

Лабораторная работа № 2. Измерение и тестирование пропускной способности сети. Интерактивный эксперимент

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 интерактивного эксперимента по измерению пропускной способности моделируемой сети в среде Mininet.

2.2. Предварительные сведения

В контексте сеанса связи между двумя конечными устройствами на сетевом пути под *пропускной способностью (throughput)* понимается скорость в битах в секунду, с которой процесс-отправитель может доставлять данные процессу-получателю. В тоже время под *полосой пропускания (Bandwidth)* понимается физическое свойство среды передачи данных, зависящее, например, от конструкции и длины провода или волокна. Иногда термины «пропускная способность» (throughput) и «полоса пропускания» (bandwidth) используются взаимозаменяемо.

iPerf3 (сайт: <https://iperf.fr/>; лицензия: three-clause BSD license; репозиторий: <https://github.com/esnet/iperf>) представляет собой кроссплатформенное клиент-серверное приложение с открытым исходным кодом, которое можно использовать для измерения пропускной способности между двумя конечными устройствами.

iPerf3 может работать с транспортными протоколами TCP, UDP и SCTP:

- TCP и SCTP:
 - измеряет пропускную способность;
 - позволяет задать размер MSS/MTU;
 - отслеживает размер окна перегрузки TCP (CWnd).
- UDP:
 - измеряет пропускную способность;
 - измеряет потери пакетов;
 - измеряет колебания задержки (jitter);
 - поддерживает групповую рассылку пакетов (multicast).

Пользователь взаимодействует с iPerf3 с помощью команды `iperf3`. Основной синтаксис `iperf3`, используемый как на клиенте, так и на сервере, выглядит следующим образом:

```
iperf3 [-s|-c] [options]
```

Здесь `-s` — запуск сервера; `-c` — запуск клиента. Описание опций можно посмотреть, введя команду `iperf3 -h`.

iPerf3 позволяет экспортировать результаты теста в файл с облегчённым форматом обмена данными JSON (JavaScript Object Notation, нотация объектов JavaScript), что позволяет другим приложениям легко анализировать файл и интерпретировать результаты.

Для визуализации результатов измерения iPerf3 можно использовать пакет `iperf3_plotter`. Репозиторий https://github.com/ekfoury/iperf3_plotter содержит следующие скрипты:

- `preprocessor.sh`: преобразует файл JSON iPerf3 в файл значений, разделённых запятыми (CSV); использует AWK для форматирования полей файла;
- `plot_iperf.sh`: принимает JSON-файл iPerf3, вызывает препроцессор и `gnuplot` для создания выходных PDF-файлов.

2.3. Задание

1. Установить на виртуальную машину `mininet` iPerf3 и дополнительное программное обеспечение для визуализации и обработки данных.
2. Провести ряд интерактивных экспериментов по измерению пропускной способности с помощью iPerf3 с построением графиков.

2.4. Последовательность выполнения работы

2.4.1. Установка необходимого программного обеспечения

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

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

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

```
1 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:

```
1 sudo dhclient eth1
2 ifconfig
```

4. Обновите репозитории программного обеспечения на виртуальной машине:

```
1 sudo apt-get update
```

5. Установите `iperf3`:

```
1 sudo apt-get install iperf3
```

6. Установите необходимое дополнительное программное обеспечение на виртуальную машину:

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

7. Разверните `iperf3_plotter`. Для этого:

- перейдите во временный каталог и скачайте репозиторий:

```
1 cd /tmp
2 git clone https://github.com/ekfoury/iperf3_plotter.git
```

- установите `iperf3_plotter`:

```
1 cd /tmp/iperf3_plotter
2 sudo cp plot_* /usr/bin
3 sudo cp *.sh /usr/bin
```

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

2.4.2. Интерактивные эксперименты

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

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

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

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

```
1 mininet> net
2 mininet> links
3 mininet> dump
```

3. Проведите простейший интерактивный эксперимент по измерению пропускной способности с помощью `iPerf3`:

- В терминале h2 запустите сервер iPerf3:

```
1  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:

```
1  iperf3 -c 10.0.0.2
```

Здесь параметр -с 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);
- Swnd: 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.

4. Проведите аналогичный эксперимент в интерфейсе mininet.

- Запустите сервер iPerf3 на хосте h2:

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

- Запустите клиент iPerf3 на хосте h1:

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

- Остановите серверный процесс:

```
1 mininet> h2 killall iperf3
```

- Сравните результат с отчётом предыдущего эксперимента.

5. 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:

```
1 iperf3 -s
```

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

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

- Для остановки сервера нажмите `Ctrl` + `C` в терминале хоста h2.

6. 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:

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

- В терминале h1 запустите клиент iPerf3:

```
1 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.

7. Задайте на клиенте iPerf3 отправку определённого объёма данных. Используйте опцию `-n` для установки количества байт для передачи:

- В терминале h2 запустите сервер iPerf3:

```
1 iperf3 -s
```

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

```
1 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 объём данных, указанный пользователем.

- Остановите сервер iPerf3, нажав Ctrl+c в терминале хоста h2.
- 8. 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:

```
1 iperf3 -s
```

- В терминале h1 запустите клиент iPerf3, задав протокол UDP:

```
1 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.
- 9. 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:

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

- В терминале h1 запустите клиент iPerf3, указав порт:

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

- Остановите сервер iPerf3, нажав Ctrl+c в терминале хоста h2.
- 10. 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 -1 на сервере iPerf3:
 - В терминале h2 запустите сервер iPerf3, используя параметр -1, чтобы принять только одного клиента:

```
1 iperf3 -s -1
```

- В терминале h1 запустите клиент iPerf3:

```
1  iperf3 -c 10.0.0.2
```

Обратите внимание, что после завершения этого теста сервер iPerf3 немедленно останавливается.

11. Экспортируйте результаты теста измерения пропускной способности iPerf3 в файл JSON:
 - В виртуальной машине mininet создайте каталог для работы над проектом:

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

- В терминале h2 запустите сервер iPerf3:

```
1  iperf3 -s
```

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

```
1  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:

```
1  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:

```
1  cd /home/mininet/work/lab_iperf3  
2  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:

```
1  mininet> exit
```

12. Визуализируйте результаты эксперимента:
 - В виртуальной машине mininet исправьте права запуска X-соединения. Скопируйте значение куки (MIT magic cookie)¹ своего пользователя mininet в файл для пользователя root:

¹Значение для MIT-MAGIC-COOKIE приведено условно.

```
1 mininet@mininet-vm:~$ xauth list $DISPLAY
2 mininet-vm/unix:10 MIT-MAGIC-COOKIE-1
   ↪ 295acad8e35d17636924c5ab80e8462d
3
4 mininet@mininet-vm:~$ sudo -i
5 root@mininet-vm:~# xauth add mininet-vm/unix:10
   ↪ MIT-MAGIC-COOKIE-1 295acad8e35d17636924c5ab80e8462d
6 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:

```
1 cd ~/work/lab_iperf3
2 ls -l
3 sudo chown -R mininet:mininet ~/work
4 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:

```
1 cd ~/work/lab_iperf3
2 ls -l
3 cd ~/work/lab_iperf3/results
4 ls -l
```


2.5. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулировка задания работы.
3. Описание результатов выполнения задания:
 - скриншоты (снимки экрана), фиксирующие выполнение работы;
 - подробное описание настроек служб в соответствии с заданием;
 - результаты проверки корректности настроек служб в соответствии с заданием (подтверждённые скриншотами);
 - листинги (исходный код) программ (если они есть);
 - результаты выполнения программ (текст, графики или снимки экрана в зависимости от задания).
4. Выводы, согласованные с заданием работы.