

Лабораторная работа № 3

Моделирование сетей передачи данных

Доберштейн Алина Сергеевна

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выполнение лабораторной работы	7
5	Выводы	14
	Список литературы	15

Список иллюстраций

4.1	Скрипт для создания простой топологии	7
4.2	Запуск скрипта	8
4.3	Измененный скрипт	9
4.4	Корректность отработки скрипта	9
4.5	Измененный скрипт	10
4.6	Запуск скриптов	11
4.7	Измененный скрипт	12
4.8	Запуск скрипта	12
4.9	Построение графиков	13
4.10	Makefile	13
4.11	Отработка Makefile	13

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

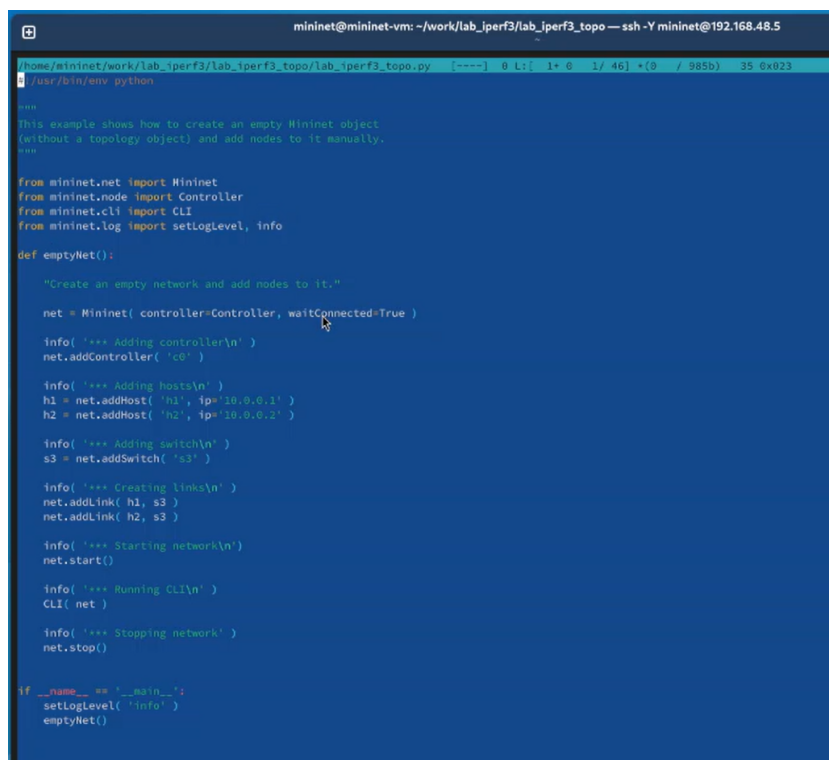
1. Воспроизвести посредством API Mininet эксперименты по измерению пропускной способности с помощью iPerf3.
2. Построить графики по проведённому эксперименту.

3 Теоретическое введение

Application Programming Interface (API) — программный интерфейс приложения, (или интерфейс программирования приложений) представляет собой специальный протокол для взаимодействия компьютерных программ, который позволяет использовать функции одного приложения внутри другого.

4 Выполнение лабораторной работы

С помощью API Mininet создала простейшую топологию сети, состоящую из двух хостов и коммутатора: в каталоге для работы над проектом создала подкаталог lab_iperf3_topo и скопировала в него файл с примером скрипта, описывающего стандартную простую топологию сети. (рис. 4.1).



```
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 6x023
/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()
```

Рис. 4.1: Скрипт для создания простой топологии

Запустила скрипт создания топологии, посмотрела элементы топологии и завершила работу mininet (рис. 4.2).

```

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
*** 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=8016>
<Host h2: h2-eth0:10.0.0.2 pid=8019>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=8024>
<Controller c0: 127.0.0.1:6653 pid=8009>
mininet> exit

```

Рис. 4.2: Запуск скрипта

Внесла изменения в скрипт, позволяющие вывести на экран информацию о хостах h1 и h2 (IP- и MAC-адреса). (рис. 4.3).


```
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' )
    CLI( net )

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

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

Рис. 4.3: Измененный скрипт

Проверила корректность отработки скрипта (рис. 4.4).

```
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-address 10.0.0.1 and MAC address de:bb:28:42:fb:07
Host h2 has IP-address 10.0.0.2 and MAC address 66:2b:7a:34:4a:d1
*** Running CLI
```

Рис. 4.4: 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.(рис. 4.5).

```
/home/mininet/work/lab_iperf3/lab_iperf3_topo/lab_iperf3_topo2.py  [-M--]  0 L:[ 1+49 50/ 50] *(1341/1341b) <
#!/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' )
    net.stop()

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

Рис. 4.5: Измененный скрипт

Запустила на отработку сначала lab_iperf3_topo2.py, затем lab_iperf3_topo.py, сравнила результат. (рис. 4.6).

```

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 ...
*** Waiting for switches to connect
s3
Host h1 has IP-address 10.0.0.1 and MAC address 72:7c:58:c0:07:f1
Host h2 has IP-address 10.0.0.2 and MAC address de:b8:3e:57:e4:d9
*** Running CLI
*** Starting CLI:
mininet>
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links

```

Рис. 4.6: Запуск скриптов

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. (рис. 4.7).

```

/home/mininet/work/lab_iperf3/iperf3/lab_iperf3.py  [-M--]  1 L:  1+43  44/ 53) *(1170/1386b)  32 0x620
#!/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 -i' )
    time.sleep(10)
    h1.cmdPrint( 'iperf3 -c', h2.IP(), '-J > iperf.result.json' )

    ...
    info( '*** Running CLI\n' )
    CLI( net )

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

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

```

Рис. 4.7: Измененный скрипт

Запустила скрипт на отработку. (рис. 4.8).

```

mininet@mininet-vm: /work/lab_iperf3/iperf3$ sudo python lab_iperf3.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating 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
*** Starting network
*** h2 : ('iperf3 -s -D -i',)
*** 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. 4.8: Запуск скрипта

Построила графики из получившегося JSON файла (рис. 4.9).

```

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 4096 Sep  9 11:08 results
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ touch Makefile

```

Рис. 4.9: Построение графиков

Создала Makefile для проведения всего эксперимента (рис. 4.10).

```

/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. 4.10: Makefile

Проверила корректность отработки Makefile (рис. 4.11).

```

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 -1/1000000us) 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
*** 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
h1 h2

```

Рис. 4.11: Отработка Makefile

5 Выводы

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.

Список литературы