

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3

дисциплина: Моделирование сетей передачи данных

Студент: Саинт-Амур Измаэль

Группа: НПИбд-01-20

МОСКВА

2023 г.

Постановка задачи

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.

Выполнение работы

3.4. Последовательность выполнения работы

1. С помощью API Mininet создайте простейшую топологию сети, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8:

– В каталоге /work/lab_iperf3 для работы над проектом создайте подкаталог lab_iperf3_topo и скопируйте в него файл с примером скрипта mininet/examples/emptynet.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/emptynet.py
cp: missing destination file operand after '/home/mininet/mininet/examples/emptynet.py'
Try 'cp --help' for more information.
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ ↵
↵ command not found
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ cp ~/mininet/examples/emptynet.py ~/work/lab_iperf3/lab_iperf3_topo
```

:

Изучите содержание скрипта lab_iperf3_topo.py:

```

/home/mininet/work/lab_~opo/lab_iperf3_topo.py 521/985 52%
#!/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' )

```

– Запустите скрипт создания топологии lab_iperf3_topo.py:

```
1 sudo python lab_iperf3_topo.py
```

– После отработки скрипта посмотрите элементы топологии и завершите работу mininet:

```
1 mininet> net
```

```
2 mininet> links
```

```
3 mininet> dump
```

```
4 mininet> exit
```

```

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>

```

```

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=3328>
<Host h2: h2-eth0:10.0.0.2 pid=3332>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=3337>
<Controller c0: 127.0.0.1:6653 pid=3321>
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$

```

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

Для этого после строки, задающей старт работы сети, добавьте строку:

```

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

```

Здесь:

- `IP()` возвращает IP-адрес хоста или определенного интерфейса;
- `MAC()` возвращает MAC-адрес хоста или определенного интерфейса.

3. Проверьте корректность отработки изменённого скрипта.

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

5. Mininet предоставляет функции ограничения производительности и изоляции с помощью классов `CPULimitedHost` и `TCLink`. Добавьте в скрипт

настройки параметров производительности:

- Сделайте копию скрипта `lab_iperf3_topo.py`:

```

1 cp lab_iperf3_topo.py lab_iperf3_topo2.py

```

- В начале скрипта `lab_iperf3_topo2.py` добавьте записи об импорте классов `CPULimitedHost` и `TCLink`:

```

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 )

```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
 ^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line

```

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 e6:95:0b:21:38:e5
*** Running CLI
*** Starting CLI:
mininet>

```

6. Постройте графики по проводимому эксперименту: – Сделайте копию скрипта lab_iperf3_topo2.py и поместите его в подкаталог iperf:

```

/home/mininet/work/lab_iperf3/lab_iperf3_topo/lab_iperf3_topo.py Modified
"""
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.link import TCLink
import time
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' )

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_topo.py lab_iperf3_topo.py.save lab_iperf3_topo2.py
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ cp lab_iperf3_topo.py lab_iperf3_

```

Выводы

В ходе полученные навыки в измерении пропускной способности сети и проведении эксперимента в среде Mininet могут быть полезными при работе с реальными сетевыми задачами и оптимизации производительности сетей.