Лабораторная работа №3

Моделирование сетей передачи данных

Еюбоглу Тимур

Содержание

Список иллюстраций

Список таблиц

## 0.1 Цель лабораторной 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.

# 1 Выполнение лабораторной работы

С помощью API Mininet создадим простейшую топологию сети, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8. Для этого в каталоге /work/lab\_iperf3 для работы над проектом создадим подкаталог lab\_iperf3\_topo и скопируем в него файл с примером скрипта mininet/examples/emptynet.py, описывающего стандартную простую топологию сети mininet (рис. 1):

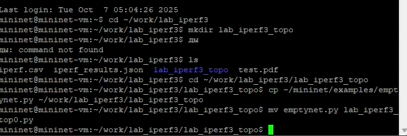


Рис. 1: Создание подкаталога, копирование файла с примером скрипта (описывающего стандартную простую топологию сети mininet)

Изучим содержание скрипта lab\_iperf3\_topo.py (рис. 2):

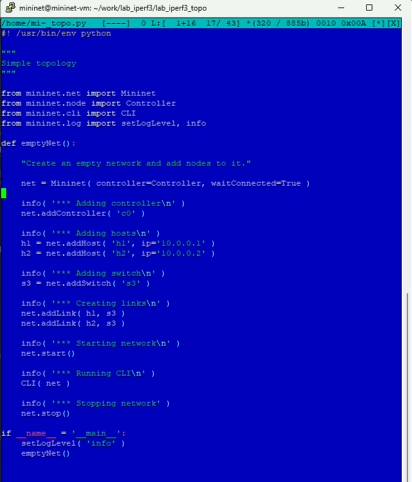


Рис. 2: Открытие файла lab\_iperf3\_topo.py

Запустим скрипт создания топологии lab\_iperf3\_topo.py. После отработки скрипта просмотрим элементы топологии и завершим работу mininet (рис. 3):

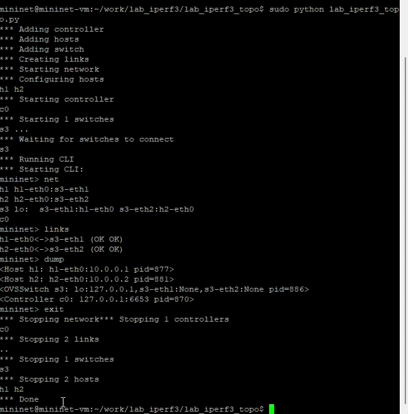


Рис. 3: Запуск скрипта создания топологии и дальнейший просмотр элементов

Следующим шагом внесём в скрипт lab\_iperf3\_topo.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, добавим нужную строку (рис. 4):

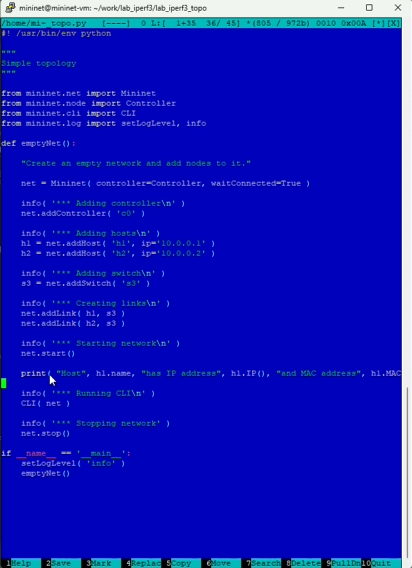


Рис. 4: Внесение изменения в скрипт, позволяющего вывести на экран информацию о хосте h1 (имя, IP-адрес, MAC-адрес)

Проверим корректность отработки изменённого скрипта (рис. 5):

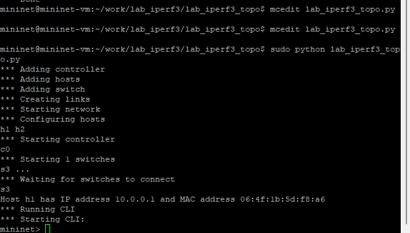


Рис. 5: Проверка корректности отработки скрипта

Затем изменим скрипт lab\_iperf3\_topo.py так, чтобы на экран выводилась информация об имени, IP-адресе и MAC-адресе обоих хостов сети и проверим корректность отработки изменённого скрипта (рис. 6 - рис. 7)

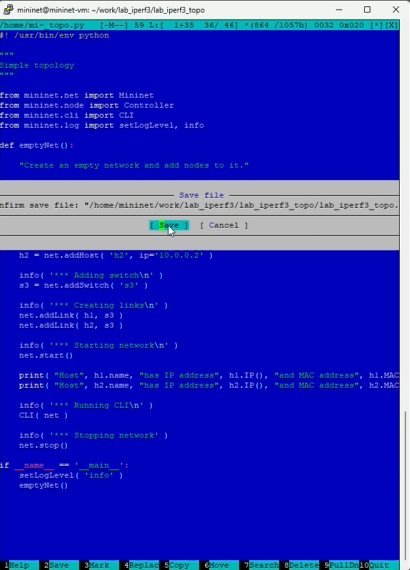


Рис. 6: Внесение изменения в скрипт, позволяющего вывести на экран информацию о двух хостах (имя, IP-адрес, MAC-адрес)

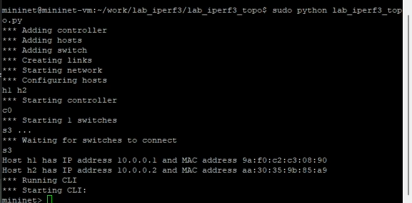


Рис. 7: Проверка корректности отработки скрипта

Mininet предоставляет функции ограничения производительности и изоляции с помощью классов CPULimitedHost и TCLink. Добавим в скрипт настройки параметров производительности. Для начала сделаем копию скрипта lab\_iperf3\_topo.py (рис. 8):



Рис. 8: Создание копии скрипта lab\_iperf3\_topo.py

В начале скрипта lab\_iperf3\_topo2.py добавим записи об импорте классов CPULimitedHost и TCLink. Далее изменим строку описания сети, указав на использование ограничения производительности и изоляции. Следующим шагом изменим функцию задания параметров виртуального хоста h1, указав, что ему будет выделено 50% от общих ресурсов процессора системы. Аналогичным образом для хоста h2 зададим долю выделения ресурсов процессора в 45%. В конце изменим функцию параметров соединения между хостом h1 и коммутатором s3 (рис. 9):

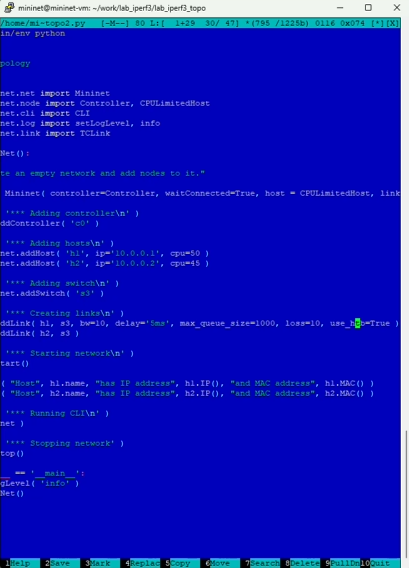


Рис. 9: Изменение скрипта lab\_iperf3\_topo2.py: добавление ипорта классов, изменение строки описания сети, изменение функции задания параметров виртуального хоста h1 и h2, изменение функции параметров соединения между хостом h1 и коммутатором s3

Запустим на отработку скрипт lab\_iperf3\_topo2.py (рис. 10):

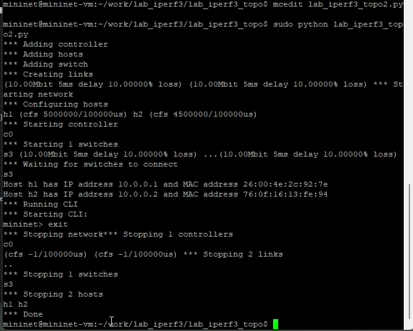


Рис. 10: Запустим на отработку скрипт lab\_iperf3\_topo2.py

Перед завершением лабораторной работы, построим графики по проводимому эксперименту. Для этого сделаем копию скрипта lab\_iperf3\_topo2.py и поместим его в подкаталог iperf (рис. 11):

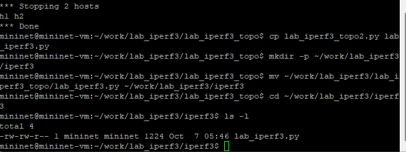


Рис. 11: Создание копии скрипта lab\_iperf3\_topo2.py и его дальнейшее помещение в подкаталог iperf

В начале скрипта lab\_iperf3.py добавим запись об импорте time и изменим код в скрипте так, чтобы (рис. 12): - 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

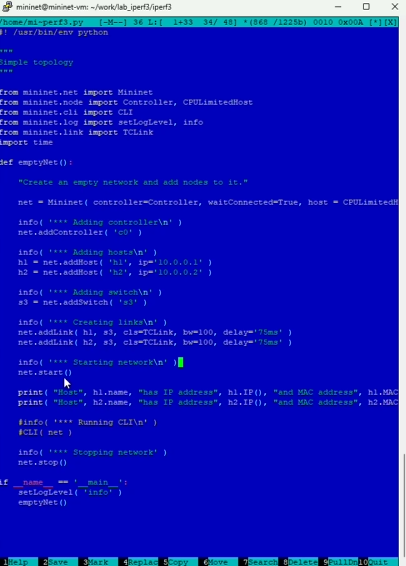


Рис. 12: Добавление в скрипт lab\_iperf3.py записи об импорте time; снятие ограничений по использованию ресурсов процессора; добавление кода, чтобы каналы между хостами и коммутатором были по 100 Мбит/с с задержкой 75 мс, без потерь

После функции старта сети опишим запуск на хосте h2 сервера iPerf3, а на хосте h1 запуск с задержкой в 10 секунд клиента iPerf3 с экспортом результатов в JSON-файл, закомментируем строки, отвечающие за запуск CLI-интерфейса (рис. 13):



Рис. 13: Описание запуска на хосте h2 сервера iPerf3, на хосте h1 запуска с задержкой в 10 секунд клиента iPerf3 с экспортом результатов в JSON-файл. Комментирование строк, отвечающих за запуск CLI-интерфейса

Запустим на отработку скрипт lab\_iperf3.py (рис. 14):

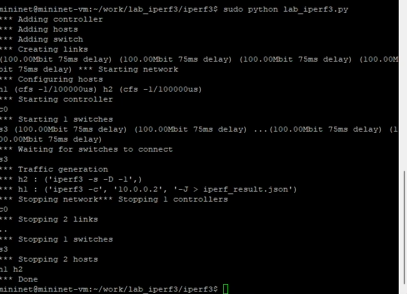


Рис. 14: Запуск скрипта lab\_iperf3.py на отработку

Построим графики из получившегося JSON-файла и создадим Makefile для проведения всего эксперимента (рис. 15):

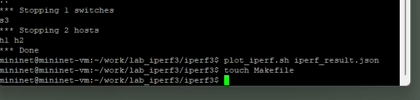


Рис. 15: Построение графиков и создание Makefile для проведения всего эксперимента

В Makefile пропишим запуск скрипта эксперимента, построение графиков и очистку каталога от результатов (рис. 16):

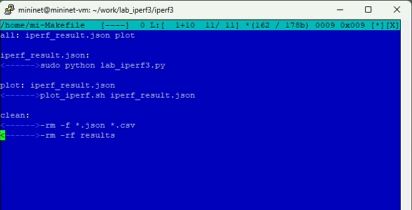


Рис. 16: Добавление скрипта в Makefile

Проверим корректность отработки Makefile (рис. 17):

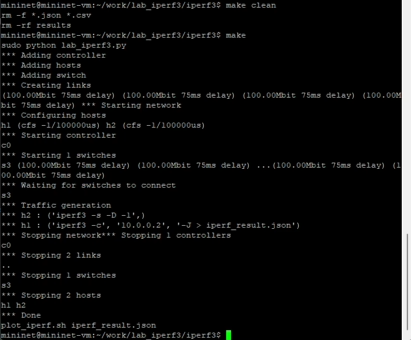


Рис. 17: Проверка корректности отработки Makefile

# 2 Вывод

## 2.1 Вывод

В ходе выполнения лабораторной работы 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

# 3 Список литературы. Библиография

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