Моделирование сетей передачи данных

Отчёт по лабораторной работе №3: Измерение и тестирование пропускной способности сети. Воспроизводимый эксперимент

Кармацкий Никита Сергеевич

Содержание

# 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 создаем 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 создаем подкаталог lab\_iperf3\_topo и скопируем в него файл с примером скрипта mininet/examples/emptynet.py, описывающего стандартную простую топологию сети Mininet. (рис. 1):

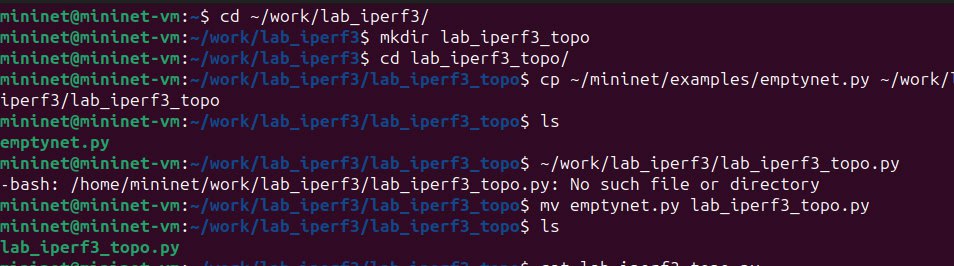


Рис. 1: Создание подкаталога и копирование файлов

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

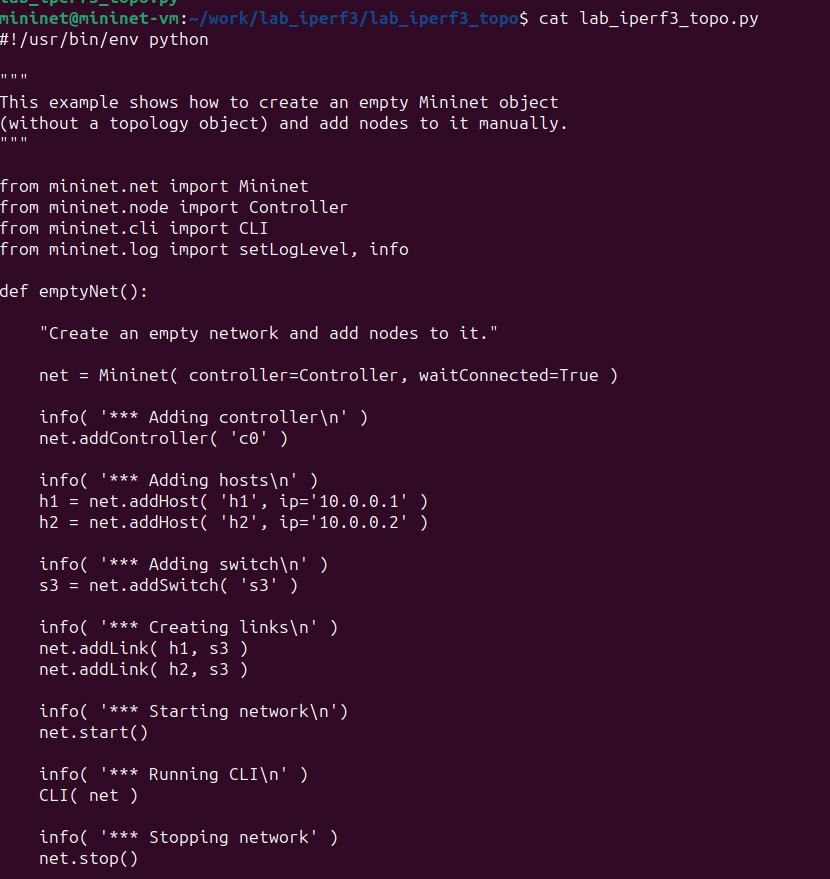


Рис. 2: Скрипт lab\_iperf3\_topo.py

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

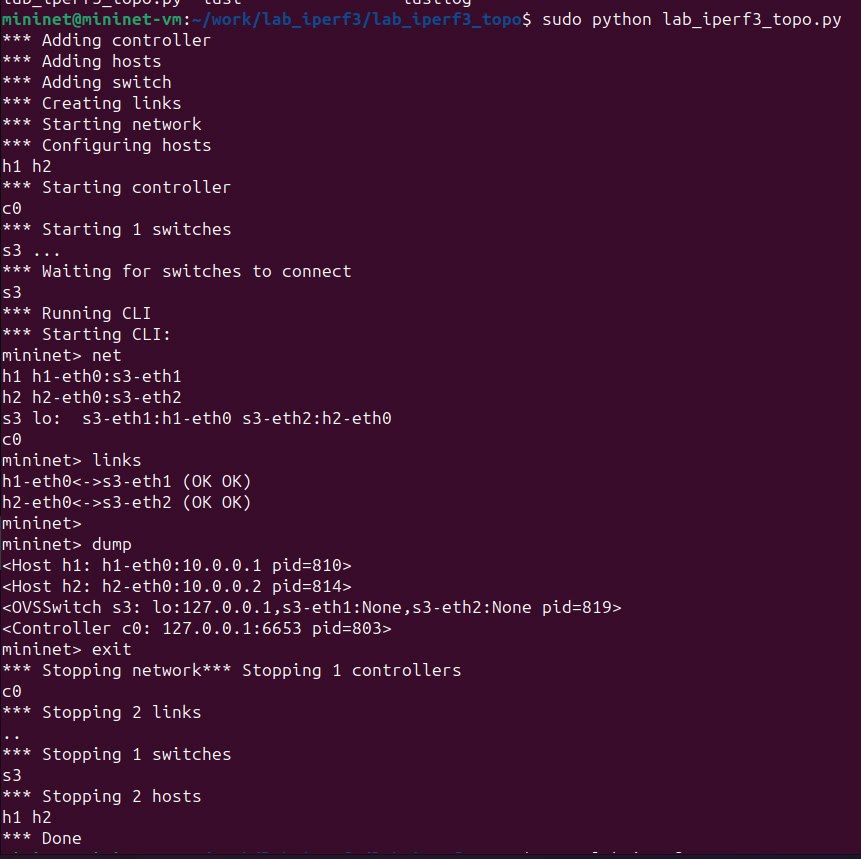


Рис. 3: Запуск скрипта и просмотр элементов топологии

1. 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):

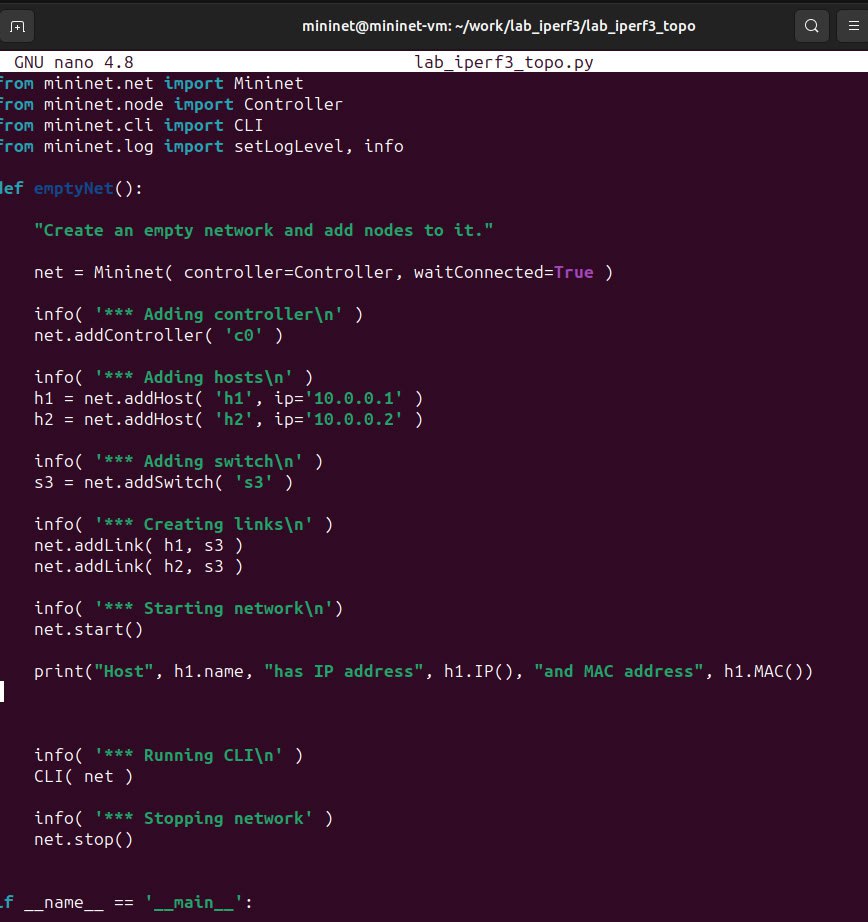


Рис. 4: Редактирование скрипта для просмотра информации по хосту h1

1. Проверим корректность отработки изменённого скрипта (рис. 5):

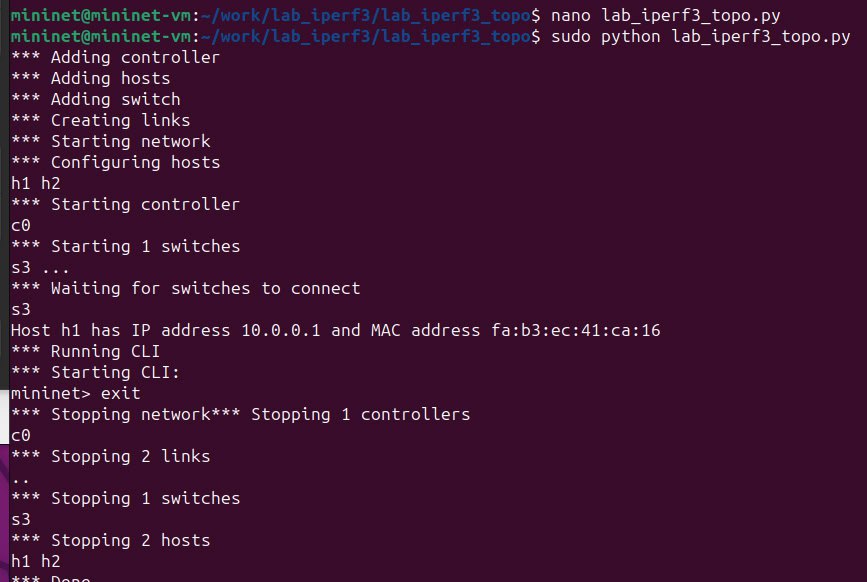


Рис. 5: Проверка работоспособности скрипта

1. 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):

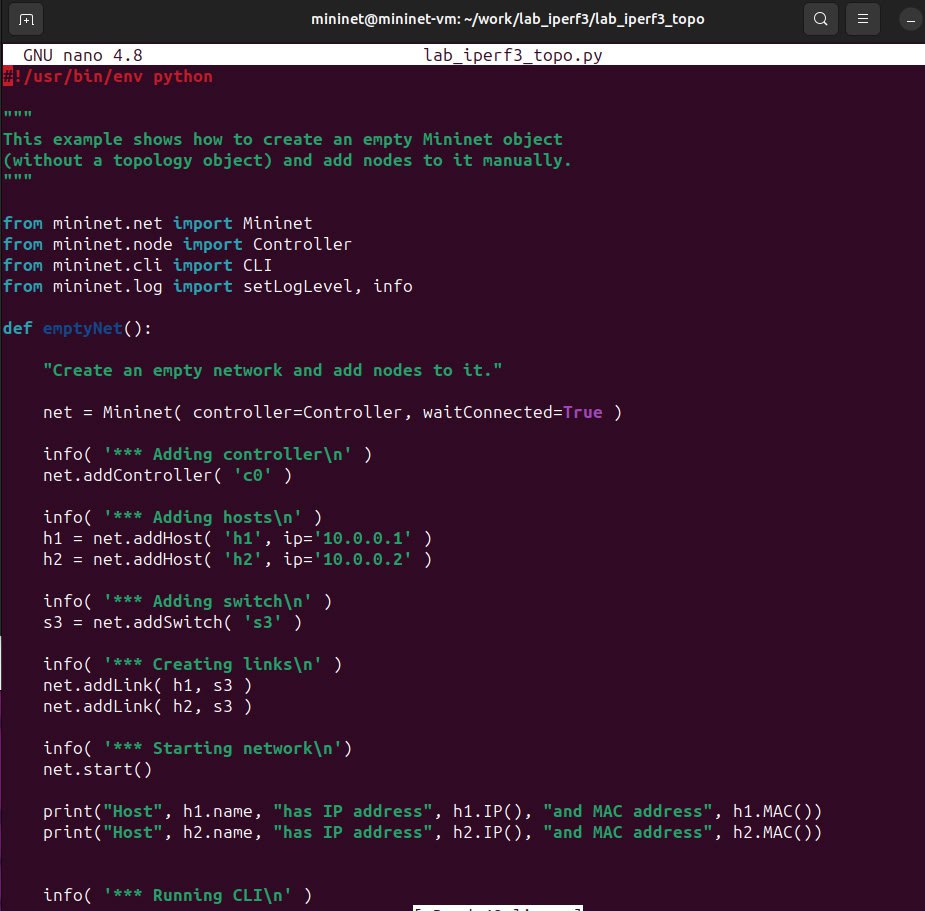


Рис. 6: Изменение скрипта для просмотра инофрмации по двум хостам

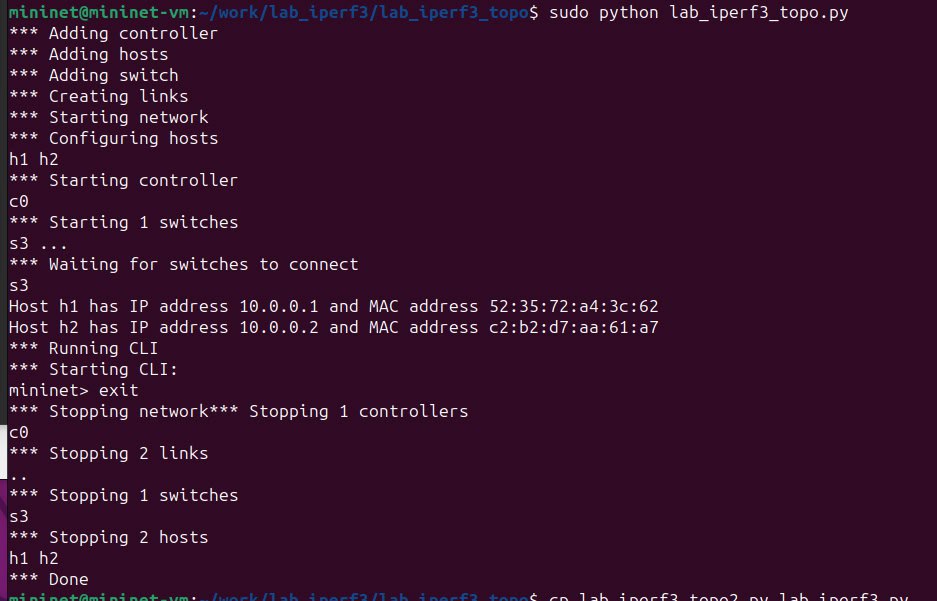


Рис. 7: Просмотр результатов работы скрипта

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

Рис. 8: Копирование скрипта lab_iperf3_topo.py

Рис. 8: Копирование скрипта lab\_iperf3\_topo.py

1. 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):

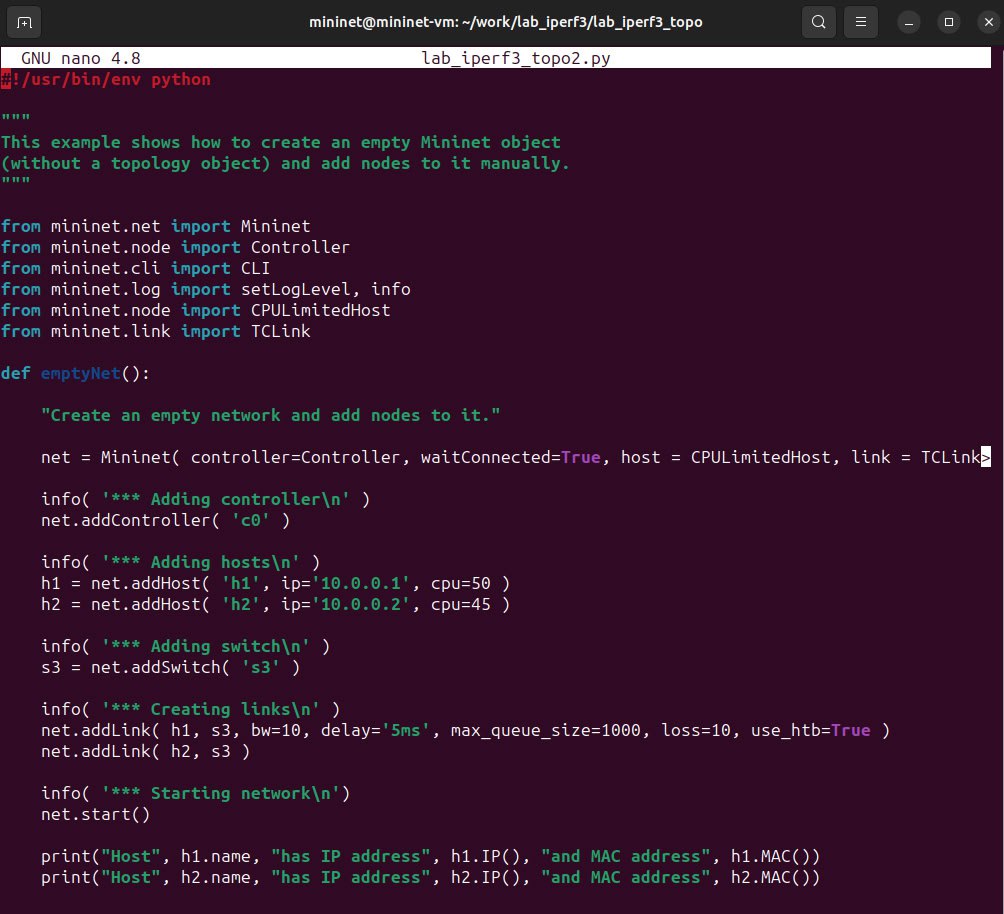


Рис. 9: Изменение скрипта: Добавляем импорт новых классов, меняем строку описание сети, задаем новые параметры для хоста h1 и h2, меняем соелиение между хостом h1 и коммутатором s3

1. Запускаем скрипт на отработку (рис. 10):

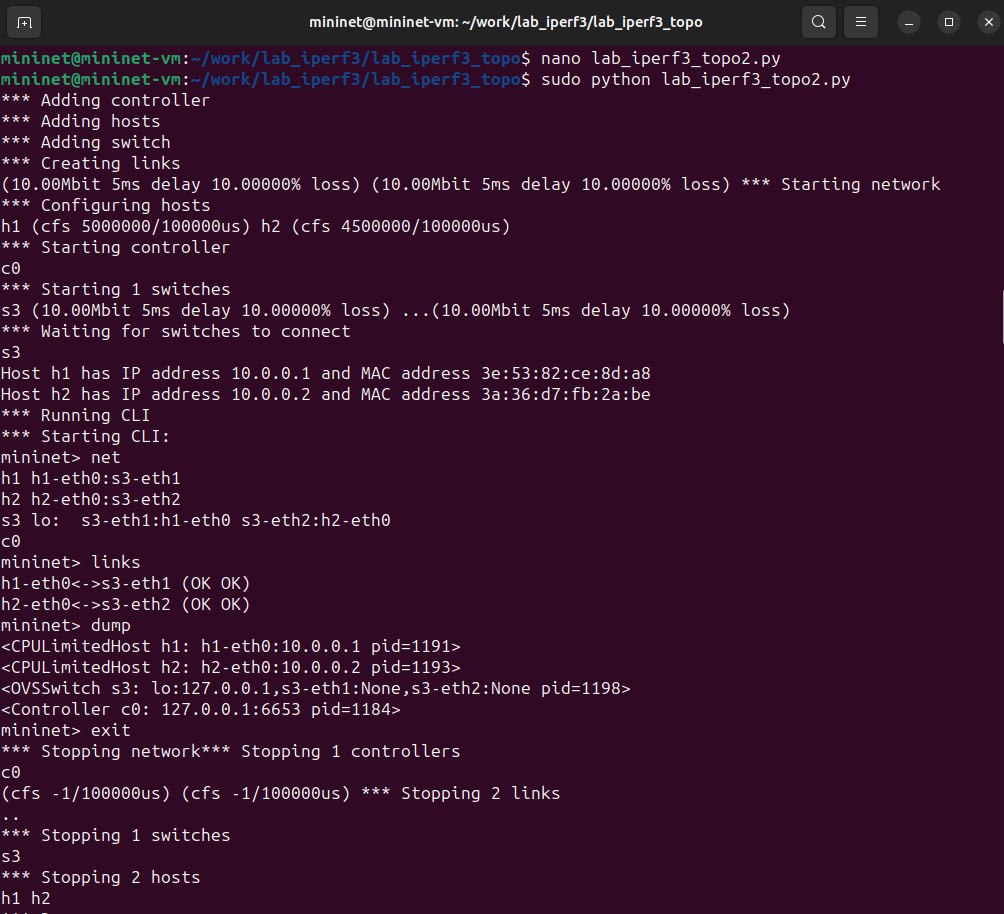


Рис. 10: Запуск скрипта и результаты работы

1. 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):

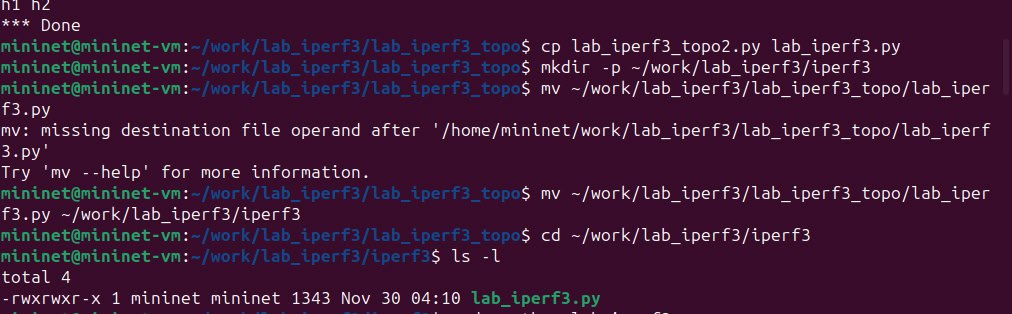


Рис. 11: Создаем новый подкаталог и копируем наш скрипт lab\_iperf\_topo2.py

1. 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
* 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

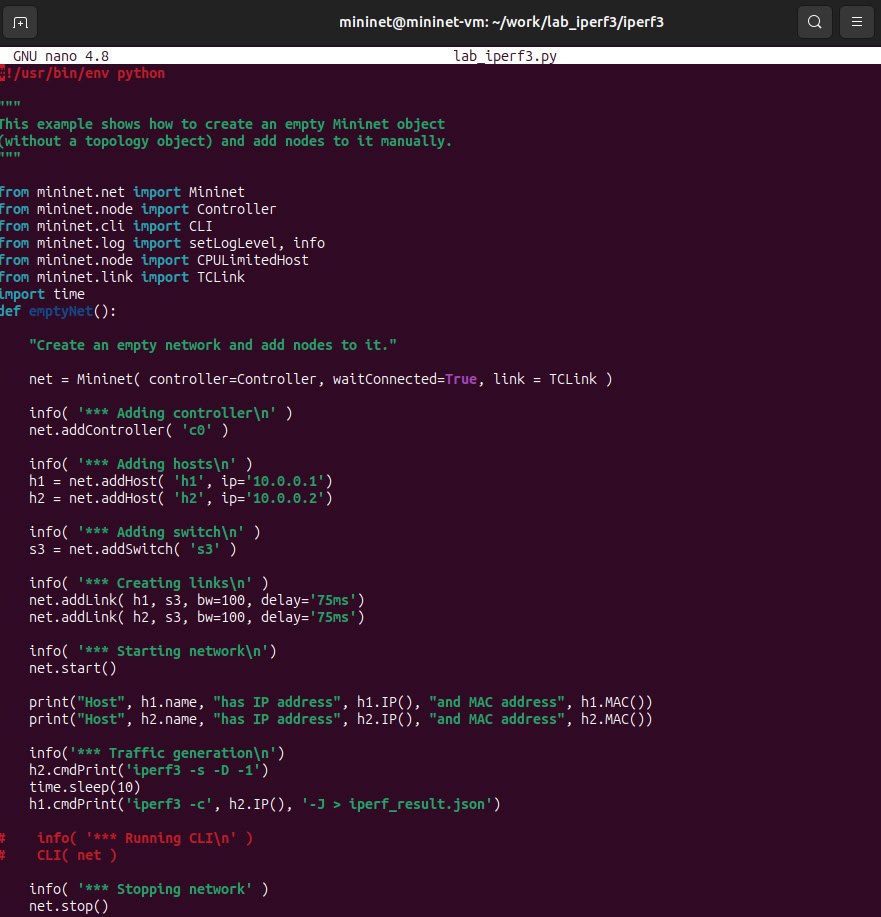


Рис. 12: Добавление в скрипт библиотеки time, изменение в работе хостов, настройка каналов между коммутатором и хостами, добавление функции записи сервера iperf3ан хосте 2 и запуска клиента через 10 секунд на хосте 1, запись результатов в файл

1. Запускаем на отработку скрипт (рис. 13):

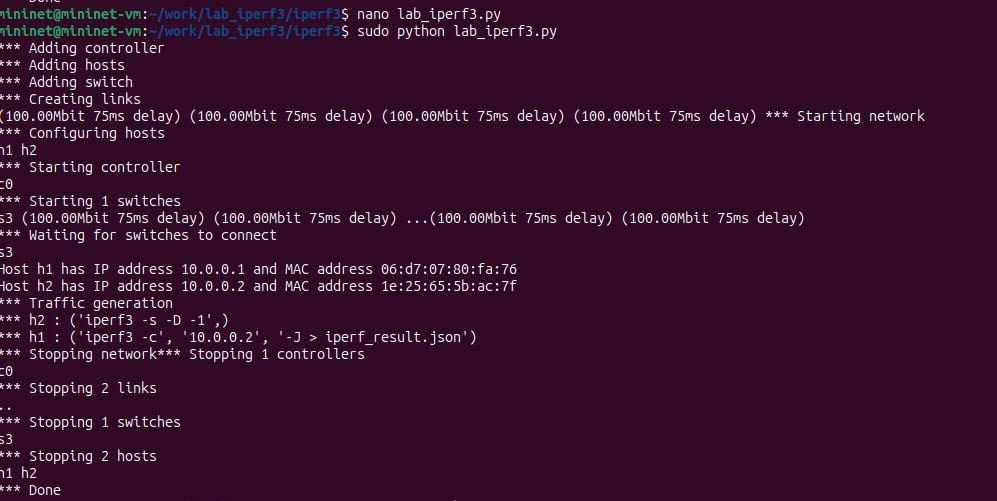


Рис. 13: Запуск скрипта lab\_iperf3.py на отработку

1. Строим графики по получившимся резулттатм работы скрипта, а так же создание Makefile для проведения эксперимента (рис. 14):

Рис. 14: Постройка графиков и создание makefile

Рис. 14: Постройка графиков и создание makefile

1. Напишем скрипт для запуска эксперимента, постройки графиков и удаление файлов из каталога с результатами (рис. 15):

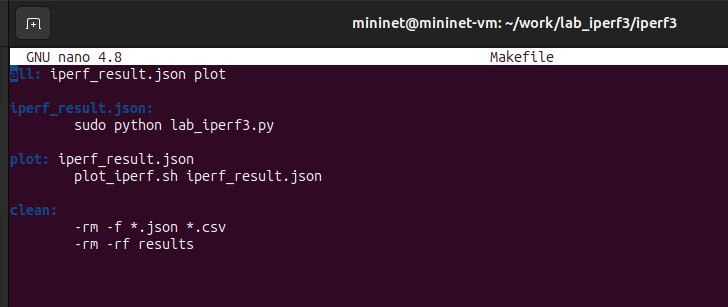


Рис. 15: Скрипт в Makefile

1. Проверка корректности работы скрипта Makefile(рис. 16):

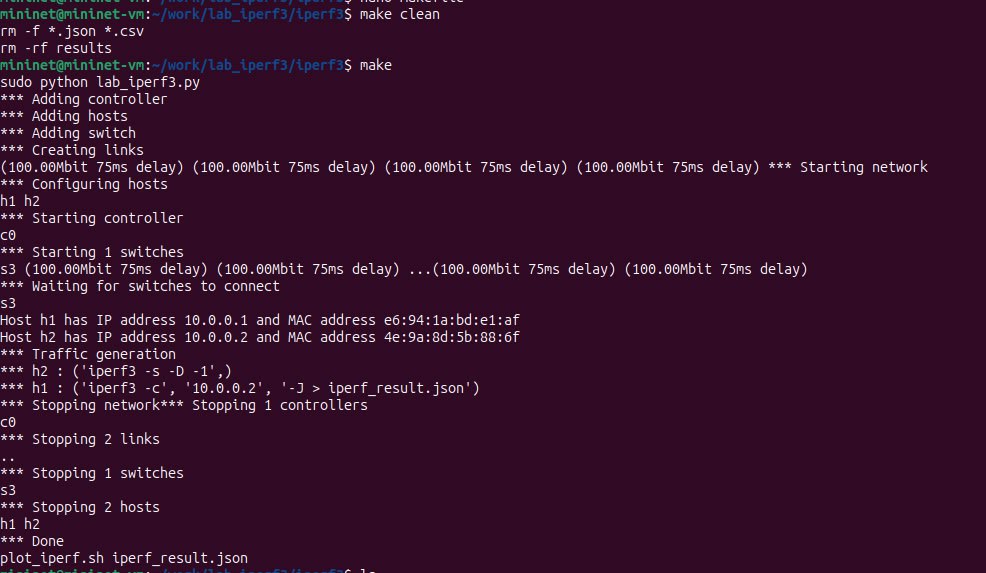


Рис. 16: Проверка корректности работы скрипта Makefile

# 3 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

# 4 Список литературы. Библиография

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