

Лабораторная работа № 3

**Измерение и тестирование пропускной способности сети.
Воспроизводимый эксперимент**

Оразгелдиев Язгелди

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	16

Список иллюстраций

3.1	Создание каталога и копирование файла	7
3.2	Скрипт lab_iperf3_topo	8
3.3	Запуск скрипта	8
3.4	Элементы топологии	9
3.5	Запуск скрипта	9
3.6	Внесение изменений	10
3.7	Запуск скрипта	10
3.8	Внесение изменений в скрипт	10
3.9	Внесение изменений в скрипт	11
3.10	Внесение изменений в скрипт	11
3.11	Внесение изменений в скрипт	11
3.12	Внесение изменений в скрипт	12
3.13	Запуск скрипта	12
3.14	Запуск скрипта	12
3.15	Создание подкаталога и копирование скрипта	13
3.16	Внесение изменений в скрипт	13
3.17	Внесение изменений в скрипт	13
3.18	Внесение изменений в скрипт	14
3.19	Запуск скрипта	14
3.20	Построение графика	14
3.21	Написание скрипта Makefile	15
3.22	Работа скрипта Makefile	15

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 Задание

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

3 Выполнение лабораторной работы

С помощью API Mininet создали простейшую топологию сети, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8. Для этого в каталоге /work/lab_iperf3 для работы над проектом создали подкаталог lab_iperf3_topo и скопировали в него файл с примером скрипта mininet/examples/emphynet.py, описывающего стандартную простую топологию сети mininet

```
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$ 4 cp ~/mininet/examples/emphynet.py
4: command not found
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ cp ~/mininet/examples/emphynet.py ~/work/lab_iperf3/lab_iperf3_topo
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ mv emphynet.py lab_iperf3_topo.py
```

Рисунок 3.1: Создание каталога и копирование файла

Изучил содержимое скрипта

```

/home/mininet/work/lab_iperf3/lab_iperf3_topo/lab_iperf3_topo.py  [----]  0 L: 1+10 11/ 46] *(243 / 985b) 102 0x066
#!/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()

```

Рисунок 3.2: Скрипт lab_iperf3_topo

Запустили скрипт создания топологии 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
*** Running CLI
*** Starting CLI:
mininet>

```

Рисунок 3.3: Запуск скрипта

После отработки скрипта посмотрели элементы топологии и завершили работу


```

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=6851>
<Host h2: h2-eth0:10.0.0.2 pid=6855>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=6860>
<Controller c0: 127.0.0.1:6653 pid=6844>
mininet> exit
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s3
*** Stopping 2 hosts
h1 h2
*** Done

```

Рисунок 3.4: Элементы топологии

Внес в скрипт `lab_iperf3_toro.py` изменение, позволяющее вывести на экран информацию о хосте `h1`, а именно имя хоста, его IP-адрес, MAC-адрес.

```

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()

```

Рисунок 3.5: Запуск скрипта

Изменили скрипт так чтоб на экран выводилась информация о хосте 1 и 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
Host h1 has IP address 10.0.0.1 and MAC address 22:0c:91:05:53:ef
*** Running CLI
*** Starting CLI:
mininet>
Interrupt
mininet> exit

```

Рисунок 3.6: Внесение изменений

```

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 ce:97:47:0d:66:94
Host h2 has IP address 10.0.0.2 and MAC address 32:25:ea:e4:16:88
*** Running CLI
*** Starting CLI:
mininet>

```

Рисунок 3.7: Запуск скрипта

Добавили в скрипт настройки параметров производительности. Для этого сделали 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$ ls
lab_iperf3_topo2.py  lab_iperf3_topo.py
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$

```

Рисунок 3.8: Внесение изменений в скрипт

В начале скрипта lab_iperf3_topo2 добавил записи об импорте классов CPULimitedHost и TCLink

```

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

def emptyNet():

```

Рисунок 3.9: Внесение изменений в скрипт

В скрипте lab_iperf3_toro2 изменили строку описания сети указав использование ограничения производительности и изоляции.

```

"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' )

```

Рисунок 3.10: Внесение изменений в скрипт

В скрипте lab_iperf3_toro2 изменили функцию задания параметров виртуального хоста h1, указав, что ему будет выделено 50% от общих ресурсов процессора системы. Аналогичным образом для хоста h2 задали долю выделения ресурсов процессора 45%

```

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' )

```

Рисунок 3.11: Внесение изменений в скрипт

В скрипте lab_iperf3_toro2 изменили функцию параметров соединения между хостом h1 и коммутатором 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' )

```

Рисунок 3.12: Внесение изменений в скрипт

Запустили на отработку сначала скрипт lab_iperf3_topo2 затем lab_iperf3_topo и сравнили результаты

```

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 5000000/1000000us) h2 (cfs 4500000/1000000us)
*** 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 0e:1f:d9:4d:ec:4b
Host h2 has IP address 10.0.0.2 and MAC address 76:eb:59:82:78:e5
*** Running CLI
*** Starting CLI:
mininet> exit

```

Рисунок 3.13: Запуск 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 92:dd:03:39:51:3f
Host h2 has IP address 10.0.0.2 and MAC address 36:64:5d:15:dd:56
*** Running CLI
*** Starting CLI:

```

Рисунок 3.14: Запуск скрипта

Построили графики и по проводимому эксперименту. для этого сделали копию скрипта lab_iperf3_topo2.py и поместили 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
mv: missing destination file operand after '/home/mininet/work/lab_iperf3/lab_iperf3_topo/lab_iperf3.py'
Try 'mv --help' for more information.
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 07:37 lab_iperf3.py
mininet@mininet-vm: /work/lab_iperf3/iperf3$

```

Рисунок 3.15: Создание подкаталога и копирование скрипта

В начале скрипта lab_iperf3.py добавили запись

```

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

def emptyNet():

```

Рисунок 3.16: Внесение изменений в скрипт

Изменили код в скрипте lab_iperf3.py так чтобы – на хостах не было ограничения по использованию ресурсов процессора; – каналы между хостами и коммутатором были по 100 Мбит/с с задержкой 75 мс, без потерь, без использования ограничителей пропускной способности и максимального размера очереди.

```

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' )

```

Рисунок 3.17: Внесение изменений в скрипт

После функции старта сети описали запуск на хосте h2 сервера iPerf3, а на хосте h1 запуск с задержкой в 10 секунд клиента iPerf3 с экспортом результатов в JSON-файл, закомментировали строки, отвечающие за запуск CLI-интерфейса

```

info('*** Traffic generation\n')
h2.cmdPrint('iperf3 -s -D -1')
time.sleep(10) # Wait 10 seconds for servers to start
h1.cmdPrint( 'iperf3 -c', h2.IP(), '-J > iperf_result.json' )

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 )

```

Рисунок 3.18: Внесение изменений в скрипт

Запустили на отработку скрипт 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) *** 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
*** Traffic generation
*** h2 : ('iperf3 -s -D -1',)
*** h1 : ('iperf3 -c', '10.0.0.2', '-J > iperf_result.json')
Host h1 has IP address 10.0.0.1 and MAC address 96:4b:68:73:02:c0
Host h2 has IP address 10.0.0.2 and MAC address 72:a0:65:0c:ea:20
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s3
*** Stopping 2 hosts
h1 h2
*** Done

```

Рисунок 3.19: Запуск скрипта

Создали 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$ ls
iperf.csv iperf_result.json lab_iperf3.py results

```

Рисунок 3.20: Построение графика

Создали Makefile для проведения всего эксперимента. В Makefile прописали запуск скрипта эксперимента, построение графиков и очистку каталога от результатов

```

/home/mininet/work/lab_iperf3/iperf3/Makefile [-M--]
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

```

Рисунок 3.21: Написание скрипта Makefile

Проверили корректность работы скрипта

```

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 (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
*** Traffic generation
*** h2 : ('iperf3 -s -D -i',)
*** h1 : ('iperf3 -c', '10.0.0.2', '-J > iperf_result.json')
Host h1 has IP address 10.0.0.1 and MAC address 52:dd:b0:ad:f1:f5
Host h2 has IP address 10.0.0.2 and MAC address 2e:9b:1f:ed:47:65
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
s3
*** Stopping 1 switches
s3
*** Stopping 2 hosts
h1 h2
*** Done
plot_iperf.sh iperf_result.json

```

Рисунок 3.22: Работа скрипта Makefile

Завершили соединение с виртуальной машиной и выключили ее

4 Выводы

Я 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.