Лабораторная работа № 3

Измерение и тестирование пропускной способности сети. Воспроизводимый эксперимент

Абд эль хай мохамад

Содержание

Список иллюстраций

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

```python
#!/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()
```

Фигура № 1

Используя библиотеку mininet python, вы можете увидеть функцию, которая называется emptyNet(), которая начинается с создания объекта сети из класса Mininet, а затем мы начинаем определять эту сеть, назначая значения атрибуту объект net.

```
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> dump
<Host h1: h1-eth0:10.0.0.1 pid=926>
<Host h2: h2-eth0:10.0.0.2 pid=929>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=934>
<Controller c0: 127.0.0.1:6653 pid=919>
mininet> exit
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s3
*** Stopping 2 hosts
h1 h2
*** Done
```

Фигура № 2

```
*** Done
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ nvim 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 addr 10.0.0.1 and mac addr:  1e:b3:70:df:d5:57
*** Running CLI
*** Starting CLI:
mininet>
```

Фигура № 3

```
*** Done
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
*** Configuring hosts
h1 (cfs 10000000/100000us) h2 (cfs 9000000/100000us)
*** 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 addr 10.0.0.1 and mac addr:  9a:57:ca:9d:d2:be
Host h2  has IP addr 10.0.0.2 and mac addr:  26:58:e5:1f:6a:ba
*** Running CLI
*** Starting CLI:
mininet> h2 iperf3 -s &
mininet> h1 iperf3 -c h2
Connecting to host 10.0.0.2, port 5201
[  5] local 10.0.0.1 port 55924 connected to 10.0.0.2 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.06   sec   693 KBytes  5.36 Mbits/sec   31   8.48 KBytes
[  5]   1.06-2.00   sec   127 KBytes  1.11 Mbits/sec   21   1.41 KBytes
[  5]   2.00-3.01   sec  0.00 Bytes  0.00 bits/sec    1   1.41 KBytes
[  5]   3.01-4.00   sec   382 KBytes  3.15 Mbits/sec   21   8.48 KBytes
[  5]   4.00-5.00   sec   255 KBytes  2.08 Mbits/sec   19   5.66 KBytes
[  5]   5.00-6.00   sec   509 KBytes  4.17 Mbits/sec   20   4.24 KBytes
[  5]   6.00-7.00   sec   255 KBytes  2.09 Mbits/sec   16   9.90 KBytes
[  5]   7.00-8.00   sec   127 KBytes  1.04 Mbits/sec   19   2.83 KBytes
[  5]   8.00-9.00   sec   127 KBytes  1.04 Mbits/sec    8   8.48 KBytes
[  5]   9.00-10.00  sec   382 KBytes  3.13 Mbits/sec   20   2.83 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  2.79 MBytes  2.34 Mbits/sec  176             sender
[  5]   0.00-10.02  sec  2.48 MBytes  2.08 Mbits/sec                  receiver
```

Фигура № 4

```
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 addr 10.0.0.1 and mac addr:  9a:dc:ef:bf:a5:53
Host h2  has IP addr 10.0.0.2 and mac addr:  8a:c6:7e:24:bc:0a
*** Running CLI
*** Starting CLI:
mininet> h2 iperf3 -s &
mininet> h1 iperf3 -c h2
Connecting to host 10.0.0.2, port 5201
[  5] local 10.0.0.1 port 55916 connected to 10.0.0.2 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  4.64 GBytes  39.9 Gbits/sec    2   1.33 MBytes
[  5]   1.00-2.00   sec  5.36 GBytes  46.1 Gbits/sec    0   1.33 MBytes
[  5]   2.00-3.00   sec  5.33 GBytes  45.8 Gbits/sec    0   1.39 MBytes
[  5]   3.00-4.00   sec  4.46 GBytes  38.3 Gbits/sec    0   2.06 MBytes
[  5]   4.00-5.00   sec  5.17 GBytes  44.4 Gbits/sec    0   2.06 MBytes
[  5]   5.00-6.00   sec  4.60 GBytes  39.5 Gbits/sec    3   2.91 MBytes
[  5]   6.00-7.00   sec  5.12 GBytes  44.0 Gbits/sec    0   2.91 MBytes
[  5]   7.00-8.00   sec  5.23 GBytes  44.9 Gbits/sec    0   2.91 MBytes
[  5]   8.00-9.00   sec  5.24 GBytes  45.0 Gbits/sec    0   2.91 MBytes
[  5]   9.00-10.00  sec  5.20 GBytes  44.6 Gbits/sec    1   2.91 MBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  50.4 GBytes  43.3 Gbits/sec    6             sender
[  5]   0.00-10.00  sec  50.4 GBytes  43.3 Gbits/sec                  receiver
```

```
*** Starting 1 switches
s3 ...
*** Waiting for switches to connect
s3
Host h1  has IP addr 10.0.0.1 and mac addr:  b6:32:08:06:62:f1
Host h2  has IP addr 10.0.0.2 and mac addr:  82:e8:52:4d:fc:2a
*** Running CLI
*** Starting CLI:
```

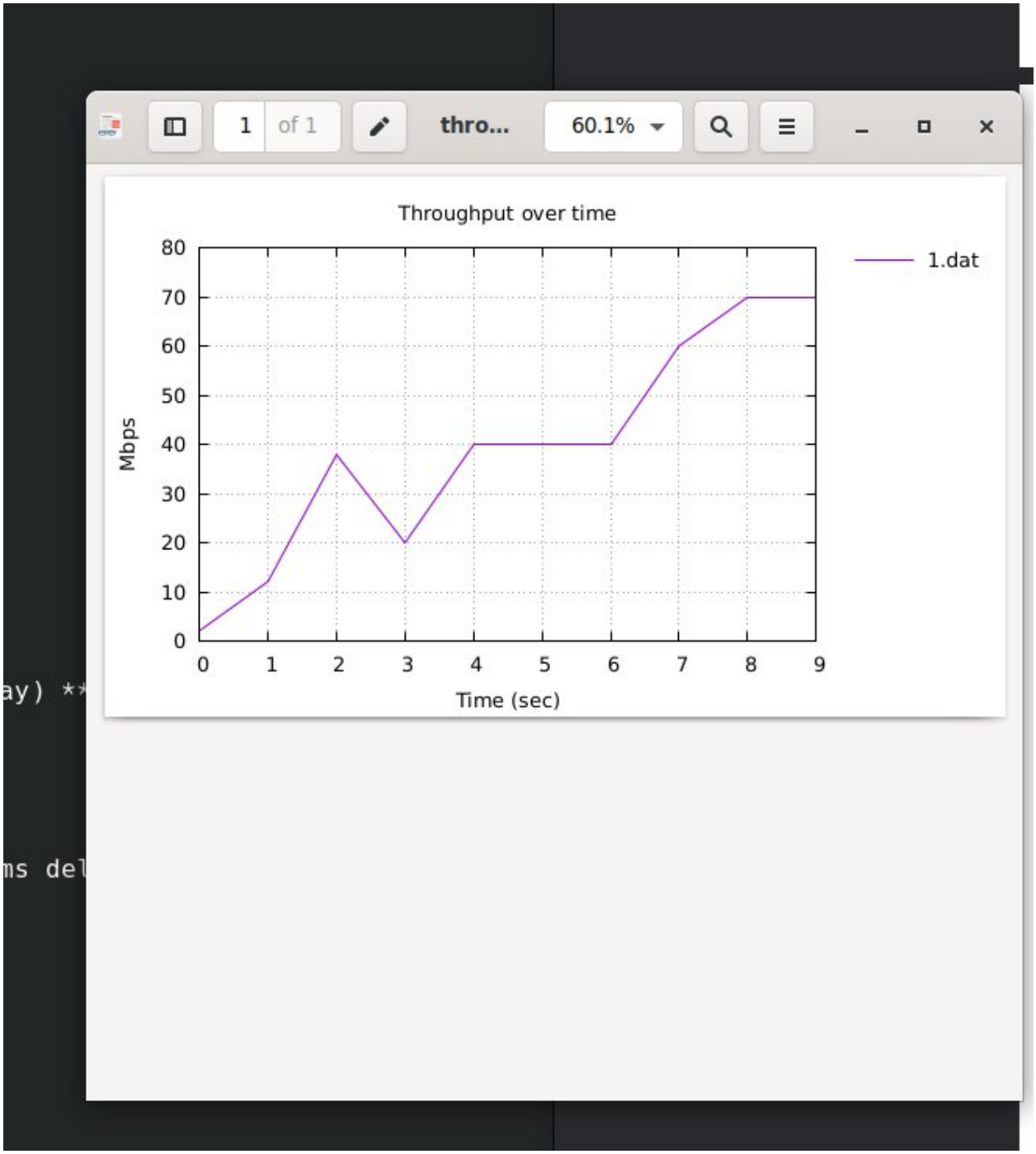
Фигура № 5

```
  net.start()
  info( '*** Starting network\n')
  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' )

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

Фигура № 6

Интегрирована команда iperf3 для запуска на клиентском терминале h1.



Фигура № 7

3. Вывод

Использовал mininet python api для создания простой сети и добавил к ней iperf3.