

# Лабораторная работа №3 Дисциплина: Моделирование сетей передачи данных

Леснухин Даниил Дмитриевич Российский университет  
дружбы народов Москва

# Цель работы

Основной целью работы является:

- Знакомство с инструментом iPerf3
- Проведение воспроизводимого эксперимента в Mininet
- Изучение анализа и построения графиков по результатам

# Задание

- Воспроизвести эксперименты через API Mininet
- Построить графики по результатам эксперимента

# Теоретическое введение

- Mininet — эмулятор компьютерных сетей
- Поддержка хостов, коммутаторов и OpenFlow-контроллеров
- Возможность тестирования сетевых протоколов и анализа задержек

iPerf3 — клиент-серверное приложение для измерения пропускной способности:

- TCP/SCTP: пропускная способность, MSS/MTU, CWnd
- UDP: пропускная способность, потери пакетов, jitter, multicast

# Создание простейшей топологии

С помощью API Mininet создаем два хоста и один коммутатор (10.0.0.0/8).

```
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ echo emptynet.py
emptynet.py
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$ ls
lab_iperf3_topo.py
mininet@mininet-vm:~/work/lab_iperf3/lab_iperf3_topo$ |
```

# Изучение скрипта `topologie`

Скрипт `lab_iperf3_topo.py` содержит:

- `addSwitch()`, `addHost()`, `addLink()`
- Класс `Mininet` и методы `start()`, `pingAll()`, `stop()`
- `dumpNodeConnections()`, `setLogLevel()`

```
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> links
h1-eth0<->s3-eth1 (OK OK)
h2-eth0<->s3-eth2 (OK OK)
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=7221>
<Host h2: h2-eth0:10.0.0.2 pid=7223>
```

# Изменение скрипта

Добавляем вывод информации о хостах (имя, IP, MAC).

```
mininet@mininet-vm: ~/work, x + v
/home/mininet/work/lab_iperf3/lab_iperf3_topo/lab_iperf3_topo.py [-----] 75 L: [ 1+35 36/ 48] *(976 /115)
#!/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' )
```

# Настройка ресурсов хостов

- h1: 50% CPU
- h2: 45% CPU

Настройка двунаправленного канала с параметрами:

- Пропускная способность (bw)
- Задержка (delay)
- Потери (loss)
- Размер очереди (max\_queue\_size)
- use\_htb

```
/home/mininet/work/lab_iperf3/lab_iperf3_topo/lab_iperf3_topo2.py  [-M--] 89 L:[ 1+31 32/ 50] *(926
#!/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.node import CPULimitedHost
from mininet.link import TCLink

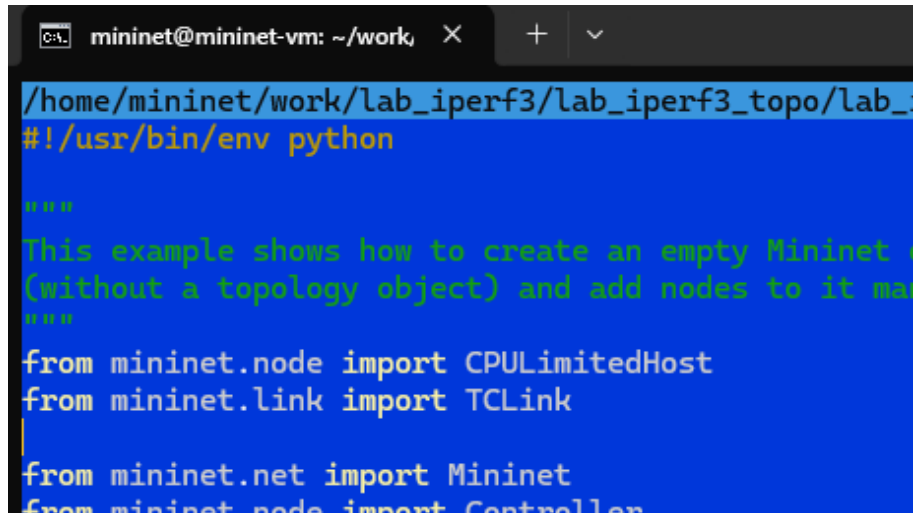
from mininet.net import Mininet
from mininet.node import Controller
from mininet.cli import CLI
from mininet.log import setLogLevel, info

def emptyNet():
```



# Импорт классов CPULimitedHost и TCLink

Добавляем в начале скрипта для ограничения ресурсов и настройки каналов.

A screenshot of a terminal window with a dark grey title bar. The title bar contains a terminal icon, the text 'mininet@mininet-vm: ~/work', and window control buttons (close, maximize, and a dropdown arrow). The terminal has a blue background and displays a Python script. The script starts with a shebang line, followed by a docstring that explains the example. Then, it imports 'CPULimitedHost' and 'TCLink' from 'mininet.node', and 'Mininet' from 'mininet.net'.

```
mininet@mininet-vm: ~/work × + ▾  
/home/mininet/work/lab_iperf3/lab_iperf3_topo/lab_  
#!/usr/bin/env python  
  
"""  
This example shows how to create an empty Mininet  
(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
```

# Запуск скрипта эксперимента

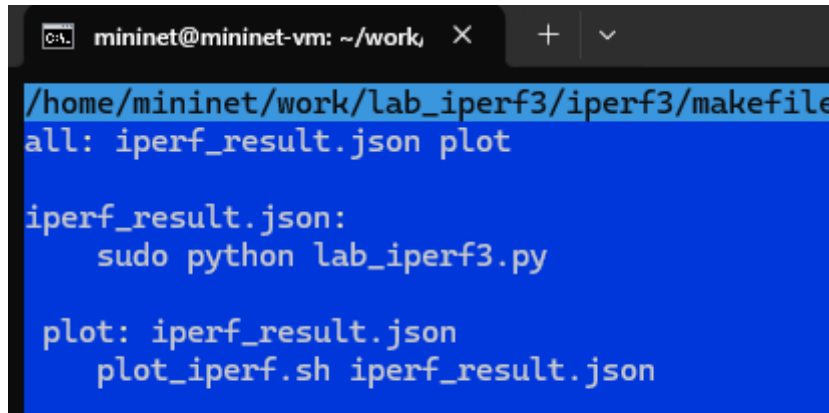
Запускаем `lab_iperf3.py` и наблюдаем результаты.

```
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) *** 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
*** 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 5a:c9:ba:f8:43:ee
Host h2 has IP address 10.0.0.2 and MAC address fe:e9:4d:c5:81:c5
*** Running CLI
*** Starting CLI:
mininet> |
```

# Создание Makefile

Makefile для автоматизации:

- Запуск эксперимента
- Построение графиков
- Очистка результатов

A terminal window with a dark background and a blue prompt. The window title is 'mininet@mininet-vm: ~/work'. The terminal shows a Makefile with the following content:

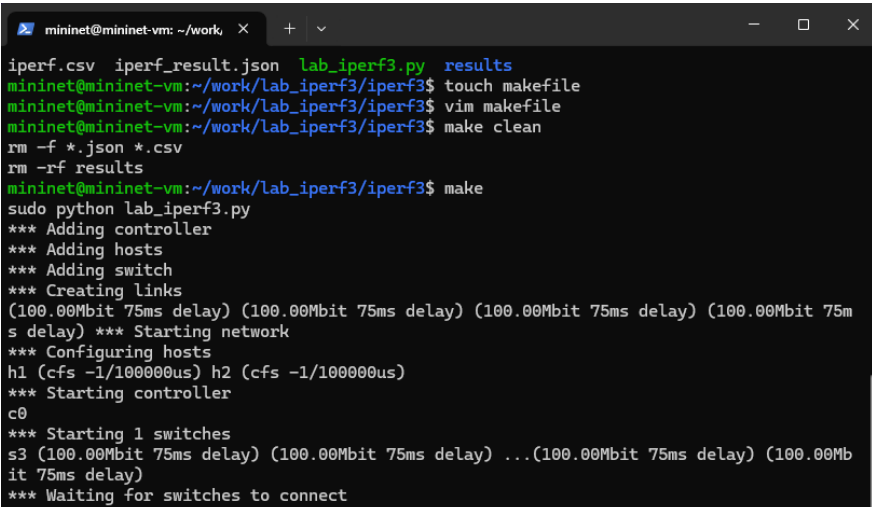
```
/home/mininet/work/lab_iperf3/iperf3/makefile
all: iperf_result.json plot

iperf_result.json:
    sudo python lab_iperf3.py

plot: iperf_result.json
    plot_iperf.sh iperf_result.json
```

# Проверка Makefile

Проверяем корректность работы Makefile и получение ожидаемых результатов.



```
mininet@mininet-vm: ~/work x + v
iperf.csv iperf_result.json lab_iperf3.py results
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ touch makefile
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ vim 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
s 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) (100.00Mb
it 75ms delay)
*** Waiting for switches to connect
```

# Выводы

В ходе работы:

- Ознакомился с iPerf3 и Mininet
- Провел воспроизводимый эксперимент по измерению пропускной способности
- Настроил ресурсы хостов и параметры каналов
- Построил графики и проверил корректность работы Makefile