

# Лабораторная работа № 3. Измерение и тестирование пропускной способности сети. Воспроизводимый эксперимент

Моделирование сетей передачи данных

---

Демидова Е. А.

17 ноября 2024

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Демидова Екатерина Алексеевна
- студентка группы НКНбд-01-21
- Российский университет дружбы народов
- <https://github.com/eademidova>



## Введение

---

## 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.

## Задачи

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

## Выполнение лабораторной работы

---

# Создание сети с помощью API Mininet

```
mininet@mininet-vm:~/work/lab_iperf3/results$ cd ~/work/lab_iperf3
mininet@mininet-vm:~/work/lab_iperf3$ mkdir lab_iperf3_topo
mininet@mininet-vm:~/work/lab_iperf3$ cd lab_iperf3_topo/
mininet@mininet-vm:~/work/lab_iperf3_topo$ cp ~/mininet/examples/emphynet.py ~/work/lab_iperf3_topo/lab_iperf3_topo.py
mininet@mininet-vm:~/work/lab_iperf3_topo$ mv emphynet.py lab_iperf3_topo.py
mininet@mininet-vm:~/work/lab_iperf3_topo$ cat lab_iperf3_topo.py
#!/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. 1: Простейшая сеть

## Создание сети с помощью API Mininet

```
mininet@mininet-vn:~/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=4935>
<Host h2: h2-eth0:10.0.0.2 pid=4937>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=4942>
<Controller c0: 127.0.0.1:6653 pid=4928>
mininet> exit
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s3
*** Stopping 2 hosts
h1 h2
*** Done
mininet@mininet-vn:~/work/lab_iperf3/lab_iperf3_topo$
```

Рис. 2: Запуск скрипта



```
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 )

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

Рис. 3: Изменение скрипта для вывода информации о хосте h1

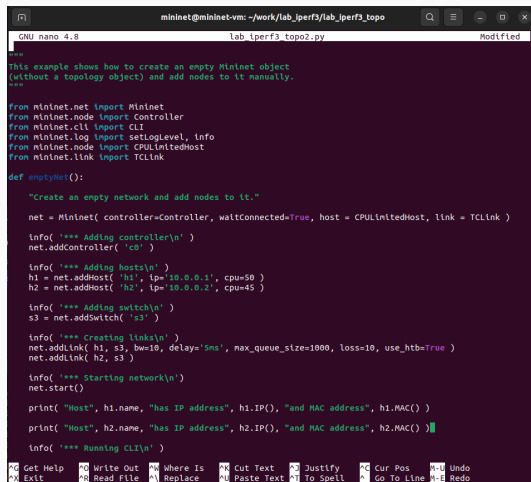
```
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ sudo python lab_iperf3_topo.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
*** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s3 ...
*** Waiting for switches to connect
s3
Host h1 has IP address 10.0.0.1 and MAC address 7a:86:57:9c:a7:d2
*** Running CLI
*** Starting CLI:
```

Рис. 4: Запуск скрипта

```
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ nano 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 ee:97:b2:93:70:7f
Host h2 has IP address 10.0.0.2 and MAC address ce:77:ff:94:c6:e4
*** Running CLI
*** Starting CLI:
mininet> 
```

Рис. 5: Изменение скрипта для вывода информации о хосте h1

# Создание сети с помощью API Mininet



```
mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo
GNU nano 4.8 lab_iperf3_topo2.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.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' )

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

Рис. 6: Добавление в скрипт настроек параметров производительности

# Создание сети с помощью API Mininet

```
mininet@mininet-vm: ~/work/lab_iperf3/iperf3
GNU nano 4.8                               Lab iperf3.py
(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' )

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

    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 )

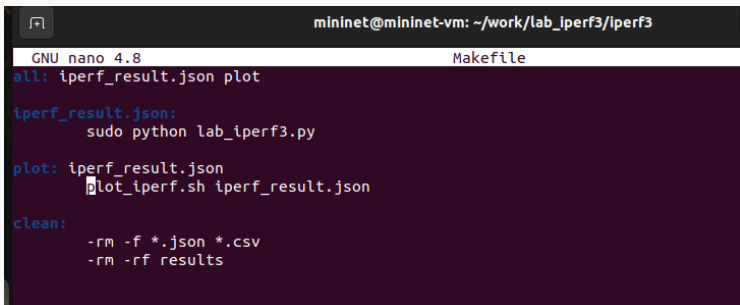
    info( '*** Stopping network' )
    net.stop()
```

Рис. 7: Изменение скрипта

## Создание сети с помощью API Mininet

```
mininet@mininet-vn: /work/lab_iperf3/lab_iperf3_topo$ cp lab_iperf3_topo.py lab_iperf3_topo2.py
mininet@mininet-vn: /work/lab_iperf3/lab_iperf3_topo$ nano lab_iperf3_topo2.py
mininet@mininet-vn: /work/lab_iperf3/lab_iperf3_topo$ cp lab_iperf3_topo2.py lab_iperf3.py
mininet@mininet-vn: /work/lab_iperf3/lab_iperf3_topo$ mkdir -p ~/work/lab_iperf3/iperf3
mininet@mininet-vn: /work/lab_iperf3/lab_iperf3_topo$ mv ~/work/lab_iperf3/lab_iperf3_topo/lab_iperf3
.py ~/work/lab_iperf3/iperf3
mininet@mininet-vn: /work/lab_iperf3/lab_iperf3_topo$ cd ~/work/lab_iperf3/iperf3
mininet@mininet-vn: /work/lab_iperf3/iperf3$ ls -l
total 4
-rwxrwxr-x 1 mininet mininet 1348 Nov 17 03:43 lab_iperf3.py
mininet@mininet-vn: /work/lab_iperf3/iperf3$ nano lab_iperf3.py
mininet@mininet-vn: /work/lab_iperf3/iperf3$ nano lab_iperf3.py
mininet@mininet-vn: /work/lab_iperf3/iperf3$
mininet@mininet-vn: /work/lab_iperf3/iperf3$ mininet@mininet-vn: /work/lab_iperf3/iperf3$ nano lab_ip
erf3.py
-bash: mininet@mininet-vn: /work/lab_iperf3/iperf3$: No such file or directory
mininet@mininet-vn: /work/lab_iperf3/iperf3$ plot_iperf.sh iperf_result.json
Error: iperf_result.json is not a file. Quitting...
mininet@mininet-vn: /work/lab_iperf3/iperf3$ sudo python lab_iperf3.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
(100.00Mbit 75ns delay) (100.00Mbit 75ns delay) (100.00Mbit 75ns delay) (100.00Mbit 75ns delay) *** S
tarting network
*** Configuring hosts
h1 (cfs -1/1000000us) h2 (cfs -1/1000000us)
*** Starting controller
c0
*** Starting 1 switches
s3 (100.00Mbit 75ns delay) (100.00Mbit 75ns delay) ... (100.00Mbit 75ns delay) (100.00Mbit 75ns 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')
Host h1 has IP address 10.0.0.1 and MAC address d6:d2:b7:f1:d1:12
Host h2 has IP address 10.0.0.2 and MAC address 9a:70:7a:33:67:b7
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s3
*** Stopping 2 hosts
h1 h2
*** Done
mininet@mininet-vn: /work/lab_iperf3/iperf3$ plot_iperf.sh iperf_result.json
mininet@mininet-vn: /work/lab_iperf3/iperf3$ touch Makefile
mininet@mininet-vn: /work/lab_iperf3/iperf3$
```

Рис. 8: Запуск скрипта



The screenshot shows a terminal window with the title bar "mininet@mininet-vm: ~/work/lab\_iperf3/iperf3". The terminal is running the GNU nano 4.8 text editor, editing a file named "Makefile". The content of the Makefile is as follows:

```
GNU nano 4.8 Makefile
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. 9: Создание Makefile

## Создание сети с помощью API Mininet

```
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) *** S
tarting 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')
Host h1 has IP address 10.0.0.1 and MAC address 4a:34:98:34:85:93
Host h2 has IP address 10.0.0.2 and MAC address 3a:1c:38:87:b4:8a
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s3
*** Stopping 2 hosts
h1 h2
*** Done
plot_iperf.sh iperf_result.json
mininet@mininet-vm:~/work/lab_iperf3/iperf3$
```

Рис. 10: Изменение протокола передачи



## Выводы

---

В результате выполнения работы познакомились 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. Mininet [Электронный ресурс]. Mininet Project Contributors. URL: <http://mininet.org/> (дата обращения: 11.12.2024).
2. IPerf [Электронный ресурс]. URL: <https://iperf.fr/>.