

Отчёт по лабораторной работе №3 по дисциплине Моделирование сетей передачи данных

**Измерение и тестирование пропускной способности сети.
Воспроизводимый эксперимент**

Шаповалова Диана Дмитриевна, НПИбд-02-21, 1032211220

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	19
4	Список литературы	20

Список иллюстраций

2.1	Создаем подкаталог	7
2.2	Содержание скрипта lab_iperf3_topo.py	8
2.3	Запускаем скрипт, смотрим элементы топологии	9
2.4	Проверяем корректность отработки изменённого скрипта	11
2.5	Добавляем в скрипт настройки параметров производительности	13
2.6	Запускаем скрипт lab_iperf3_topo2.py	14
2.7	Копируем скрипт lab_iperf3_topo2.py и помещаем его в подкаталог iperf	15
2.8	Меняем код в скрипте lab_iperf3.py	15
2.9	Запускаем скрипт lab_iperf3.py	16
2.10	Прописываем запуск скрипта эксперимента, построение графиков и очистку каталога от результатов	17
2.11	Проверяем корректность отработки Makefile	18

Список таблиц

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

1. С помощью API Mininet создайте простейшую топологию сети, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8:

– В каталоге `/work/lab_iperf3` для работы над проектом создайте подкаталог `lab_iperf3_topo` и скопируйте в него файл с примером скрипта `mininet/examples/emptynet.py`, описывающего стандартную простую топологию сети mininet:

```
mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo
login as: mininet
mininet@192.168.56.104's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-42-generic x86_64)

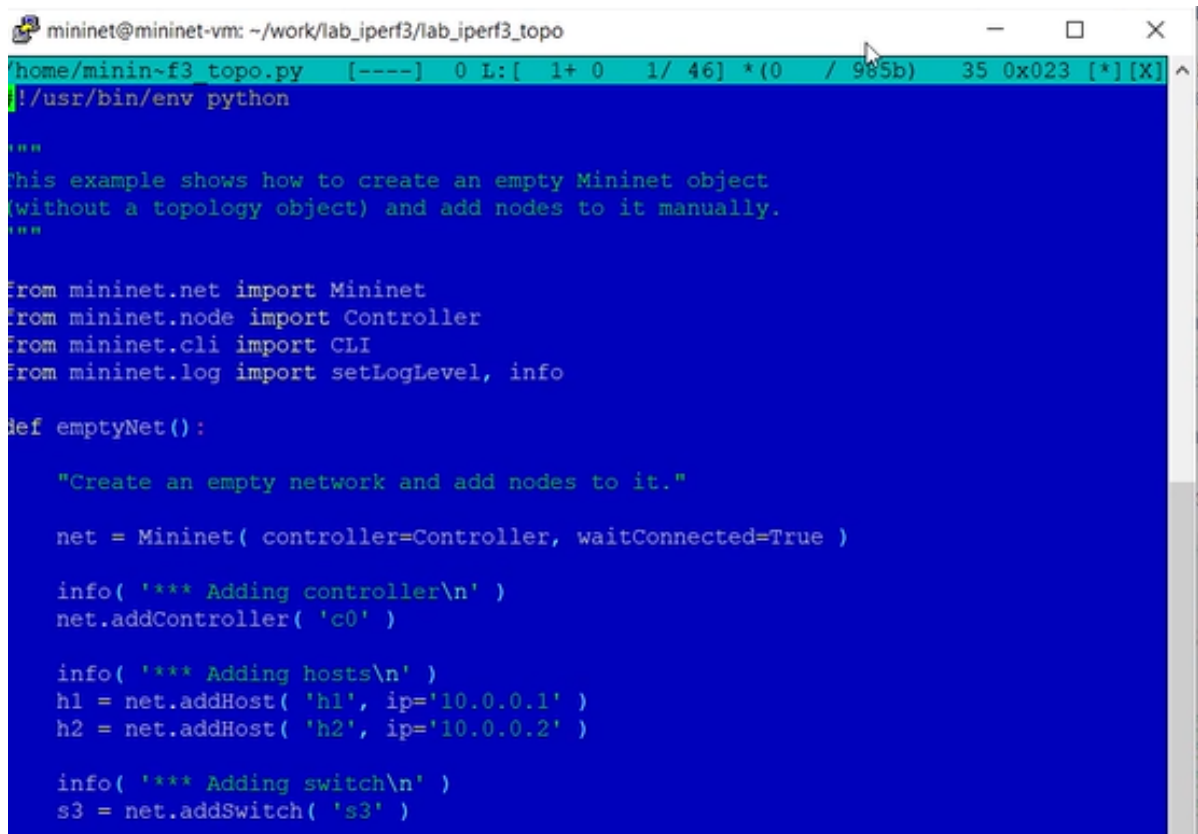
 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

New release '22.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Nov 27 04:19:07 2024
mininet@mininet-vm:~$ cd ~/work/lab_iperf3
mininet@mininet-vm:~/work/lab_iperf3$ mkdir lab_iperf3_topo
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ cd ~/work/lab_iperf3/lab_iperf3_topo
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ cp ~/mininet/examples/empty
net.py
cp: missing destination file operand after '/home/mininet/mininet/examples/emptyn
et.py'
Try 'cp --help' for more information.
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ cp ~/mininet/examples/empty
net.py ~/work/lab_iperf3/lab_iperf3_topo
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ mv emptynet.py lab_iperf3_t
opo.py
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ ls
lab_iperf3_topo.py
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ mc lab_iperf_topo.py
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ mcedit lab_iperf_topo.py
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ mcedit lab_iperf_topo.py
```

Рис. 2.1: Создаем подкаталог

– Изучите содержание скрипта lab_iperf3_topo.py:



```
mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo
/home/mininet-f3_topo.py  [----]  0 L:[ 1+ 0  1/ 46] *(0 / 985b) 35 0x023 [*][X] ^
#!/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' )
```

Рис. 2.2: Содержание скрипта lab_iperf3_topo.py

Основные элементы:

- addSwitch(): добавляет коммутатор в топологию и возвращает имя коммутатора;
- addHost(): добавляет хост в топологию и возвращает имя хоста;
- addLink(): добавляет двунаправленную ссылку в топологию (и возвращает ключ ссылки; ссылки в Mininet являются двунаправленными, если не указано иное);
- Mininet: основной класс для создания и управления сетью;
- start(): запускает сеть;
- pingAll(): проверяет подключение, пытаясь заставить все узлы пинговать друг друга;
- stop(): останавливает сеть;
- net.hosts: все хосты в сети;

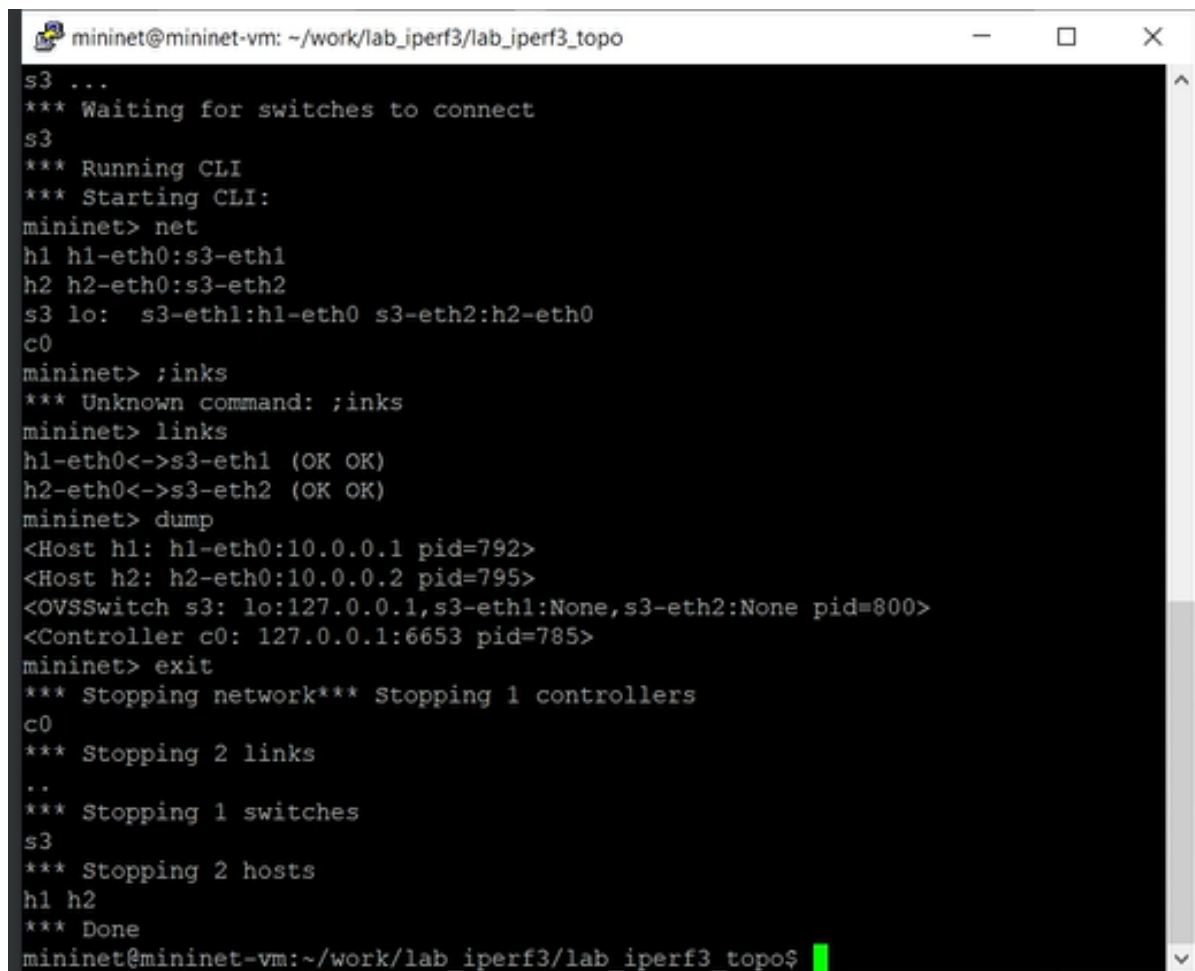
- `dumpNodeConnections()`: сбрасывает подключения к/от набора узлов;
- `setLogLevel('info' | 'debug' | 'output')`: устанавливает уровень вывода Mininet по умолчанию; рекомендуется `info`.

Запустите скрипт создания топологии `lab_iperf3_topo.py`:

```
sudo python lab_iperf3_topo.py
```

После отработки скрипта посмотрите элементы топологии и завершите работу mininet:

```
mininet> net
mininet> links
mininet> dump
mininet> exit
```



```
mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo
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> ;inks
*** Unknown command: ;inks
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=792>
<Host h2: h2-eth0:10.0.0.2 pid=795>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=800>
<Controller c0: 127.0.0.1:6653 pid=785>
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$
```

Рис. 2.3: Запускаем скрипт, смотрим элементы топологии

2. Внесите в скрипт `lab_iperf3_toro.py` изменение, позволяющее вывести на экран информацию о хосте `h1`, а именно имя хоста, его IP-адрес, MAC-адрес. Для этого после строки, задающей старт работы сети, добавьте строку:

```
print( "Host", h1.name, "has IP address", h1.IP(), "and MAC address", h1.MAC() )
```

Здесь:

- `IP()` возвращает IP-адрес хоста или определенного интерфейса;
- `MAC()` возвращает MAC-адрес хоста или определенного интерфейса.

3. Проверьте корректность отработки изменённого скрипта.
4. Измените скрипт `lab_iperf3_toro.py` так, чтобы на экран выводилась информация об имени, IP-адресе и MAC-адресе обоих хостов сети. Проверьте корректность отработки изменённого скрипта.

```
mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo
*** Done
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ mcedit 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 9e:8f:21:5d:54:14
Host h2 has IP address 10.0.0.2 and MAC address 72:4b:3d:59:29:df
*** 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$ cp lab_iperf3_topo.py lab_iperf3_topo2.py
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ mcedit lab_iperf3_topo2.py
```

Рис. 2.4: Проверяем корректность отработки изменённого скрипта

5. Mininet предоставляет функции ограничения производительности и изоляции с помощью классов `CPULimitedHost` и `TCLink`. Добавьте в скрипт настройки параметров производительности:

– Сделайте копию скрипта `lab_iperf3_topo.py`:

```
cp lab_iperf3_topo.py lab_iperf3_topo2.py
```

– В начале скрипта `lab_iperf3_topo2.py` добавьте записи об импорте классов `CPULimitedHost` и `TCLink`:

```
from mininet.node import CPULimitedHost
```

```
from mininet.link import TCLink
```

– В скрипте lab_iperf3_topo2.py измените строку описания сети, указав на использование ограничения производительности и изоляции:

```
net = Mininet( controller=Controller, waitConnected=True, host = CPULimitedHost,  
link = TCLink )
```

– В скрипте lab_iperf3_topo2.py измените функцию задания параметров виртуального хоста h1, указав, что ему будет выделено 50% от общих ресурсов процессора системы:

```
h1 = net.addHost( 'h1', ip='10.0.0.1', cpu=50 )
```

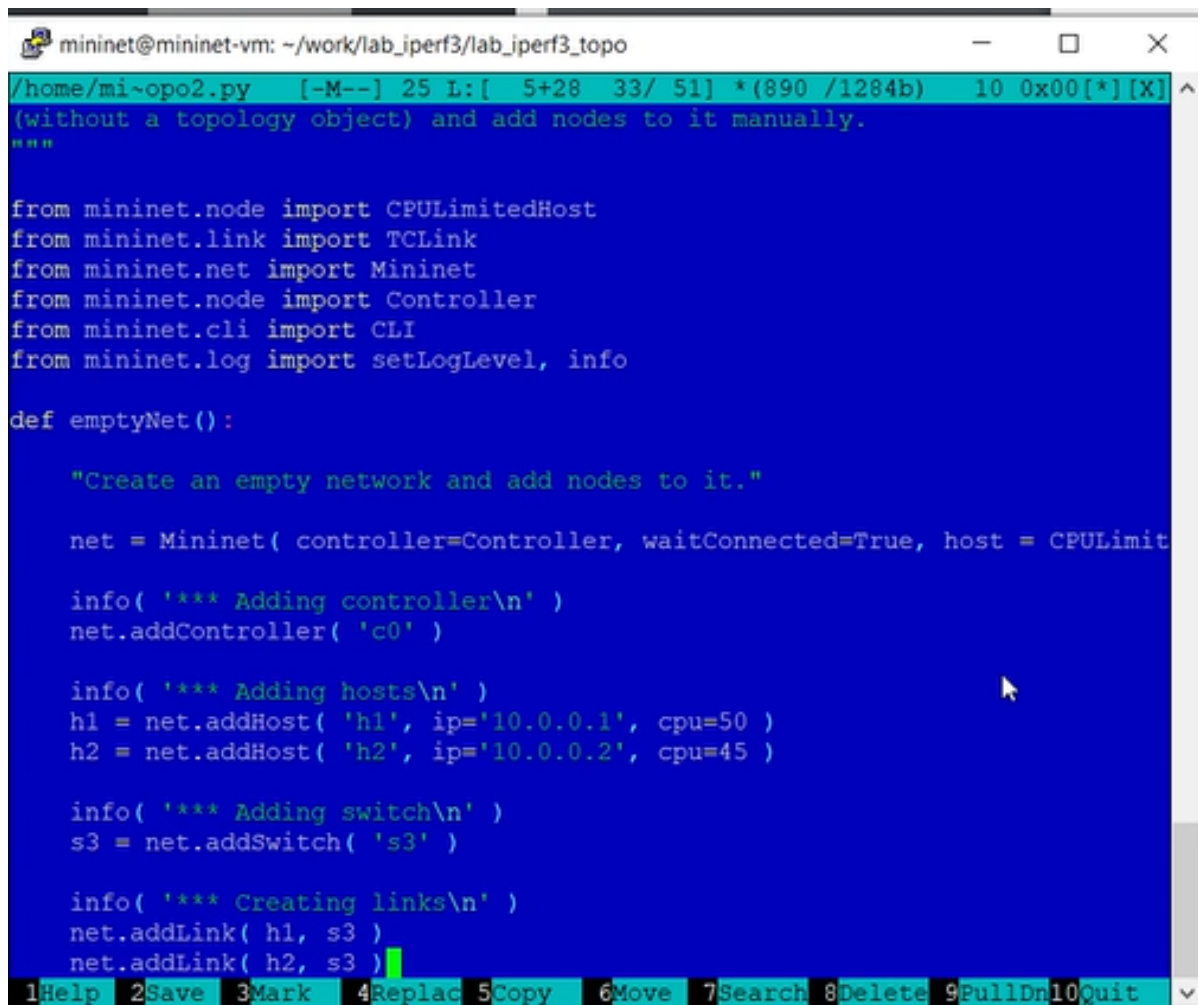
– Аналогичным образом для хоста h2 задайте долю выделения ресурсов процессора в 45%.

– В скрипте lab_iperf3_topo2.py измените функцию параметров соединения между хостом h1 и коммутатором s3:

```
net.addLink( h1, s3, bw=10, delay='5ms', max_queue_size=1000, loss=10,  
use_htb=True )
```

Здесь добавляется двунаправленный канал с характеристиками пропускной способности, задержки и потерь:

- параметр пропускной способности (bw) выражается числом в Мбит;
- задержка (delay) выражается в виде строки с заданными единицами измерения (например, 5ms, 100us, 1s);
- потери (loss) выражаются в процентах (от 0 до 100);
- параметр максимального значения очереди (max_queue_size) выражается в пакетах;
- параметр use_htb указывает на использование ограничителя интенсивности входящего потока Hierarchical Token Bucket (HTB).



```
mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo
/home/mi~opo2.py  [-M--] 25 L:[ 5+28 33/ 51] *(890 /1284b) 10 0x00[*][X] ^
(without a topology object) and add nodes to it manually.
"""

from mininet.node import CPULimitedHost
from mininet.link import TCLink
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, host = CPULimit

    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 )
    net.addLink( h2, s3 )

1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn10Quit
```

Рис. 2.5: Добавляем в скрипт настройки параметров производительности

– Запустите на отработку скрипт lab_iperf3_topo2.py

```
mininet@mininet-vm: ~/work/lab_iperf3/lab_iperf3_topo
SyntaxError: unmatched ') '
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ mcedit lab_iperf3_topo2
.py

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 5000000/100000us) h2 (cfs 4500000/100000us)
*** Starting controller
c0
*** Starting 1 switches
s3 (10.00Mbit 5ms delay 10.00000% loss) ... (10.00Mbit 5ms delay 10.00000% los
s)
*** Waiting for switches to connect
s3
Host h1 has IP address 10.0.0.1 and MAC address 12:6d:1d:e0:03:63
Host h2 has IP address 10.0.0.2 and MAC address de:69:ef:b7:ee:62
*** Running CLI
*** Starting CLI:
mininet> exit
*** Stopping network*** Stopping 1 controllers
c0
(cfs -1/100000us) (cfs -1/100000us) *** Stopping 2 links
..
*** Stopping 1 switches
s3
*** Stopping 2 hosts
h1 h2
*** Done
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$
```

Рис. 2.6: Запускаем скрипт lab_iperf3_topo2.py

6. Постройте графики по проводимому эксперименту:

– Сделайте копию скрипта lab_iperf3_topo2.py и поместите его в подкаталог iperf:

```
cp lab_iperf3_topo2.py lab_iperf3.py
```

```
mkdir -p ~/work/lab_iperf3/iperf3
```

```
mv ~/work/lab_iperf3/lab_iperf3_topo/lab_iperf3.py ~/work/lab_iperf3/iperf3
```

```
cd ~/work/lab_iperf3/iperf3
```

ls -l

```
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$  
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ mkdir -p ~/work/lab_iperf3/iperf3  
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ mv ~/work/lab_iperf3/lab_iperf3_topo/lab_iperf3.py ~/work/lab_iperf3/iperf3  
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ cd ~/work/lab_iperf3/iperf3  
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ ls -l  
total 4  
-rwxrwxr-x 1 mininet mininet 1346 Nov 27 01:41 lab_iperf3.py  
mininet@mininet-vm:~/work/lab_iperf3/iperf3$
```

Рис. 2.7: Копируем скрипт lab_iperf3_topo2.py и помещаем его в подкаталог iperf

- В начале скрипта lab_iperf3.py добавьте запись
`import time`
- Измените код в скрипте lab_iperf3.py так, чтобы:
 - на хостах не было ограничения по использованию ресурсов процессора;
 - каналы между хостами и коммутатором были по 100 Мбит/с с задержкой 75 мс, без потерь, без использования ограничителей пропускной способности и максимального размера очереди
- После функции старта сети опишите запуск на хосте h2 сервера iPerf3, а на хосте h1 запуск с задержкой в 10 секунд клиента iPerf3 с экспортом результатов в JSON-файл, прокомментируйте строки, отвечающие за запуск CLI-интерфейса

```
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( '*** Traffic generation\n' )  
h2.cmdPrint( 'iperf3 -s -D -1' )  
time.sleep(10) #Wait 10 sec for servers to start  
h1.cmdPrint( 'iperf3 -c', h2.IP(), '-J > iperf_result.json' )
```

Рис. 2.8: Меняем код в скрипте lab_iperf3.py

- Запустите на отработку скрипт lab_iperf3.py:

```
sudo python 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 5000000/100000us) h2 (cfs 4500000/100000us)
*** 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
*** Traffic generation
*** h2 : ('iperf3 -s -D -1',)
*** h1 : ('iperf3 -c', '10.0.0.2', '-J > iperf_result.json')
*** Running CLI
*** Starting CLI:
```

Рис. 2.9: Запускаем скрипт lab_iperf3.py

- Постройте графики из получившегося JSON-файла:
`plot_iperf.sh iperf_result.json`
- Создайте Makefile для проведения всего эксперимента:
`touch Makefile`
- В Makefile пропишите запуск скрипта эксперимента, построение графиков и очистку каталога от результатов:


```
mininet@mininet-vm: ~/work/lab_iperf3/iperf3
/home/mininet/.makefile [-M--] 0 L:[ 1+11 12/ 12] *(179 / 179b) <f
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. 2.10: Прописываем запуск скрипта эксперимента, построение графиков и очистку каталога от результатов

– Проверьте корректность отработки Makefile:

make clean

make

```
mininet@mininet-vm: ~/work/lab_iperf3/iperf3
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 5000000/1000000us) h2 (cfs 4500000/1000000us)
*** 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
*** 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
(cfs -1/1000000us) (cfs -1/1000000us) *** 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. 2.11: Проверяем корректность отработки Makefile

3 Выводы

Мы познакомились с инструментом для измерения пропускной способности сети в режиме реального времени — iPerf3, а также получили навыки проведения воспроизводимого эксперимента по измерению пропускной способности моделируемой сети в среде Mininet.

4 Список литературы

[1] Mininet: <https://mininet.org/>