

Лабораторная работа №3

Моделирование сетей передачи данных

Амуничников Антон Игоревич

2025-10-08

1. Информация

2. Вводная часть

3. Выполнение лабораторной работы

1. Информация

1.1 Докладчик

- Амуничников Антон Игоревич

1.1 Докладчик

- Амуничников Антон Игоревич
- Группа: НПИбд-01-22

1.1 Докладчик

- Амуничников Антон Игоревич
- Группа: НПИбд-01-22
- Российский университет дружбы народов им. П. Лумумбы

1.1 Докладчик

- Амуничников Антон Игоревич
- Группа: НПИбд-01-22
- Российский университет дружбы народов им. П. Лумумбы
- 1132227133@pfur.ru

2. Вводная часть

2.1 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.

2.2 Задание

1. Воспроизвести посредством API Mininet эксперименты по измерению пропускной способности с помощью iPerf3.

2.2 Задание

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

3. Выполнение лабораторной работы

3.1 Создание директорий

```
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 ~/work/lab_iperf3/lab_iperf3_topo
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ mv emptynet.py lab_iperf3_topo.py
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ nano lab_iperf3_topo.py
```

Рисунок 1: Копирование файла emptynet.py

3.2 Запуск 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
*** 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=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
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$
```

Рисунок 2: Создание топологии и ее основные параметры

3.3 Изменение lab_iperf3_toro.py

```
info( '*** Starting network\n')  
net.start()  
print( "Host", h1.name, "has IP address", h1.IP(), "and MAC address", h1.MAC() )
```

Рисунок 3: Изменение скрипта lab_iperf3_toro.py

3.4 Проверка 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 address 10.0.0.1 and MAC address 62:b5:4a:52:08:7d
*** Running CLI
*** Starting CLI:
mininet> 
```

Рисунок 4: Проверка работы внесенных изменений

3.5 Изменение lab_iperf3_topo.py

```
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' )
```

[Wrote 46 lines]

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

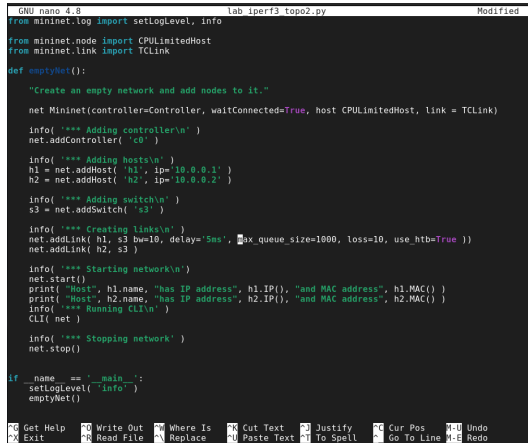
Рисунок 5: Изменение скрипта lab_iperf3_topo.py

3.6 Проверка 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 address 10.0.0.1 and MAC address 96:f4:97:8c:e6:66
Host h2 has IP address 10.0.0.2 and MAC address b6:ea:83:43:bc:ee
*** Running CLI
*** Starting CLI:
mininet> S
```

Рисунок 6: Проверка работы внесенных изменений

3.7 Изменение lab_iperf3_topo.py



```
GNU nano 4.8 lab_iperf3_topo2.py Modified
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=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()
```

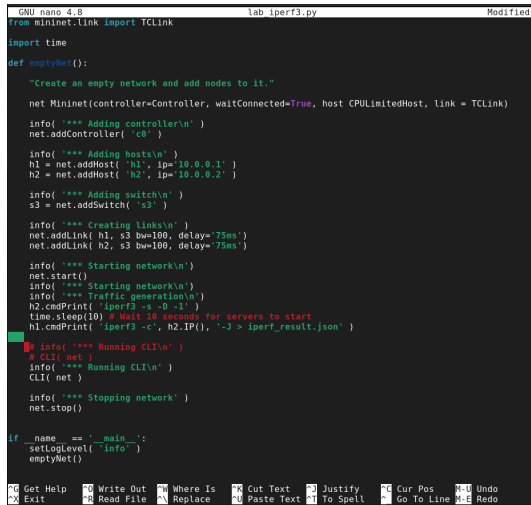
Рисунок 7: Настройка параметров производительности

3.8 Проверка lab_iperf3_topo.py

```
*** 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 72:65:e6:b3:ea:39
Host h2 has IP address 10.0.0.2 and MAC address 9e:18:b4:0b:68:f8
*** Running CLI
*** Starting CLI:
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$ 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 66:67:86:2a:0f:45
Host h2 has IP address 10.0.0.2 and MAC address 86:4f:69:3b:44:52
*** Running CLI
*** Starting CLI:
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$
```

Рисунок 8: Запуск скрипта с настройкой параметров производительности и без нее

3.9 Создание lab_iperf3.py



```
GNU nano 4.8 lab_iperf3.py Modified
from mininet.link import TCLink

import time

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' )
    info( '*** Traffic generation\n' )
    h2.cmdPrint( 'iperf3 -s -D -l' )
    time.sleep(10) # Wait 10 seconds for servers to start
    h1.cmdPrint( 'iperf3 -c', h2.IP(), '-J > iperf_result.json' )

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

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

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

Рисунок 9: Изменения кода в скрипте lab_iperf3.py

3.10 Запуск lab_iperf3.py

```
total 4
-rwxrwxr-x 1 mininet mininet 1323 Oct 11 01:29 lab_iperf3.py
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ nano lab_iperf3.py
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ nano lab_iperf3.py
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ sudo python lab_iperf3.py
  File "lab_iperf3.py", line 22
    net.Mininet(controller=Controller, waitConnected=True, host CPULimitedHost, link = TCLink)
    ^
SyntaxError: invalid syntax
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ nano lab_iperf3.py
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ sudo python lab_iperf3.py
  File "lab_iperf3.py", line 35
    net.addLink( h1, s3 bw=100, delay='75ms')
                  ^
SyntaxError: invalid syntax
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ nano 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 -1/100000us) 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')
*** Running CLI
*** Starting CLI:
mininet> exit
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s3
*** Stopping 2 hosts
h1 h2
*** Done
```

3.11 Makefile



```
GNU nano 4.8 Makefile Modified
all: iperf_result.json plot2
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
```

Рисунок 11: Создание Makefile

3.12 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 -l/100000us) h2 (cfs -l/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 -l',)
```

Рисунок 12: Проверка работы Makefile

3.13 Выводы

- В результате выполнения данной лабораторной работы я познакомился с инструментом для измерения пропускной способности сети в режиме реального времени — iPerf3, а также получил навыки проведения воспроизводимого эксперимента по измерению пропускной способности моделируемой сети в среде Mininet.