

Отчёт по лабораторной работе №3

НПИбд-02-22

Чесноков Артемий Павлович

Содержание

1 Цель работы	5
2 Выполнение лабораторной работы	6
3 Выводы	15
Список литературы	16

Список иллюстраций

2.1	in dir lab_iperf3_topo	6
2.2	Выводится консоль iperf3	7
2.3	net links dump	7
2.4	via print	8
2.5	наблюдаем корректное отображение для 1 хоста	8
2.6	тот же код только для h2	9
2.7	наблюдаем корректное отображение для 2 хоста	9
2.8	CPULimitHost & TCLink	10
2.9	в вызове Mininet	10
2.10	ура! добавили	11
2.11	наблюдаем ограничение сри	11
2.12	переименовали / перенесли / etc	12
2.13	так же для h1 добавляем вывод в отчёт	12
2.14	так же видим отключенный cli	13
2.15	для автоматизации	13
2.16	ура! получилось	14

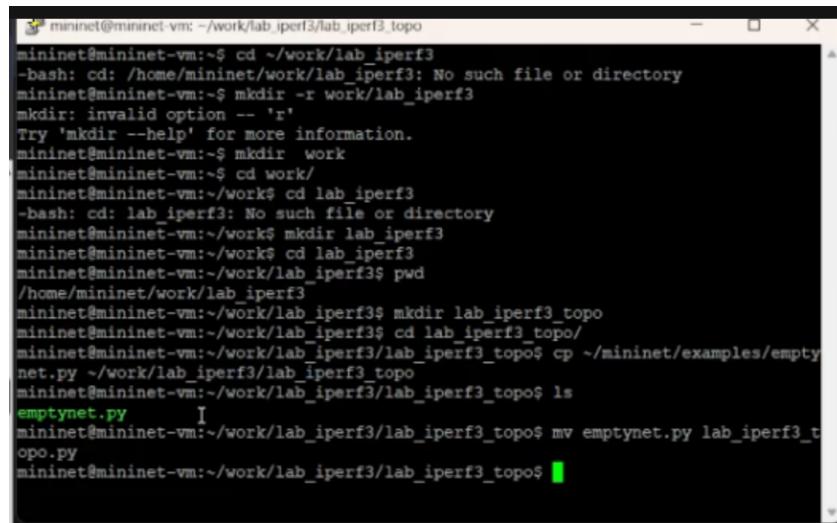
Список таблиц

1 Цель 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.

2 Выполнение лабораторной работы

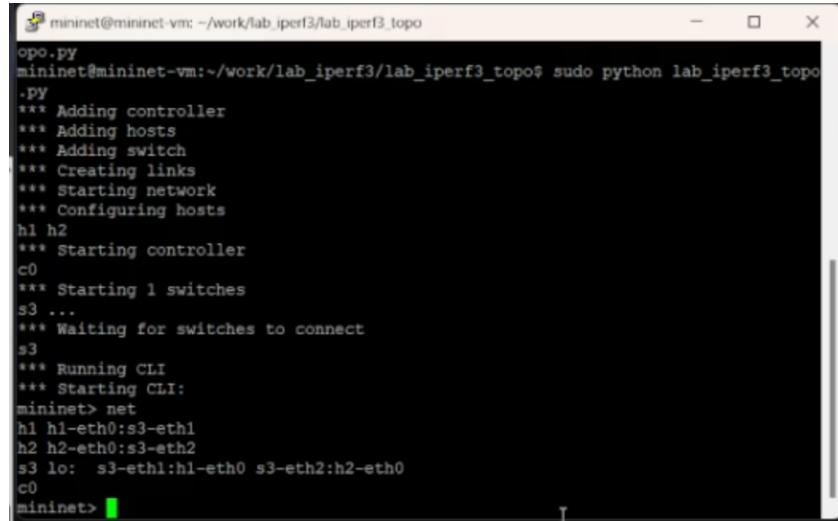
Создаем необходимые папки и копируем emptynet.py (рис. 2.1).



```
mininet@mininet-vm:~/work/lab_iperf3$ cd ~/work/lab_iperf3/lab_iperf3_topo
mininet@mininet-vm:~/work/lab_iperf3$ mkdir -r work/lab_iperf3
mkdir: invalid option -- 'r'
try 'mkdir --help' for more information.
mininet@mininet-vm:~/work/lab_iperf3$ mkdir work
mininet@mininet-vm:~/work/lab_iperf3$ cd work/
mininet@mininet-vm:~/work/lab_iperf3$ cd lab_iperf3
mininet@mininet-vm:~/work/lab_iperf3$ mkdir lab_iperf3
mininet@mininet-vm:~/work/lab_iperf3$ cd lab_iperf3
mininet@mininet-vm:~/work/lab_iperf3$ pwd
/home/mininet/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 ./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. 2.1: in dir lab_iperf3_topo

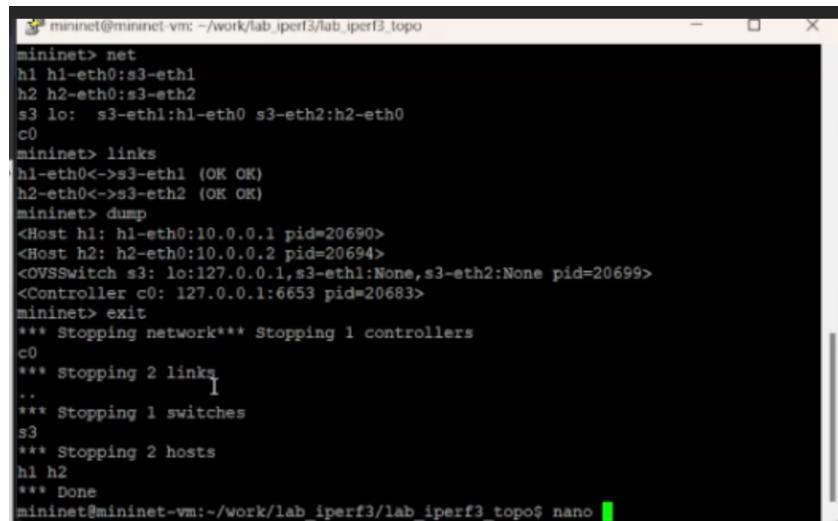
Запускаем файл (рис. 2.2).



```
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:
mininet> net
h1 h1-eth0:s3-eth1
h2 h2-eth0:s3-eth2
s3 lo: s3-eth1:h1-eth0 s3-eth2:h2-eth0
c0
mininet>
```

Рис. 2.2: Выводится консоль iperf3

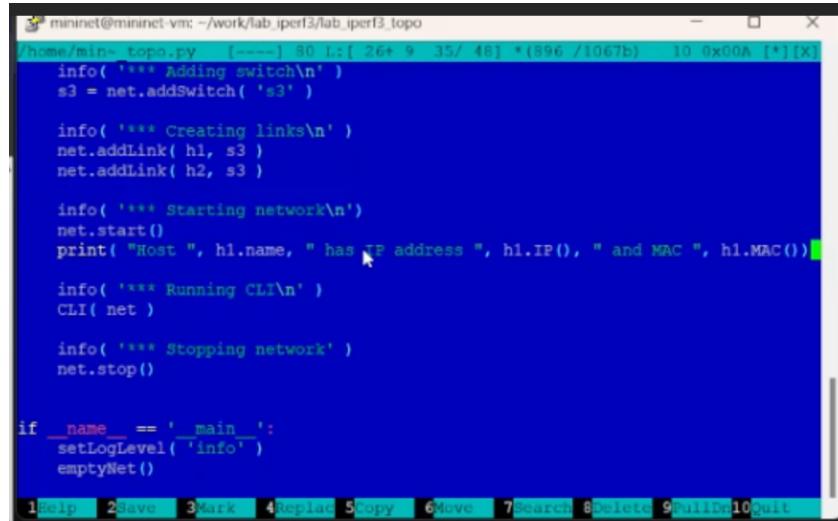
Смотрим все элементы топологии (рис. 2.3).



```
mininet> net
mininet>
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=20690>
<Host h2: h2-eth0:10.0.0.2 pid=20694>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=20699>
<Controller c0: 127.0.0.1:6653 pid=20683>
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$ nano
```

Рис. 2.3: net links dump

В коде выводим IP&MAC адрес (рис. 2.4).



```
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo
/home/mininet-topo.py [----] 80 L:[ 26+ 9 35/ 48] *(896 /1067b) 10.0x00A [*]{X}
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 ", h1.MAC() )

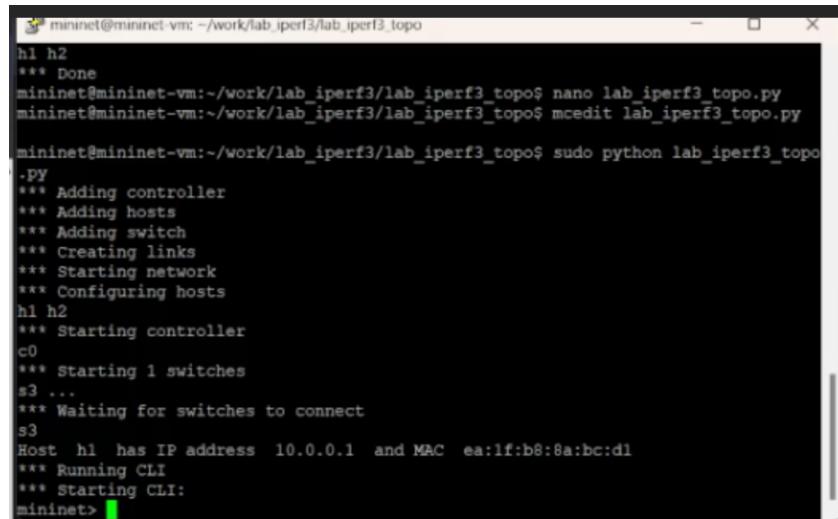
info( '*** Running CLI\n' )
CLI( net )

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

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

Рис. 2.4: via print

Запускаем отредактированную программу (рис. 2.5).



```
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ nano lab_iperf3_topo.py
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
*** Done
*** 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 ea:lf:b8:8a:bc:d1
*** Running CLI
*** Starting CLI:
mininet>
```

Рис. 2.5: наблюдаем корректное отображение для 1 хоста

Добавляем такое же отображение для 2 хоста (рис. 2.6).

```

mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo
/home/mininet-vm ~/work/lab_iperf3/lab_iperf3_topo
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 ", h1.MAC() )
print( "Also for h2: 1) IP: ", h2.IP(), " 2)MAC: ", h2.MAC() )
info( '*** Running CLI\n' )
CLI( net )

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

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

```

Рис. 2.6: тот же код только для h2

Запускаем отредактированную программу (рис. 2.7).

```

*** 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 da:15:b4:a3:3b:2f
Also for h2: 1) IP: 10.0.0.2 2)MAC: b2:7f:b0:bc:a2:a9
*** 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$ 

```

Рис. 2.7: наблюдаем корректное отображение для 2 хоста

Добавляем импорты CPULimitHost & TCLink (рис. 2.8).

```
mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo
/home/mininet-vm/mininet/mininet.py [-M--] 61 L:[ 3+18 19/ 50] *(488 /1206b) 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.node import CPULimitedHost
from mininet.node import TCLink
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' )

    1 help 2Save 3Mark 4Replace 5Copy 6Move 7Search 8Delete 9PullIn 10Quit
```

Рис. 2.8: CPULimitHost & TCLink

Расширяем net импортированными модулями (рис. 2.9).

```
mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo
/home/mininet-vm/mininet/mininet.py [-M--] 40 L:[ 6+20 26/ 51] *(672 /1244b) 39 0x027 [*]{X}
www

from mininet.net import Mininet
from mininet.node import Controller
from mininet.node import CPULimitedHost
from mininet.node import TCLink
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, 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' )
    1 help 2Save 3Mark 4Replace 5Copy 6Move 7Search 8Delete 9PullIn 10Quit
```

Рис. 2.9: в вызове Mininet

Добавляем ограничение CPU на хосты (рис. 2.10).

```

mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo
/home/mininet-vm/emptyNet.py [ -M-- ] B.L:[ 14+20 34/ 52] *(888 /1330b) 32 0x020 [*]{X}

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 )

1 Help 2Save 3Park 4Replace 5Copy 6Save 7Search 8Delete 9PullIn 10Quit

```

Рис. 2.10: ура! добавили

Запускаем код (рис. 2.11).

```

mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo
ImportError: cannot import name 'TCLink' from 'mininet.node' (/usr/local/lib/python3.8/dist-packages/mininet/node.py)
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ mcedit lab_iperf3_topo2.py

mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ sudo ./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 10000000/100000us) h2 (cfs 9000000/10000us)
*** 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 76:4e:8c:13:7a:d2
Also for h2: 1) IP: 10.0.0.2 2)MAC: c2:22:78:09:9b:0a
*** Running CLI
*** Starting CLI:
mininet>

```

Рис. 2.11: наблюдаем ограничение срн

Изменяем структуру (рис. 2.12).

```

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
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$ mininet@mininet-vm:~/work/lab_iperf3/iperf3$ ls -al
total 12
drwxrwxr-x 2 mininet mininet 4096 Oct 11 11:13 .
drwxrwxr-x 4 mininet mininet 4096 Oct 11 11:13 ..
-rwxrwxr-x 1 mininet mininet 1330 Oct 11 11:13 lab_iperf3.py
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ 

```

Рис. 2.12: переименовали / перенесли / etc

Выполняем iperf команды прямо в коде(рис. 2.13).

```

mininet@mininet-vm:~/work/lab_iperf3/iperf3
/home/miniperf3.py [-M--] 6 L:[ 27+16 45/ 55] * (1143/1276b) 67 0x043 [*]{X}
    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()

    h2.cmdPrint("iperf3 -s -D -l")
    time.sleep(10)
    h1.cmdPrint("iperf3 -c", h2.IP(), "-J > iperf_result.json")

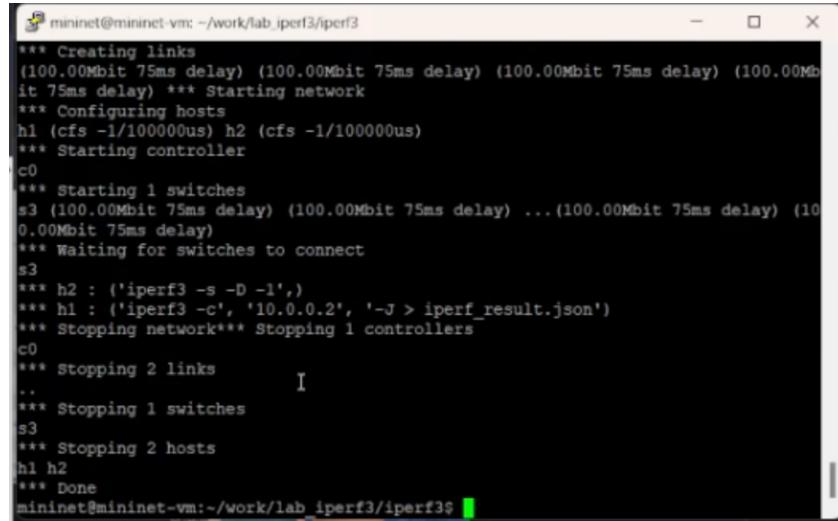
    # info( '*** Running CLI\n' )
    # CLI( net )

    info( '*** Stopping network' )
    net.stop()
1Delg 2Save 3Mark 4Replace 5Copy 6Move 7Search 8Delete 9Find 10Quit

```

Рис. 2.13: так же для h1 добавляем вывод в отчёт

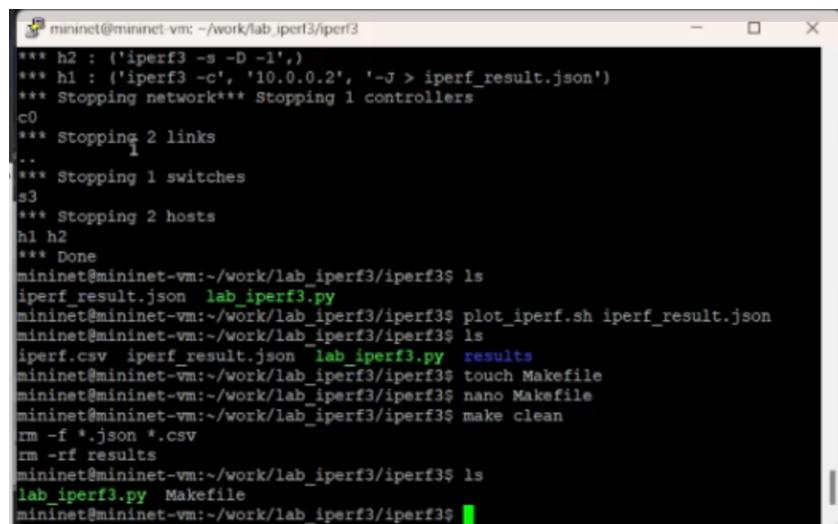
Запускаем измененный код (рис. 2.14).



```
mininet@mininet-vm: ~/work/lab_iperf3/iperf3
*** Creating links
(100.00Mbit 75ms delay) (100.00Mbit 75ms delay) (100.00Mbit 75ms delay) (100.00Mbit 75ms delay) *** Starting network
*** Configuring hosts
h1 (cfs -1/100000us) h2 (cfs -1/100000us)
*** 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
*** 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
mininet@mininet-vm:~/work/lab_iperf3/iperf3$
```

Рис. 2.14: так же видим отключенный cli

Создаем Мэйкфайл и настраиваем его (рис. 2.15).



```
mininet@mininet-vm: ~/work/lab_iperf3/iperf3
*** 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
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ ls
iperf_result.json  lab_iperf3.py
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ plot_iperf.sh iperf_result.json
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ ls
iperf.csv  iperf_result.json  lab_iperf3.py  results
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ touch Makefile
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ nano Makefile
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ make clean
rm -f *.json *.csv
rm -rf results
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ ls
lab_iperf3.py  Makefile
mininet@mininet-vm:~/work/lab_iperf3/iperf3$
```

Рис. 2.15: для автоматизации

Пробуем наш Мэйкфайл(рис. 2.16).

```
mininet@mininet-vm: ~/work/lab_iperf3/iperf3
*** Configuring hosts
h1 (cfs -1/100000us) h2 (cfs -1/100000us)
*** 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
*** 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$ ls
iperf.csv  iperf_result.json  lab_iperf3.py  Makefile  results
mininet@mininet-vm:~/work/lab_iperf3/iperf3$
```

Рис. 2.16: ура! получилось

3 Выводы

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