

Лабораторная работа № 3

Измерение и тестирование пропускной способности сети. Воспроизводимый эксперимент

Доберштейн А. С.

Российский университет дружбы народов, Москва, Россия

Информация

- Доберштейн Алина Сергеевна
- НФИбд-02-22
- Российский университет дружбы народов
- 1132226448@pfur.ru

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.

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

Выполнение лабораторной работы

С помощью API Mininet создала простейшую топологию сети.

```
mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo — ssh -Y mininet@192.168.48.5
/home/mininet/work/lab_iperf3/lab_iperf3_topo/lab_iperf3_topo.py [----] 0 L: [ 1+ 0 1/ 46] +{0 / 985b} 35 0x023
$ /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()
```

Выполнение лабораторной работы

Запустила скрипт создания топологии, посмотрела элементы топологии и завершила работу mininet.

```
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
```

Выполнение лабораторной работы

Внесла изменения в скрипт, позволяющие вывести на экран информацию о хостах h1 и h2 (IP- и MAC-адреса).

```
mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo — ssh -Y mininet@192.168.48.5  
/home/mininet/work/lab_iperf3/lab_iperf3_topo/lab_iperf3_topo.py  [-M--]  0 L:  1+47  48/ 48) *(1155/1155b) <EOF>  
#!/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()  
    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' )
```


Выполнение лабораторной работы

Проверила корректность отработки скрипта.

```
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ sudo python lab_iperf3_topo.p
*** 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 de:bb:20:42:fb:07
Host h2 has IP-address 10.0.0.2 and MAC address 66:2b:7a:34:4a:d1
*** Running CLI
```

Рис. 4: Корректность отработки скрипта

Сделала копию скрипта и в начале скрипта `lab_iperf3_topo2.py` добавила запись об импорте классов, изменила строку описания сети (добавила ограничения производительности и изоляции), функцию задания параметров виртуального хоста `h1` и `h2` (им выделено 50% от общих ресурсов процессора системы), функцию параметров соединения между `s1` и `s3`.

Выполнение лабораторной работы

```
/home/mininet/work/lab_iperf3/lab_iperf3_topo/lab_iperf3_topo2.py [-M--] 6 L: [ 1+49 50/ 56] +{(1341/1341b) 0
#!/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
from mininet.node import CPULimitedHost
from mininet.link import TCLink

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 )

    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 )

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

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

Выполнение лабораторной работы

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

```
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 1a:c1:85:ec:ab:75
Host h2 has IP-address 10.0.0.2 and MAC address 6e:dc:18:2b:c3:cd
*** Running CLI
*** Starting CLI:
mininet>
*** 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$ 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 ...
```

Сделала копию скрипта `lab_iperf3_topo2.py`, поместила его в подкаталог `iperf3`. В начале скрипта добавила запись `import time`. Изменила код в скрипте: на хостах убрала ограничение по использованию ресурсов процессора, каналы между хостами по 100Мбит/с с задержкой 75 мс, без потерь, без использования ограничителей пропускной способности и максимального размера очереди. После функции старта 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.

Выполнение лабораторной работы

```
/home/mininet/work/lab_iperf3/iperf3/lab_iperf3.py  [-M--]  1  L:  1+43  44/ 53)  *(1170/1306b)  32 0x020
#!/usr/bin/env python

"""
This example shows how to create an empty Mininet object
(without a topology object) and add nodes to it manually.
"""

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.node import CPULimitedHost
from mininet.link import TCLink

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' )

    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()
    info( '*** Starting network\n' )
    h2.cmdPrint( 'iperf3 -s -D -1' )
    time.sleep(10)
    h1.cmdPrint( 'iperf3 -c', h2.IP(), '-3 > iperf.result.json' )

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

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

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

Выполнение лабораторной работы

Запустила скрипт на отработку.

```
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 -1/100000us) 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
*** Starting network
*** 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
```

Рис. 8: Запуск скрипта

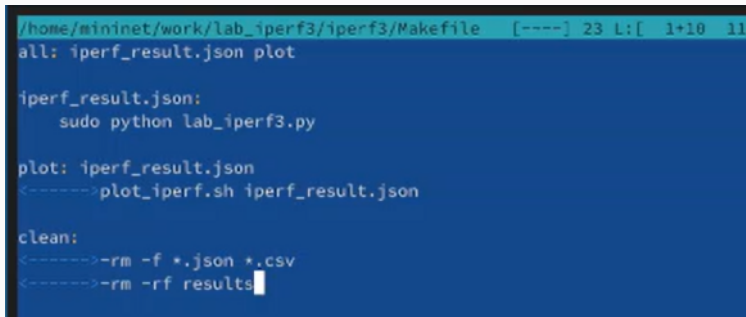
Построила графики из получившегося JSON файла.

```
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ plot_iperf.sh iperf.result.json
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ ls -l
total 20
-rw-rw-r-- 1 mininet mininet  971 Sep  9 11:08 iperf.csv
-rw-r--r-- 1 root    root    7773 Sep  9 11:07 iperf.result.json
-rwxrwxr-x 1 mininet mininet 1306 Sep  9 11:06 lab_iperf3.py
drwxrwxr-x 2 mininet mininet 4896 Sep  9 11:08 results
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ touch Makefile
```

Рис. 9: Построение графиков

Выполнение лабораторной работы

Создала Makefile для проведения всего эксперимента.



```
/home/mininet/work/lab_iperf3/iperf3/Makefile [----] 23 L:[ 1+10 11
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. 10: Makefile

Выполнение лабораторной работы

Проверила корректность отработки Makefile.

```
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
*** Starting network
*** 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
```

Рис. 11: Отработка Makefile

В ходе выполнения лабораторной работы я познакомилась 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.