измерение и тестирование пропускной способности сети. Воспроизводимый эксперимент

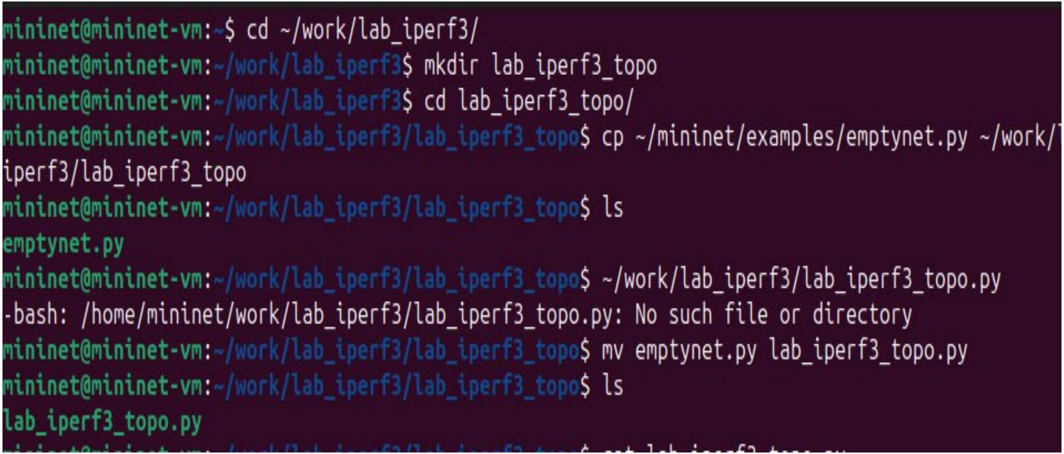
Кармацкий Никита Сергеевич

Российский университет дружбы народов, Москва, Россия

Цель лабораторной работы

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.

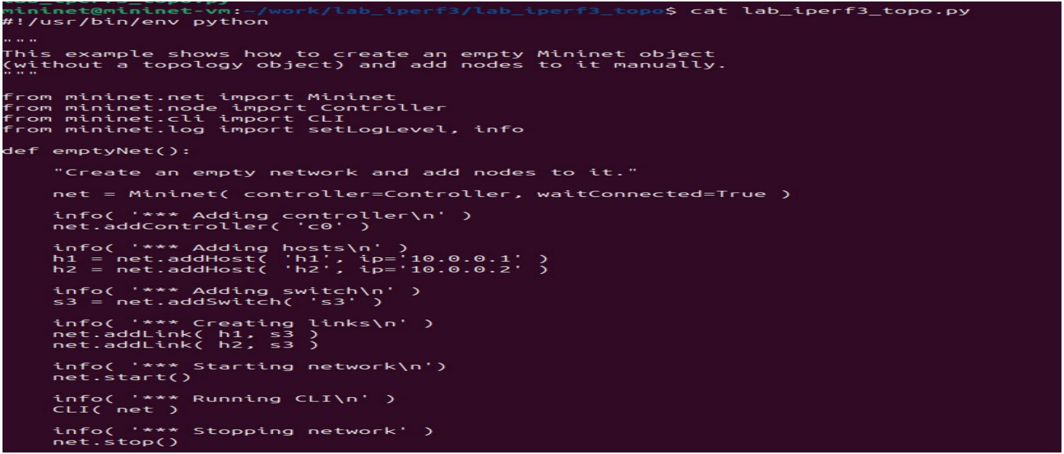
Выполнение лабораторной работы: 1. Создание простейшей топологии



```
mininet@mininet-vm:~$ cd ~/work/lab_iperf3/
mininet@mininet-vm:~/work/lab_iperf3$ mkdir lab_iperf3_topo
mininet@mininet-vm:~/work/lab_iperf3$ cd lab_iperf3_topo/
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ cp ~/mininet/examples/emptynet.py ~/work/l
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$ ~/work/lab_iperf3/lab_iperf3_topo.py
-bash: /home/mininet/work/lab_iperf3/lab_iperf3_topo.py: No such file or directory
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$ ls
lab_iperf3_topo.py
```

Рис. 1: Создание подкаталога и копирование файлов

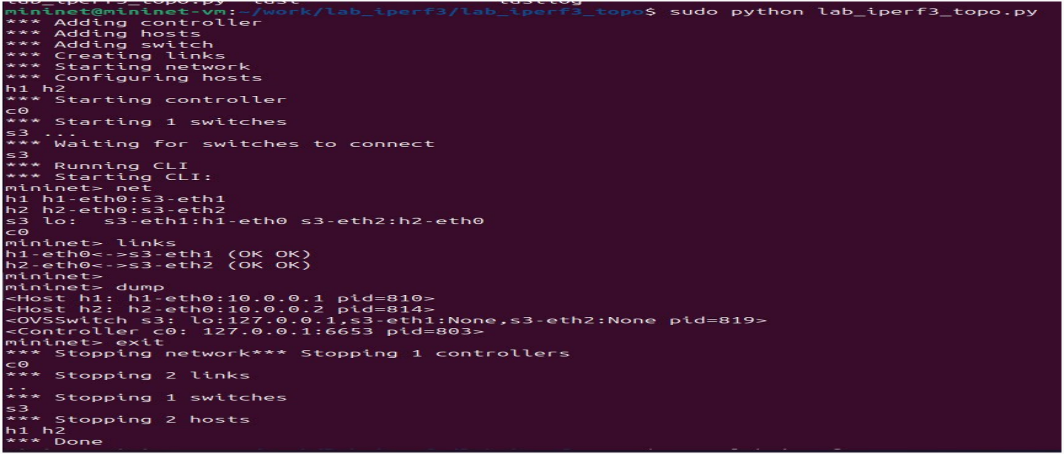
1. Создание простейшей топологии



```
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ cat lab_iperf3_topo.py
#!/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' )
    net.stop()
```

Рис. 2: Скрипт lab_iperf3_topo.py

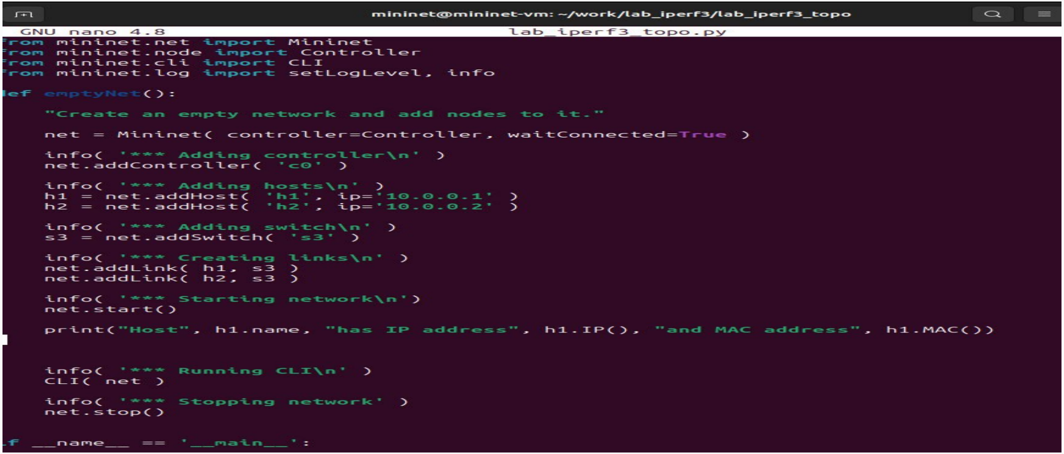
1. Создание простейшей топологии



```
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>
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=810>
<Host h2: h2-eth0:10.0.0.2 pid=814>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=819>
<Controller c0: 127.0.0.1:6653 pid=803>
mininet> exit
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s3
*** Stopping 2 hosts
h1 h2
*** Done
```

Рис. 3: Запуск скрипта и просмотр элементов топологии

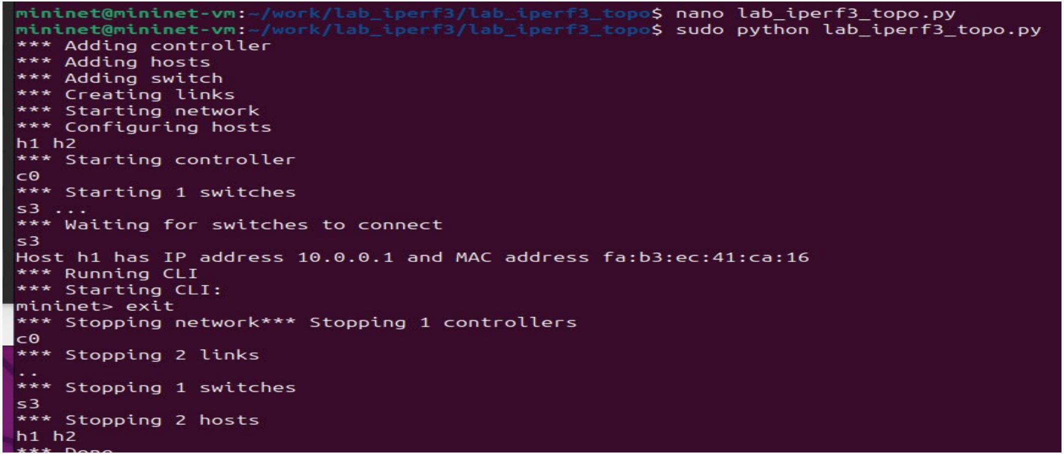
2. Внесеение изменений в скрипт



```
 GNU nano 4.8                                      lab_iperf3_topo.py
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' )
    net.stop()

if __name__ == '__main__':
```

Рис. 4: Редактирование скрипта для просмотра информации по хосту h1

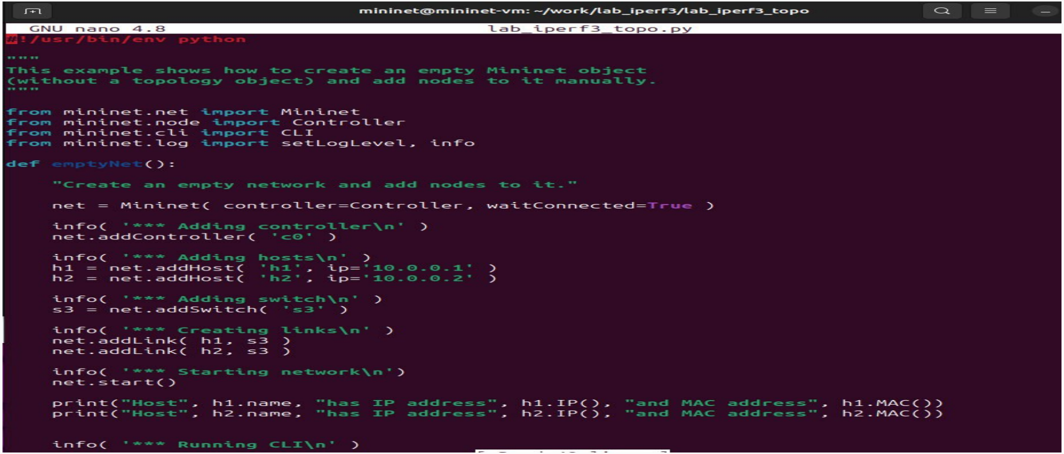
2. Проверка результатов



```
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ nano 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 fa:b3:ec:41:ca:16
*** Running CLI
*** Starting CLI:
mininet> exit
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s3
*** Stopping 2 hosts
h1 h2
*** Done
```

Рис. 5: Проверка работоспособности скрипта

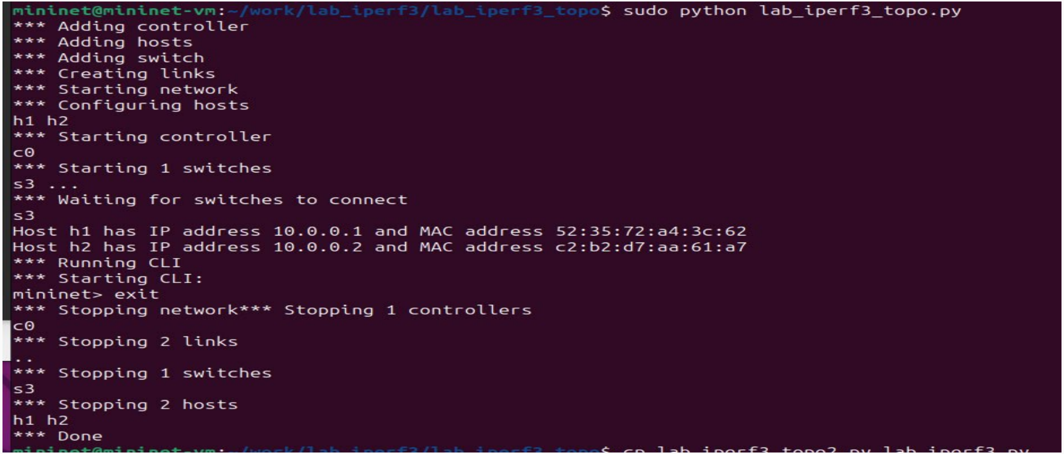
2. Внесеение изменений в скрипт



```
  GNU nano 4.8                                 lab_iperf3_topo.py
#!/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())
    print("Host", h2.name, "has IP address", h2.IP(), "and MAC address", h2.MAC())


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

Рис. 6: Изменение скрипта для просмотра инофрмации по двум хостам

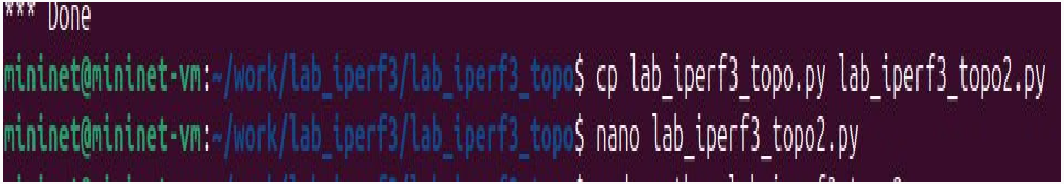
2. Проверка результатов



```
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 52:35:72:a4:3c:62
Host h2 has IP address 10.0.0.2 and MAC address c2:b2:d7:aa:61:a7
*** Running CLI
*** Starting CLI:
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$ cp lab_iperf3_topo2.py lab_iperf3.py
```

Рис. 7: Просмотр результатов работы скрипта

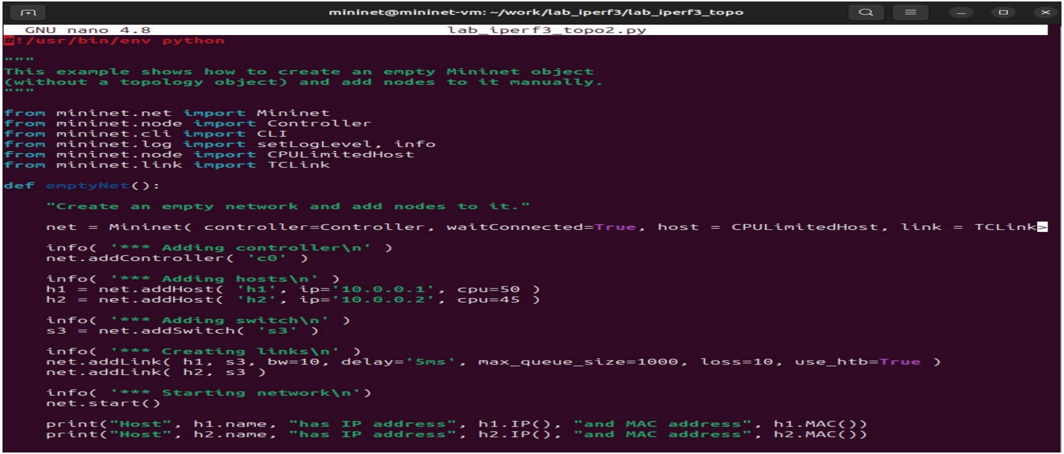
3. Добавление в скрипт настроек производительности



```
*** Done
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$ nano lab_iperf3_topo2.py
```

Рис. 8: Копирование скрипта lab_iperf3_topo.py

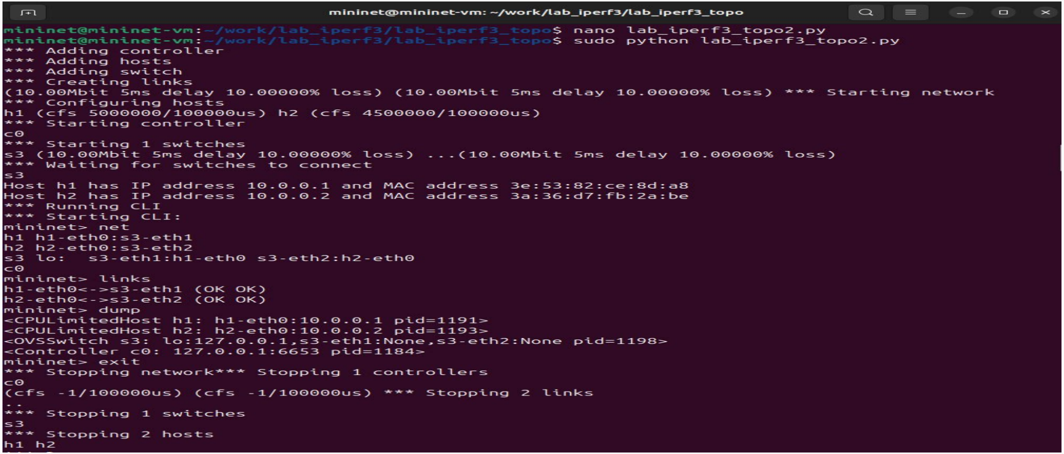
3. Добавление в скрипт настроек производительности



```
  GNU nano 4.8                         lab_iperf3_topo2.py
#!/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, link = TCLink>

    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_htb=True )
    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())
```

Рис. 9: Изменение скрипта: Добавляем импорт новых классов, меняем строку описание сети, задаем новые параметры для хоста h1 и h2, меняем соедиение между хостом h1 и коммутатором s3

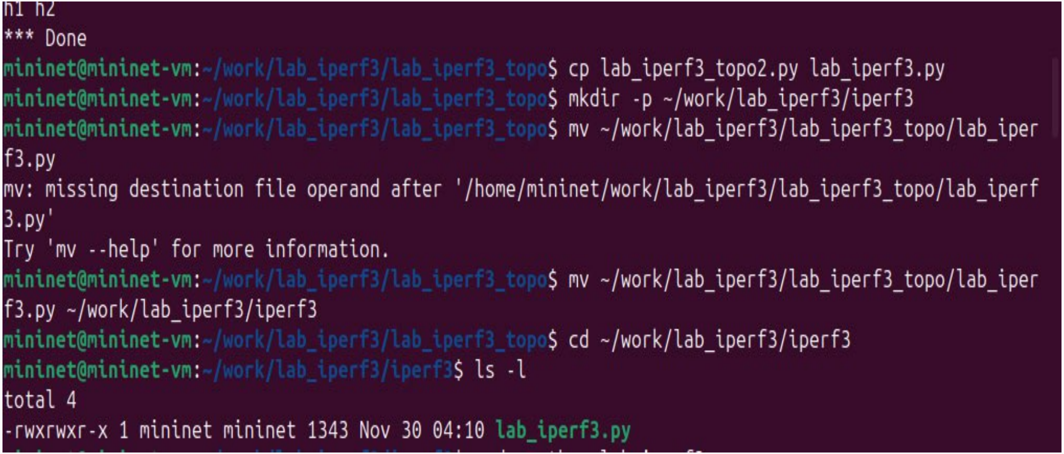
3. Добавление в скрипт настроек производительности



```
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ nano lab_iperf3_topo2.py
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 3e:53:82:ce:8d:a8
Host h2 has IP address 10.0.0.2 and MAC address 3a:36:d7:fb:2a:be
*** 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
<CPULimitedHost h1: h1-eth0:10.0.0.1 pid=1191>
<CPULimitedHost h2: h2-eth0:10.0.0.2 pid=1193>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=1198>
<Controller c0: 127.0.0.1:6653 pid=1184>
mininet> exit
*** Stopping network*** Stopping 1 controllers
c0
(cfs -1/100000us) (cfs -1/100000us) *** Stopping 2 links
..
*** Stopping 1 switches
s3
*** Stopping 2 hosts
h1 h2
```

Рис. 10: Запуск скрипта и результаты работы

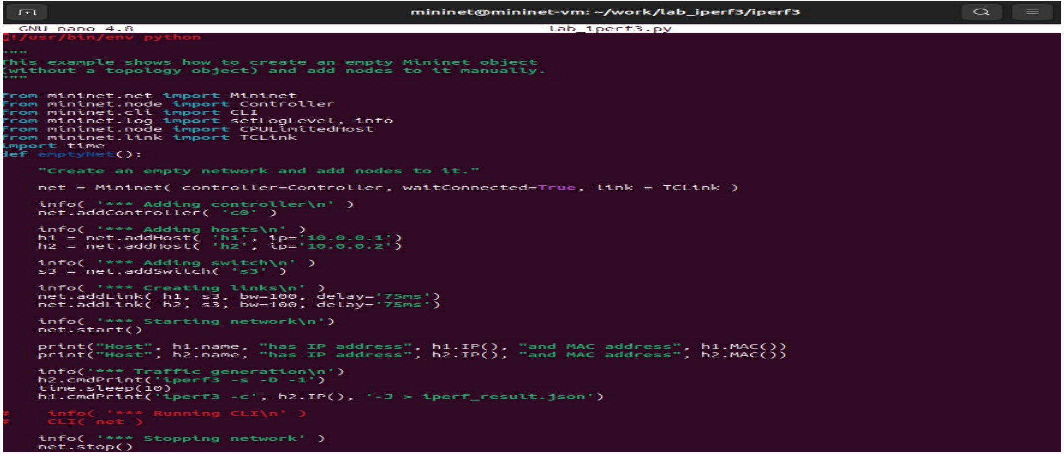
3. Добавление в скрипт настроек производительности



```
h1 h2
*** Done
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_iper
f3.py
mv: missing destination file operand after '/home/mininet/work/lab_iperf3/lab_iperf3_topo/lab_iperf
3.py'
Try 'mv --help' for more information.
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ mv ~/work/lab_iperf3/lab_iperf3_topo/lab_iper
f3.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 1343 Nov 30 04:10 lab_iperf3.py
```

Рис. 11: Создаем новый подкаталог и копируем наш скрипт lab_iperf_topo2.py

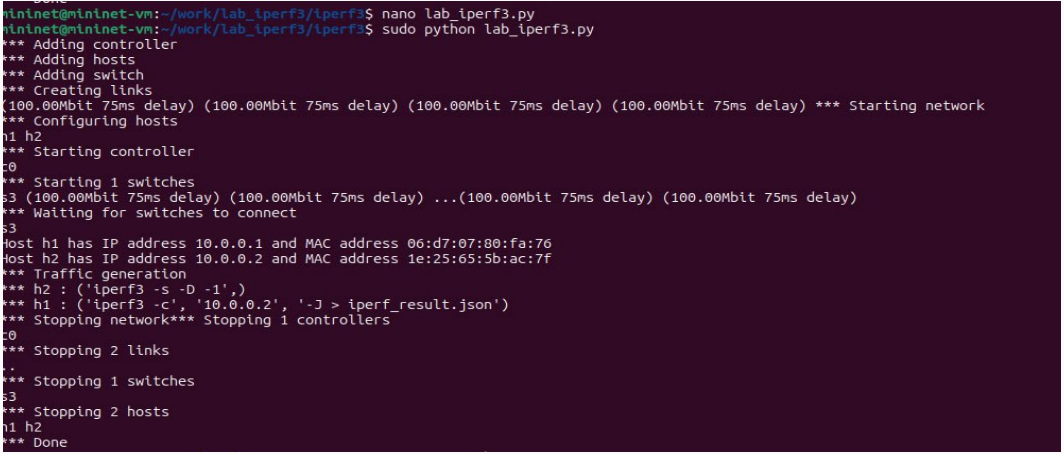
3. Добавление в скрипт настроек производительности



```
#!/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
import time
def emptyNet():

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

    net = Mininet( controller=Controller, waitConnected=True, link = TCLink )

    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, bw=100, delay='75ms')
    net.addLink( h2, s3, bw=100, delay='75ms')

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

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

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

Рис. 12: Добавление в скрипт библиотеки time, изменение в работе хостов, настройка каналов между коммутатором и хостами, добавление функции записи сервера iperf3ан хосте 2 и запуска клиента через 10 секунд на хосте 1, запись результатов в файл

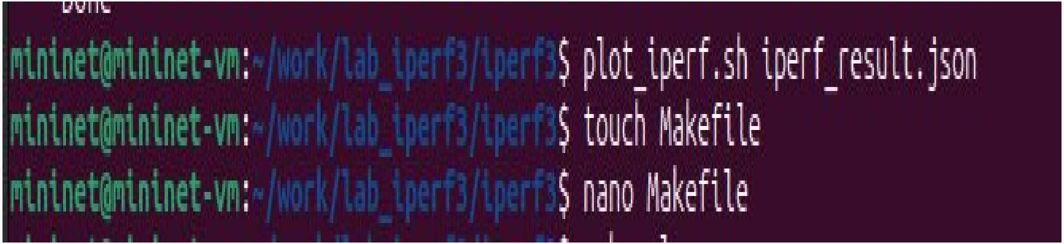
3. Добавление в скрипт настроек производительности



```
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ nano lab_iperf3.py
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) (100.00Mbit 75ms delay) *** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s3 (100.00Mbit 75ms delay) (100.00Mbit 75ms delay) ...(100.00Mbit 75ms delay) (100.00Mbit 75ms delay)
*** Waiting for switches to connect
s3
Host h1 has IP address 10.0.0.1 and MAC address 06:d7:07:80:fa:76
Host h2 has IP address 10.0.0.2 and MAC address 1e:25:65:5b:ac:7f
*** 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
```

Рис. 13: Запуск скрипта lab_iperf3.py на отработку

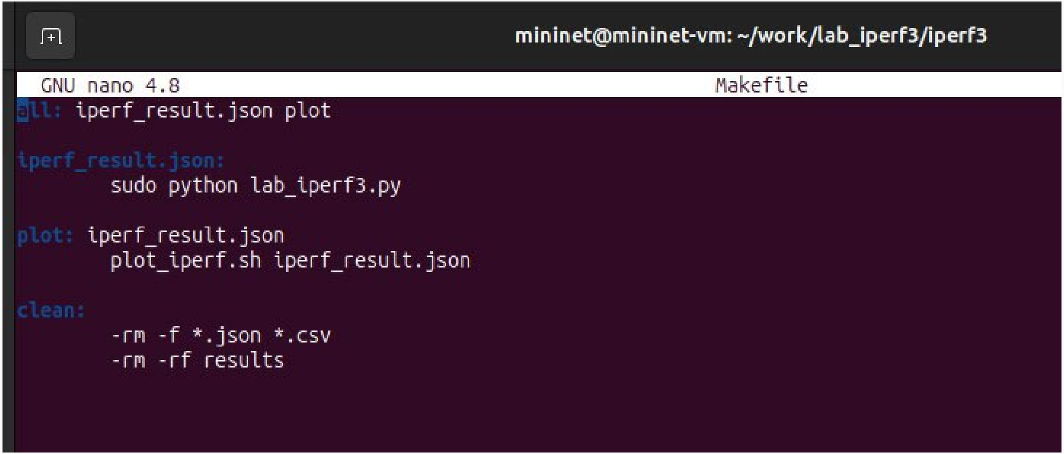
4. Построение графиков по проводимому эксперименту



```
     Done
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$ nano Makefile
```

Рис. 14: Постройка графиков и создание makefile

4. Построение графиков по проводимому эксперименту

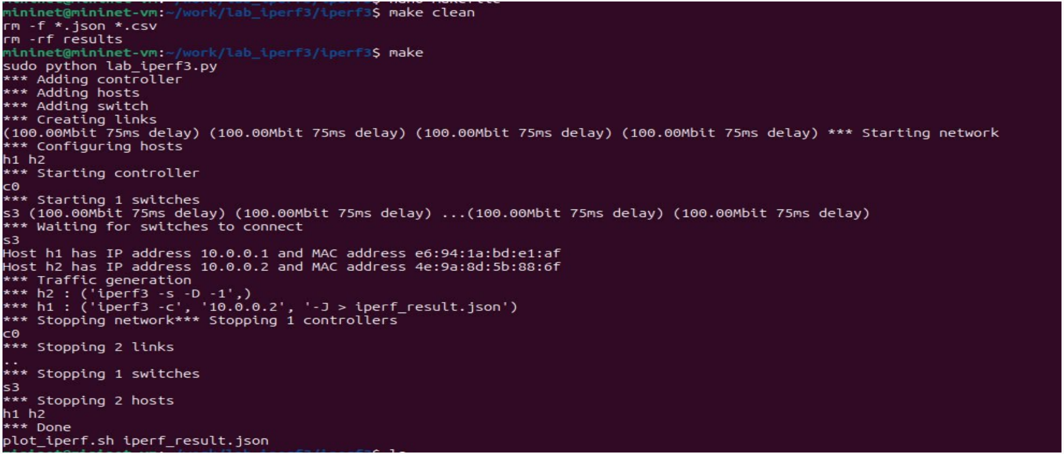


mininet@mininet-vm: ~/work/lab_iperf3/iperf3

```
 GNU nano 4.8                                                  Makefile
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. 15: Скрипт в Makefile

4. 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) (100.00Mbit 75ms delay) *** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s3 (100.00Mbit 75ms delay) (100.00Mbit 75ms delay) ...(100.00Mbit 75ms delay) (100.00Mbit 75ms delay)
*** Waiting for switches to connect
s3
Host h1 has IP address 10.0.0.1 and MAC address e6:94:1a:bd:e1:af
Host h2 has IP address 10.0.0.2 and MAC address 4e:9a:8d:5b:88:6f
*** 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
```

Рис. 16: Проверка корректности работы скрипта 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/