

Отчет по лабораторной работе №3

Дисциплина: Моделирование сетей передачи данных

Лобанова Полина Иннокентьевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	16
	Список литературы	17

Список иллюстраций

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

1. С помощью API Mininet создала 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 создала подкаталог lab_iperf3_topo и скопировала 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$ cp ~/mininet/examples/emphynet.py ~/work/lab_iperf3/lab_iperf3_topo
mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo$ ls
emphynet.py
mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo$ mv emphynet.py lab_iperf3_topo.py
mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo$ ls
lab_iperf3_topo.py
```

Рис. 3.1: Создание каталога и копирование файла

2. Изучила содержание скрипта lab_iperf3_topo.py.

```

/home/mininet/work/lab_iperf3/lab_iperf3_topo/lab_iperf3_topo.py 985/985 100%
#!/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()

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

```

Рис. 3.2: Скриншот *lab_iperf3_topo.py*

3. Запустила 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

```

Рис. 3.3: Запуск скрипта

4. После отработки скрипта посмотрела элементы топологии и завершила 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=892>
<Host h2: h2-eth0:10.0.0.2 pid=895>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=900>
<Controller c0: 127.0.0.1:6653 pid=885>
mininet> exit
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s3
*** Stopping 2 hosts
h1 h2
*** Done

```

Рис. 3.4: Элементы топологии

- Внесла в скрипт lab_iperf3_topo.py изменение, позволяющее вывести на экран информацию о хосте h1, а именно имя хоста, его IP-адрес, MAC-адрес.

```

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()

```

Рис. 3.5: Внесение изменений в скрипт

- Проверила корректность отработки изменённого 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 76:61:fb:6e:b7:67
*** Running CLI
*** Starting CLI:
mininet>

```

Рис. 3.6: Запуск скрипта

7. Изменила скрипт `lab_iperf3_topo.py` так, чтобы на экран выводилась информация об имени, IP-адресе и MAC-адресе обоих хостов сети. Проверила корректность отработки изменённого скрипта.

```
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 )
```

Рис. 3.7: Внесение изменений в 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
Host h1 has IP address 10.0.0.1 and MAC address c2:d4:cf:16:d3:7f
Host h2 has IP address 10.0.0.2 and MAC address c2:1f:57:58:09:71
*** Running CLI
*** Starting CLI:
mininet>
```

Рис. 3.8: Запуск скрипта

8. Добавила в скрипт настройки параметров производительности. Для этого сделала 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$
```

Рис. 3.9: Внесение изменений в скрипт

9. В начале скрипта `lab_iperf3_topo2.py` добавила 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
```

Рис. 3.10: Внесение изменений в скрипт

10. В скрипте lab_iperf3_topo2.py изменила строку описания сети, указав на использование ограничения производительности и изоляции.

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

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

Рис. 3.11: Внесение изменений в скрипт

11. В скрипте lab_iperf3_topo2.py изменила функцию задания параметров виртуального хоста h1, указав, что ему будет выделено 50% от общих ресурсов процессора системы.

```
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 )
```

Рис. 3.12: Внесение изменений в скрипт

12. Аналогичным образом для хоста h2 задала долю выделения ресурсов процессора в 45%.

```
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 )
```

Рис. 3.13: Внесение изменений в скрипт

13. В скрипте lab_iperf3_topo2.py изменила функцию параметров соединения между хостом h1 и коммутатором s3.

```

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 ba:71:c1:2d:32:aa
Host h2 has IP address 10.0.0.2 and MAC address 4a:6b:a3:f5:a7:fc
*** Running CLI
*** Starting CLI:
mininet>

```

Рис. 3.14: Внесение изменений в скрипт

14. Запустила на отработку сначала скрипт lab_iperf3_topo2.py, затем lab_iperf3_topo.py и сравнила результат.

```

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 42:de:a2:64:83:ac
Host h2 has IP address 10.0.0.2 and MAC address 02:14:ad:48:3b:4b
*** Running CLI
*** Starting CLI:
mininet>

```

Рис. 3.15: Запуск скрипта

15. Построила графики по проводимому эксперименту. Для этого сделала копию скрипта lab_iperf3_topo2.py и поместила 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$ ls
lab_iperf3.py lab_iperf3_topo2.py lab_iperf3_topo.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
lab_iperf3.py
mininet@mininet-vm:~/work/lab_iperf3/iperf3$

```

Рис. 3.16: Создание подкаталога и копирование скрипта

16. В начале скрипта lab_iperf3.py добавила запись import time.

```
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
```

Рис. 3.17: Внесение изменений в скрипт

17. Изменила 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' )
```

Рис. 3.18: Внесение изменений в скрипт

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

```
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 )
```

Рис. 3.19: Внесение изменений в скрипт

19. Запустила 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) (100.00Mbit 75ms delay) **
* Starting network
*** Configuring hosts
h1 (cfs -1/1000000us) h2 (cfs -1/1000000us)
*** Starting controller
c0
*** Starting 1 switches
s3 (100.00Mbit 75ms delay) (100.00Mbit 75ms delay) ... (100.00Mbit 75ms delay) (100.00Mbit 75ms del
ay)
*** Waiting for switches to connect
s3
Host h1 has IP address 10.0.0.1 and MAC address 46:4c:cf:9c:2f:55
Host h2 has IP address 10.0.0.2 and MAC address 0e:55:9e:a4:f5:4c
*** 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
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ ls
iperf_result.json  lab_iperf3.py
mininet@mininet-vm:~/work/lab_iperf3/iperf3$
```

Рис. 3.20: Запуск скрипта

20. Построила 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
mininet@mininet-vm:~/work/lab_iperf3/iperf3$
```

Рис. 3.21: Построение графиков

21. Создала Makefile для проведения всего эксперимента. В Makefile прописала запуск скрипта эксперимента, построение графиков и очистку каталога от результатов.

```

GNU nano 4.8 /home/mininet/work/lab_iperf3/iperf3/Makefile Modified
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. 3.22: Написание скрипта Makefile

22. Проверила корректность отработки Makefile.

```

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$ 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$ 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 -1/1000000us) h2 (cfs -1/1000000us)
*** 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 fe:8a:6a:64:9f:90
Host h2 has IP address 10.0.0.2 and MAC address 7e:c2:5c:57:f2:6f
*** 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
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. 3.23: Запуск скрипта Makefile

23. Завершила соединение с виртуальной машиной mininet и выключила её.

4 Выводы

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

Список литературы