

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра теории вероятностей и кибербезопасности**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

*дисциплина: Моделирование сетей передачи данных*

Студент: Быстров Глеб

Группа: НПИбд-01-20

**МОСКВА**

2023 г.

## **Цель работы**

В данной лабораторной работе мне будет необходимо познакомиться с инструментом для измерения пропускной способности сети в режиме реального времени — iPerf3, а также получить навыки проведения интерактивного эксперимента по измерению пропускной способности моделируемой сети в среде Mininet.

## Описание процесса выполнения работы

### 2.3.1. Установка необходимого программного обеспечения

1. Запустите виртуальную среду с mininet. Из основной ОС подключитесь к виртуальной машине:

```
ssh -Y mininet@192.168.x.y
```

После подключения к виртуальной машине mininet посмотрите IP-адреса машины:

```
ifconfig
```

Для доступа к сети Интернет должен быть активен адрес NAT: 10.0.0.x.

Если активен только внутренний адрес машины вида 192.168.x.y, то активируйте второй интерфейс, набрав в командной строке:

```
sudo dhclient eth1
```

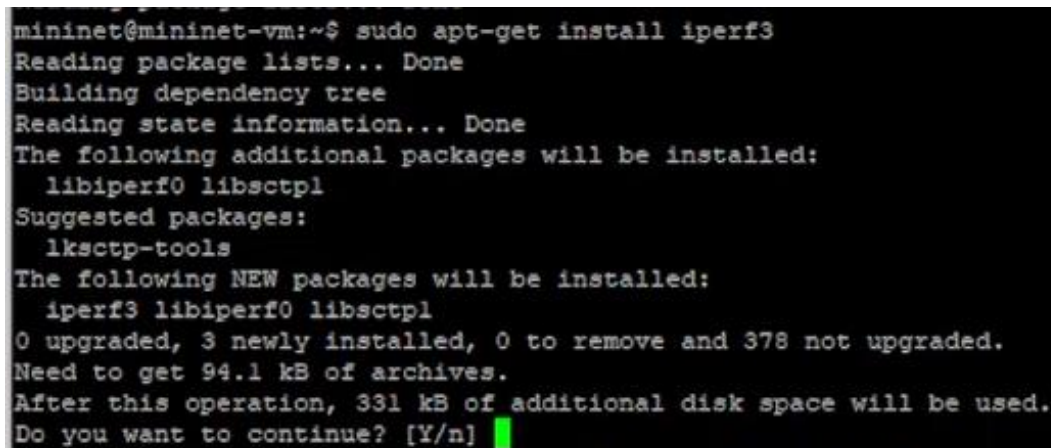
```
ifconfig
```

Обновите репозитории программного обеспечения на виртуальной машине:

```
sudo apt-get update
```

Установите iperf3:

```
sudo apt-get install iperf3 (рис. 2.1).
```



```
mininet@mininet-vm:~$ sudo apt-get install iperf3
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libiperf0 libscpt1
Suggested packages:
  lksctp-tools
The following NEW packages will be installed:
  iperf3 libiperf0 libscpt1
0 upgraded, 3 newly installed, 0 to remove and 378 not upgraded.
Need to get 94.1 kB of archives.
After this operation, 331 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Рис. 2.1. Установка iperf3

2. Установите необходимое дополнительное программное обеспечение на виртуальную машину:

```
sudo apt-get install git jq gnuplot-nox evince
```

Разверните iperf3\_plotter. Для этого:

перейдите во временный каталог и скачайте репозиторий:

```
cd /tmp
```

```
git clone https://github.com/ekfoury/iperf3_plotter.git
```

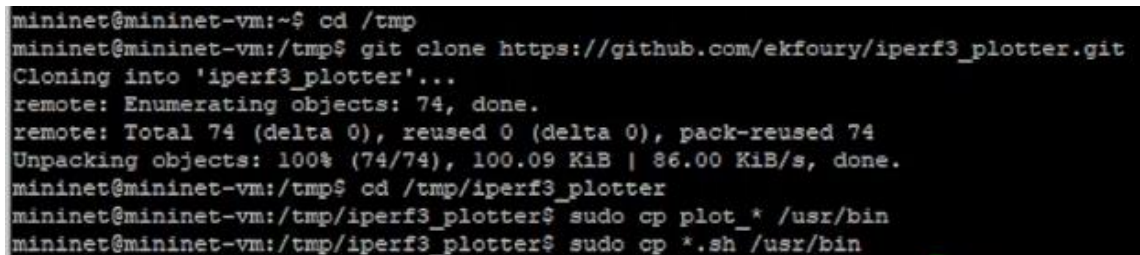
установите iperf3\_plotter:

```
cd /tmp/iperf3_plotter
```

```
sudo cp plot_* /usr/bin
```

```
sudo cp *.sh /usr/bin
```

Обратите внимание, что скрипты не работают с путями, имеющими в названии пробелы и кириллицу (рис. 2.2).



```
mininet@mininet-vm:~$ cd /tmp
mininet@mininet-vm:/tmp$ git clone https://github.com/ekfoury/iperf3_plotter.git
Cloning into 'iperf3_plotter'...
remote: Enumerating objects: 74, done.
remote: Total 74 (delta 0), reused 0 (delta 0), pack-reused 74
Unpacking objects: 100% (74/74), 100.09 KiB | 86.00 KiB/s, done.
mininet@mininet-vm:/tmp$ cd /tmp/iperf3_plotter
mininet@mininet-vm:/tmp/iperf3_plotter$ sudo cp plot_* /usr/bin
mininet@mininet-vm:/tmp/iperf3_plotter$ sudo cp *.sh /usr/bin
```

Рис. 2.2. Установка и развертывание iperf3\_plotter

### 2.3.2. Интерактивные эксперименты

3. Задайте простейшую топологию, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8:

```
sudo mn --topo=single,2 -x
```

После введения этой команды запустятся терминалы двух хостов, коммутатора и контроллера. Терминалы коммутатора и контроллера можно закрыть (рис. 2.3).

```

mininet@mininet-vm:/tmp/iperf3_plotter$ sudo mn --topo=single,2 -x
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Running terms on localhost:10.0
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>

```

Рис. 2.3. Простейшая топология

4. В терминале виртуальной машины посмотрите параметры запущенной в интерактивном режиме топологии:

```
mininet> net
```

```
mininet> links
```

```
mininet> dump (рис. 2 .4).
```

```

mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0
c0
mininet> links
h1-eth0<->s1-eth1 (OK OK)
h2-eth0<->s1-eth2 (OK OK)
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=4312>
<Host h2: h2-eth0:10.0.0.2 pid=4316>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=4321>
<Controller c0: 127.0.0.1:6653 pid=4305>

```

Рис. 2.4. Параметры запущенной топологии

5. Проведите простейший интерактивный эксперимент по измерению пропускной способности с помощью iPerf3 (рис. 2.5).

В терминале h2 запустите сервер iPerf3:

```
iperf3 -s
```

После запуска этой команды хост h2 перейдёт в состояние прослушивания

5201-го порта в ожидании входящих подключений.

В терминале хоста h1 запустите клиент iPerf3:

```
iPerf3 -c 10.0.0.2
```

Здесь параметр -c указывает, что хост h1 настроен как клиент, а параметр 10.0.0.2 является IP-адресом сервера iPerf3 (хост h2).

Дождитесь окончания теста. По умолчанию время тестирования установлено в 10 секунд. Для прерывания работы клиента iPerf3 достаточно на хосте h1 нажать Ctrl + c , при этом сервер iPerf3 на хосте h1 продолжит прослушивать порт 5201. Для остановки как сервера, так и клиента iPerf3 необходимо в терминале хоста h2 нажать Ctrl + c .

Проанализируйте полученный в результате выполнения теста сводный отчёт, отобразившийся как на клиенте, так и на сервере iPerf3, содержащий следующие данные:

ID: идентификационный номер соединения.

интервал (Interval): временной интервал для периодических отчетов о пропускной способности (по умолчанию временной интервал равен 1 секунде);

передача (Transfer): сколько данных было передано за каждый интервал времени;

пропускная способность (Bitrate): измеренная пропускная способность в каждом временном интервале;

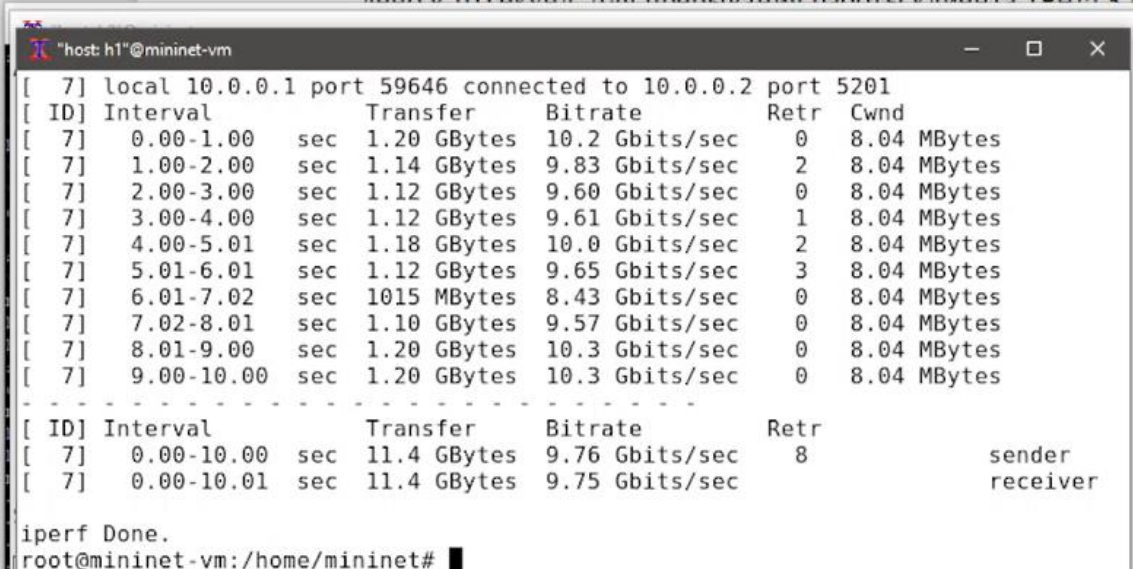
Retr: количество повторно переданных TCP-сегментов за каждый временной интервал (это поле увеличивается, когда TCP-сегменты теряются

в сети из-за перегрузки или повреждения);

Cwnd: указывает размер окна перегрузки в каждом временном интервале (TCP использует эту переменную для ограничения объёма данных, которые TCP-клиент может отправить до получения

подтверждения отправленных данных).

Суммарные данные на сервере аналогичны данным на стороне клиента iPerf3 и должны интерпретироваться таким же образом.



ID	Interval	Transfer	Bitrate	Retr	Cwnd
[ 7]	0.00-1.00 sec	1.20 GBytes	10.2 Gbits/sec	0	8.04 MBytes
[ 7]	1.00-2.00 sec	1.14 GBytes	9.83 Gbits/sec	2	8.04 MBytes
[ 7]	2.00-3.00 sec	1.12 GBytes	9.60 Gbits/sec	0	8.04 MBytes
[ 7]	3.00-4.00 sec	1.12 GBytes	9.61 Gbits/sec	1	8.04 MBytes
[ 7]	4.00-5.01 sec	1.18 GBytes	10.0 Gbits/sec	2	8.04 MBytes
[ 7]	5.01-6.01 sec	1.12 GBytes	9.65 Gbits/sec	3	8.04 MBytes
[ 7]	6.01-7.02 sec	1015 MBytes	8.43 Gbits/sec	0	8.04 MBytes
[ 7]	7.02-8.01 sec	1.10 GBytes	9.57 Gbits/sec	0	8.04 MBytes
[ 7]	8.01-9.00 sec	1.20 GBytes	10.3 Gbits/sec	0	8.04 MBytes
[ 7]	9.00-10.00 sec	1.20 GBytes	10.3 Gbits/sec	0	8.04 MBytes
-----					
ID	Interval	Transfer	Bitrate	Retr	
[ 7]	0.00-10.00 sec	11.4 GBytes	9.76 Gbits/sec	8	sender
[ 7]	0.00-10.01 sec	11.4 GBytes	9.75 Gbits/sec		receiver

iperf Done.  
root@mininet-vm:/home/mininet#

Рис. 2.5. Измерение пропускной способности с помощью iPerf3

6. Проведите аналогичный эксперимент в интерфейсе mininet.

Запустите сервер iPerf3 на хосте h2:

```
mininet> h2 iperf3 -s &
```

Запустите клиент iPerf3 на хосте h1:

```
mininet> h1 iperf3 -c h2
```

Остановите серверный процесс:

```
mininet> h2 killall iperf3
```

Сравните результат с отчётом предыдущего эксперимента (рис. 2.6).

```

mininet@mininet-vm: ~
mininet> h1 iperf3 -c h2
Connecting to host 10.0.0.2, port 5201
[ 5] local 10.0.0.1 port 59650 connected to 10.0.0.2 port 5201
[ ID] Interval          Transfer      Bitrate      Retr  Cwnd
[ 5]  0.00-1.01      sec  1004 MBytes  8.36 Gbits/sec  0  8.36 MBytes
[ 5]  1.01-2.01      sec  1.02 GBytes  8.79 Gbits/sec  0  8.36 MBytes
[ 5]  2.01-3.00      sec  1.04 GBytes  8.98 Gbits/sec  1  8.36 MBytes
[ 5]  3.00-4.01      sec  1.03 GBytes  8.79 Gbits/sec  0  8.36 MBytes
[ 5]  4.01-5.00      sec  1.19 GBytes  10.3 Gbits/sec  0  8.36 MBytes
[ 5]  5.00-6.00      sec  1.15 GBytes  9.87 Gbits/sec  1  8.36 MBytes
[ 5]  6.00-7.00      sec  1.16 GBytes  9.98 Gbits/sec  0  8.36 MBytes
[ 5]  7.00-8.01      sec  1.10 GBytes  9.39 Gbits/sec  0  8.36 MBytes
[ 5]  8.01-9.01      sec   954 MBytes  7.99 Gbits/sec  1  8.36 MBytes
[ 5]  9.01-10.00     sec   959 MBytes  8.09 Gbits/sec  0  8.36 MBytes
-----
[ ID] Interval          Transfer      Bitrate      Retr
[ 5]  0.00-10.00     sec  10.5 GBytes  9.05 Gbits/sec  3
[ 5]  0.00-10.01     sec  10.5 GBytes  9.03 Gbits/sec
sender
receiver
iperf Done.

```

Рис. 2.6. Эксперимент в интерфейсе mininet

- Для указания iPerf3 периода времени для передачи можно использовать ключ `-t` (или `--time`) — время в секундах для передачи (по умолчанию 10 секунд):

В терминале h2 запустите сервер iPerf3:

```
iperf3 -s
```

В терминале h1 запустите клиент iPerf3 с параметром `-t`, за которым следует количество секунд:

```
iperf3 -c 10.0.0.2 -t 5
```

Для остановки сервера нажмите `Ctrl + c` в терминале хоста h2 (рис. 2.7).

```

root@mininet-vm:~# iperf3 -c 10.0.0.2 -t 5
Connecting to host 10.0.0.2, port 5201
[ 7] local 10.0.0.1 port 59684 connected to 10.0.0.2 port 5201
[ ID] Interval          Transfer      Bitrate      Retr  Cwnd
[ 7]  0.00-1.00      sec  1.17 GBytes  10.0 Gbits/sec  0  8.39 MBytes
[ 7]  1.00-2.00      sec  1.17 GBytes  10.0 Gbits/sec  0  8.39 MBytes
[ 7]  2.00-3.01      sec  1.24 GBytes  10.6 Gbits/sec  0  8.39 MBytes
[ 7]  3.01-4.00      sec  1.20 GBytes  10.3 Gbits/sec  2  8.39 MBytes
[ 7]  4.00-5.00      sec  1.18 GBytes  10.1 Gbits/sec  1  5.91 MBytes
-----
[ ID] Interval          Transfer      Bitrate      Retr
[ 7]  0.00-5.00      sec  5.95 GBytes  10.2 Gbits/sec  3
[ 7]  0.00-5.01      sec  5.95 GBytes  10.2 Gbits/sec
sender
receiver
iperf Done.
root@mininet-vm:~# █

```

Рис. 2.7. Эксперимент с указанием времени



8. Настройте клиент iPerf3 для выполнения теста пропускной способности с 2-секундным интервалом времени отсчёта как на клиенте, так и на сервере.

Используйте опцию -i для установки интервала между отсчётами, измеряемого в секундах:

В терминале h2 запустите сервер iPerf3:

```
iperf3 -s -i 2
```

В терминале h1 запустите клиент iPerf3:

```
iperf3 -c 10.0.0.2 -i 2
```

Остановите сервер iPerf3, нажав Ctrl+c в терминале хоста h2.

Сравните результат с отчётами из предыдущих экспериментов. (рис. 2.8).

```
root@mininet-vm:~# iperf3 -c 10.0.0.2 -i 2
Connecting to host 10.0.0.2, port 5201
[ 7] local 10.0.0.1 port 59688 connected to 10.0.0.2 port 5201
[ ID] Interval          Transfer      Bitrate      Retr  Cwnd
[ 7]  0.00-2.00    sec    2.06 GBytes    8.83 Gbits/sec    1   8.30 MBytes
[ 7]  2.00-4.00    sec    2.13 GBytes    9.12 Gbits/sec    0   8.30 MBytes
[ 7]  4.00-6.00    sec    2.06 GBytes    8.83 Gbits/sec    2   8.30 MBytes
[ 7]  6.00-8.00    sec    2.31 GBytes    9.94 Gbits/sec    1   8.30 MBytes
[ 7]  8.00-10.00   sec    2.13 GBytes    9.16 Gbits/sec    2   8.30 MBytes
-----
[ ID] Interval          Transfer      Bitrate      Retr
[ 7]  0.00-10.00   sec    10.7 GBytes    9.17 Gbits/sec    6
[ 7]  0.00-10.00   sec    10.7 GBytes    9.17 Gbits/sec
iperf Done.
```

Рис. 2.8. Эксперимент с интервалами

9. Задайте на клиенте iPerf3 отправку определённого объёма данных.

Используйте опцию -n для установки количества байт для передачи:

В терминале h2 запустите сервер iPerf3:

```
iperf3 -s
```

В терминале h1 запустите клиент iPerf3, задав объём данных 16 Гбайт:

```
iperf3 -c 10.0.0.2 -n 16G
```

Обратите внимание, что по умолчанию iPerf3 выполняет измерение пропускной способности в течение 10 секунд, но при задании количества данных для передачи клиент iPerf3 будет продолжать

отправлять пакеты до тех пор, пока не будет отправлен весь объем данных, указанный пользователем.

Остановите сервер iPerf3, нажав Ctrl+c в терминале хоста h2 (рис. 2.9).

```
root@mininet-vm:~# iperf3 -c 10.0.0.2 -n 16G
Connecting to host 10.0.0.2, port 5201
[ 7] local 10.0.0.1 port 59692 connected to 10.0.0.2 port 5201
[ ID] Interval            Transfer          Bitrate          Retr  Cwnd
[ 7]  0.00-1.00    sec  1.10 GBytes    9.48 Gbits/sec    1   7.00 MBytes
[ 7]  1.00-2.00    sec  1.13 GBytes    9.70 Gbits/sec    0   8.11 MBytes
[ 7]  2.00-3.00    sec  1.13 GBytes    9.67 Gbits/sec    0   8.11 MBytes
[ 7]  3.00-4.00    sec  1.15 GBytes    9.85 Gbits/sec    0   8.11 MBytes
[ 7]  4.00-5.00    sec  1.17 GBytes   10.0 Gbits/sec    1   8.11 MBytes
[ 7]  5.00-6.00    sec  1021 MBytes    8.55 Gbits/sec    1   8.11 MBytes
[ 7]  6.00-7.00    sec  1.09 GBytes    9.38 Gbits/sec    1   8.11 MBytes
[ 7]  7.00-8.00    sec  1.09 GBytes    9.34 Gbits/sec    0   8.11 MBytes
[ 7]  8.00-9.00    sec  1.09 GBytes    9.38 Gbits/sec    0   8.11 MBytes
[ 7]  9.00-10.00   sec  1.14 GBytes    9.77 Gbits/sec    1   8.11 MBytes
[ 7] 10.00-11.00   sec  1.17 GBytes   10.1 Gbits/sec    1   8.11 MBytes
[ 7] 11.00-12.00   sec  1.17 GBytes   10.0 Gbits/sec    0   8.11 MBytes
```

Рис. 2.9. Отправка определённого объёма данных

10. Измените в тесте измерения пропускной способности iPerf3 протокол передачи данных с TCP (установлен по умолчанию) на UDP. iPerf3 автоматически определяет протокол транспортного уровня на стороне сервера. Для изменения протокола используйте опцию -u на стороне клиента iPerf3:

В терминале h2 запустите сервер iPerf3:

```
iperf3 -s
```

В терминале h1 запустите клиент iPerf3, задав протокол UDP:

```
iperf3 -c 10.0.0.2 -u
```

После завершения теста отобразятся следующие сводные данные:

ID, интервал, передача, битрейт: то же, что и у TCP.

Jitter: разница в задержке пакетов.

Lost/Total: указывает количество потерянных дейтаграмм по сравнению с общим количеством отправленных на сервер (и процентное соотношение).

Остановите сервер iPerf3, нажав Ctrl+c в терминале хоста h2. (рис. 2.10).

```

root@mininet-vm:~# iperf3 -c 10.0.0.2 -u
Connecting to host 10.0.0.2, port 5201
[ 7] local 10.0.0.1 port 56721 connected to 10.0.0.2 port 5201
[ ID] Interval            Transfer          Bitrate          Total Datagrams
[ 7] 0.00-1.00 sec      129 KBytes       1.05 Mbits/sec    91
[ 7] 1.00-2.00 sec      127 KBytes       1.04 Mbits/sec    90
[ 7] 2.00-3.00 sec      129 KBytes       1.05 Mbits/sec    91
[ 7] 3.00-4.00 sec      127 KBytes       1.04 Mbits/sec    90
[ 7] 4.00-5.00 sec      129 KBytes       1.05 Mbits/sec    91
[ 7] 5.00-6.00 sec      127 KBytes       1.04 Mbits/sec    90
[ 7] 6.00-7.00 sec      129 KBytes       1.05 Mbits/sec    91

```

Рис. 2.10. Изменение протокола передачи данных

11. В тесте измерения пропускной способности iPerf3 измените номер порта для отправки/получения пакетов или датаграмм через указанный порт.

Используйте для этого опцию -p:

В терминале h2 запустите сервер iPerf3, используя параметр -p, чтобы указать порт прослушивания:

```
iperf3 -s -p 3250
```

В терминале h1 запустите клиент iPerf3, указав порт:

```
iperf3 -c 10.0.0.2 -p 3250
```

Остановите сервер iPerf3, нажав Ctrl+c в терминале хоста h2 (рис. 2.11).

```

root@mininet-vm:~# iperf3 -s -p 3250
Connecting to host 10.0.0.2, port 3250
[ 7] local 10.0.0.1 port 50606 connected to 10.0.0.2 port 3250
[ ID] Interval            Transfer          Bitrate          Retr  Cwnd
[ 7] 0.00-1.00 sec      1.19 GBytes       10.2 Gbits/sec    0     8.37 MBytes
[ 7] 1.00-2.00 sec      1.17 GBytes       10.0 Gbits/sec    1     8.37 MBytes
[ 7] 2.00-3.00 sec      1.12 GBytes       9.56 Gbits/sec    1     8.37 MBytes
[ 7] 3.00-4.00 sec      1.02 GBytes       8.79 Gbits/sec    1     8.37 MBytes
[ 7] 4.00-5.00 sec      1.07 GBytes       9.20 Gbits/sec    1     8.37 MBytes
[ 7] 5.00-6.00 sec      1.04 GBytes       8.95 Gbits/sec    0     8.37 MBytes

```

Рис. 2.11. Изменение номера порта

12. По умолчанию после запуска сервер iPerf3 постоянно прослушивает входящие соединения. В тесте измерения пропускной способности iPerf3 задайте для сервера параметр обработки данных только от одного клиента с остановкой сервера по завершении теста. Для этого используйте опцию -l на сервере iPerf3:

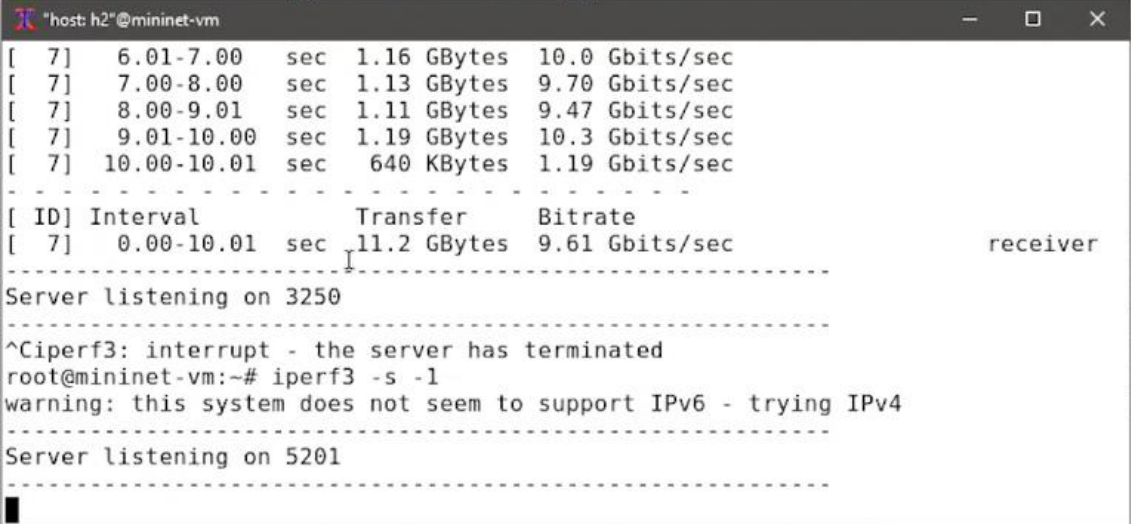
В терминале h2 запустите сервер iPerf3, используя параметр -1, чтобы принять только одного клиента:

```
iperf3 -s -1
```

В терминале h1 запустите клиент iPerf3:

```
iperf3 -c 10.0.0.2
```

Обратите внимание, что после завершения этого теста сервер iPerf3 немедленно останавливается (рис. 2.12).



```
host: h2@mininet-vm
[  7]  6.01-7.00   sec   1.16 GBytes  10.0 Gbits/sec
[  7]  7.00-8.00   sec   1.13 GBytes  9.70 Gbits/sec
[  7]  8.00-9.01   sec   1.11 GBytes  9.47 Gbits/sec
[  7]  9.01-10.00  sec   1.19 GBytes  10.3 Gbits/sec
[  7] 10.00-10.01  sec    640 KBytes  1.19 Gbits/sec
-----
[ ID] Interval        Transfer   Bitrate          receiver
[  7]  0.00-10.01  sec  11.2 GBytes  9.61 Gbits/sec
-----
Server listening on 3250
-----
^Ciperf3: interrupt - the server has terminated
root@mininet-vm:~# iperf3 -s -1
warning: this system does not seem to support IPv6 - trying IPv4
-----
Server listening on 5201
-----
```

Рис. 2.12. Принятие только одного клиента

13. Экспортируйте результаты теста измерения пропускной способности iPerf3 в файл JSON:

В виртуальной машине mininet создайте каталог для работы над проектом:

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

В терминале h2 запустите сервер iPerf3:

```
iperf3 -s
```

В терминале h1 запустите клиент iPerf3, указав параметр -J для отображения вывода результатов в формате JSON:

```
iperf3 -c 10.0.0.2 -J
```

В данном случае параметр -J выведет текст JSON на экран через стандартный вывод (stdout) после завершения теста.

Экспортируйте вывод результатов теста в файл, перенаправив стандартный вывод в файл:

```
iperf3 -c 10.0.0.2 -J > /home/mininet/work/lab_iperf3/iperf_results.json
```

Убедитесь, что файл iperf\_results.json создан в указанном каталоге.

Для этого в терминале хоста h1 введите следующие команды:

```
cd /home/mininet/work/lab_iperf3
```

```
ls -l
```

Команда cat может использоваться для отображения содержимого файла.

Остановите сервер iPerf3, нажав Ctrl+c в терминале хоста h2.

Завершите работу mininet в интерактивном режиме, введя в интерфейсе mininet:

```
mininet> exit (рис. 2.13).
```



```
"host: h1"@mininet-vm
{
  "bytes": 12529995456,
  "bits_per_second": 10020393031.465885,
  "sender": true
},
"cpu_utilization_percent": {
  "host_total": 49.69162804803144,
  "host_user": 0.87219121964966451,
  "host_system": 48.819436828381775,
  "remote_total": 17.548528507318316,
  "remote_user": 1.3731889684129586,
  "remote_system": 16.175328702759053
},
"sender_tcp_congestion": "cubic",
"receiver_tcp_congestion": "cubic"
}
root@mininet-vm:~# iperf3 -c 10.0.0.2 -J > /home/mininet/work/lab_iperf3/iperf_results.json
root@mininet-vm:~# cd /home/mininet/work/lab_iperf3
root@mininet-vm:/home/mininet/work/lab_iperf3# ls -l
total 8
-rw-r--r-- 1 root root 7806 Nov 24 10:01 iperf_results.json
root@mininet-vm:/home/mininet/work/lab_iperf3# ls
iperf_results.json
root@mininet-vm:/home/mininet/work/lab_iperf3# █
```

Рис. 2.13. Экспортирование результатов теста

#### 14. Визуализируйте результаты эксперимента:

В виртуальной машине mininet исправьте права запуска X-соединения.

Скопируйте значение куки (MIT magic cookie) своего пользователя mininet в файл для пользователя root:

```
mininet@mininet-vm:~$ xauth list $DISPLAY
```

```
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1
↪ 295acad8e35d17636924c5ab80e8462d
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth add mininet-vm/unix:10
↪ MIT-MAGIC-COOKIE-1 295acad8e35d17636924c5ab80e8462d
root@mininet-vm:~# logout
```

После выполнения этих действий графические приложения должны запускаться под пользователем mininet.

В виртуальной машине mininet перейдите в каталог для работы над проектом, проверьте и при необходимости скорректируйте права доступа

к файлу JSON:

```
cd ~/work/lab_iperf3
ls -l
sudo chown -R mininet:mininet ~/work
ls -l
```

Сгенерируйте выходные данные для файла JSON iPerf3, выполнив следующую команду (обратите внимание, что скрипт не работает с путями,

имеющими в названии файла пробелы):

```
1 plot_iperf.sh iperf3_results.json
```

Сценарий построения должен создать файл CSV (1.dat), который может использоваться другими приложениями. В подкаталоге results каталога, в котором был выполнен скрипт, сценарий должен создать графики для следующих полей файла JSON:

- окно перегрузки (cwnd.pdf);
  - повторная передача (retransmits.pdf);
  - время приема-передачи (RTT.pdf);
  - отклонение времени приема-передачи (RTT\_Var.pdf);
  - пропускная способность (throughput.pdf);
  - максимальная единица передачи (MTU.pdf);
  - количество переданных байтов (bytes.pdf).
- Убедитесь, что файлы с данными и графиками сформировались:

```
cd ~/work/lab_iperf3
ls -l
cd ~/work/lab_iperf3/results
ls -l (рис. 2.14).
```

```

mininet@mininet-vm: ~/work/lab_iperf3/results
-rw-r--r-- 1 mininet mininet 7806 Nov 24 10:01 iperf_results.json
mininet@mininet-vm:~/work/lab_iperf3$ plot_iperf.sh iperf3_results.json
Error: iperf3_results.json is not a file. Quitting...
mininet@mininet-vm:~/work/lab_iperf3$ plot_iperf.sh iperf3_result.json
Error: iperf3_result.json is not a file. Quitting...
mininet@mininet-vm:~/work/lab_iperf3$ plot_iperf.sh iperf_results.json
mininet@mininet-vm:~/work/lab_iperf3$ ls -l
total 16
-rw-rw-r-- 1 mininet mininet 969 Nov 24 10:09 iperf.csv
-rw-r--r-- 1 mininet mininet 7806 Nov 24 10:01 iperf_results.json
drwxrwxr-x 2 mininet mininet 4096 Nov 24 10:09 results
mininet@mininet-vm:~/work/lab_iperf3$ cd /results
-bash: cd: /results: No such file or directory
mininet@mininet-vm:~/work/lab_iperf3$ cd results
mininet@mininet-vm:~/work/lab_iperf3/results$ ls -l
total 88
-rw-rw-r-- 1 mininet mininet 508 Nov 24 10:09 1.dat
-rw-rw-r-- 1 mininet mininet 9800 Nov 24 10:09 bytes.pdf
-rw-rw-r-- 1 mininet mininet 9618 Nov 24 10:09 cwnd.pdf
-rw-rw-r-- 1 mininet mininet 9036 Nov 24 10:09 MTU.pdf
-rw-rw-r-- 1 mininet mininet 8998 Nov 24 10:09 retransmits.pdf
-rw-rw-r-- 1 mininet mininet 9056 Nov 24 10:09 RTT.pdf
-rw-rw-r-- 1 mininet mininet 9133 Nov 24 10:09 RTT_Var.pdf
-rw-rw-r-- 1 mininet mininet 9646 Nov 24 10:09 throughput.pdf
mininet@mininet-vm:~/work/lab_iperf3/results$

```

Рис. 2.14. Просмотр файлов с графиками

## **Вывод**

В данной лабораторной работе мне успешно удалось познакомиться с инструментом для измерения пропускной способности сети в режиме реального времени — iPerf3, а также получить навыки проведения интерактивного эксперимента по измерению пропускной способности моделируемой сети в среде Mininet.