Лабораторная работа №6

Дисциплина: Моделирование сетей передачи данных

Хрусталев Влад Николаевич

Содержание

# 1 Цель работы

Основной целью работы является знакомство с принципами работы дисциплины очереди Token Bucket Filter, которая формирует входящий/исходящий трафик для ограничения пропускной способности, а также получение навыков моделирования и исследования поведения трафика посредством проведения интерактивного и воспроизводимого экспериментов в Mininet.

# 2 Теоретическое введение

Mininet[1] – это эмулятор компьютерной сети. Под компьютерной сетью подразумеваются простые компьютеры — хосты, коммутаторы, а так же OpenFlow-контроллеры. С помощью простейшего синтаксиса в примитивном интерпретаторе команд можно разворачивать сети из произвольного количества хостов, коммутаторов в различных топологиях и все это в рамках одной виртуальной машины(ВМ). На всех хостах можно изменять сетевую конфигурацию, пользоваться стандартными утилитами(ifconfig, ping) и даже получать доступ к терминалу. На коммутаторы можно добавлять различные правила и маршрутизировать трафик.

# 3 Задание

1. Задайте топологию, состоящую из двух хостов и двух коммутаторов с назначенной по умолчанию mininet сетью 10.0.0.0/8.
2. Проведите интерактивные эксперименты по ограничению пропускной способности сети с помощью TBF в эмулируемой глобальной сети.
3. Самостоятельно реализуйте воспроизводимые эксперимент по применению TBF для ограничения пропускной способности. Постройте соответствующие графики.

# 4 Выполнение лабораторной работы

## 4.1 Запуск лабораторной топологии

Запустим виртуальную машину и настроим права запуска X-соединения(рис. 1).

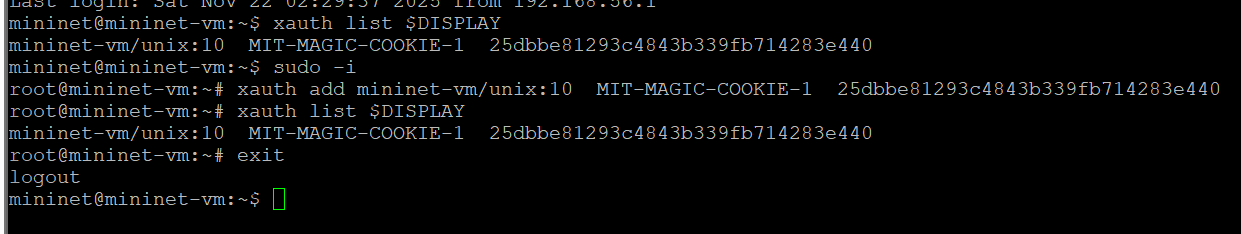


Рис. 1: Настройка X-соединения

Зададим простейшую топологию, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8 (рис. 2).

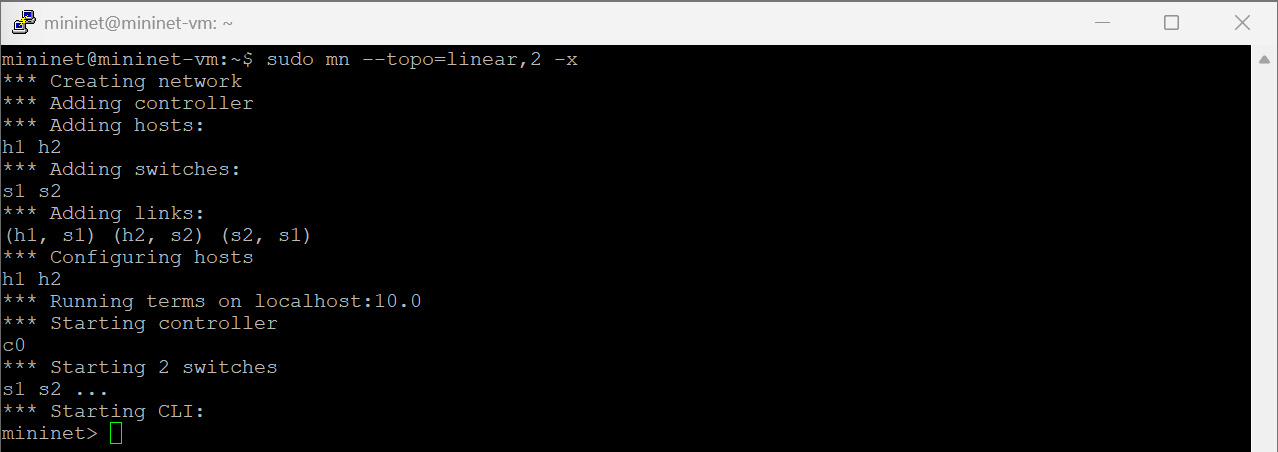


Рис. 2: Запуск простешей топологии

На хостах h1 и h2 и на коммутаторах s1, s2 введем команду ifconfig, чтобы отобразить информацию, относящуюся к их сетевым интерфейсам и назначенным им IP-адресам. В дальнейшем при работе с NETEM и командой tc будут использоваться интерфейсы h1-eth0, h2-eth0 и s1-eth2 (рис. 3)

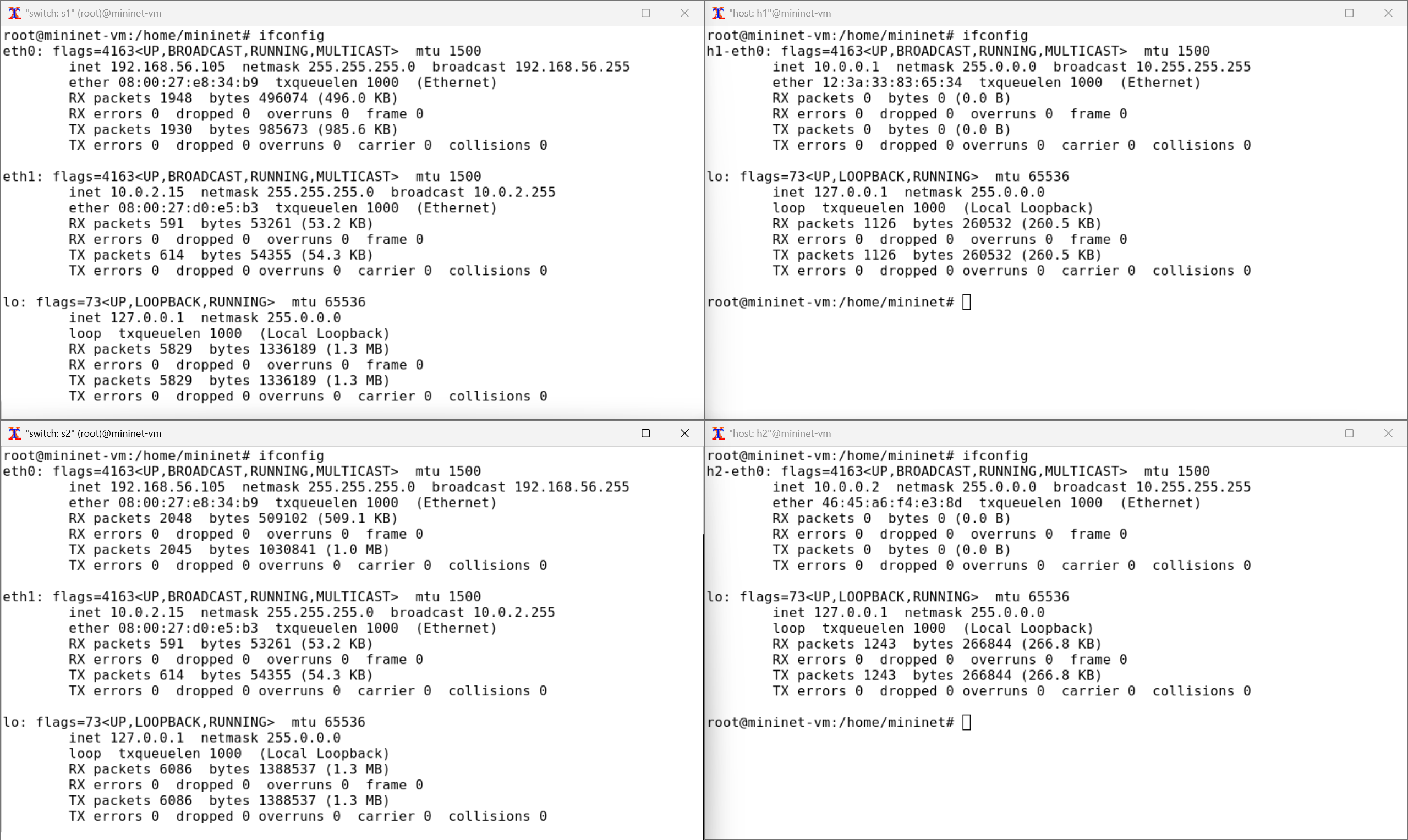


Рис. 3: Информация на хостах и коммутаторах об интерфесах

Проверим подключение между хостами h1 и h2 с помощью команды ping с параметром -c 6(рис. 4)

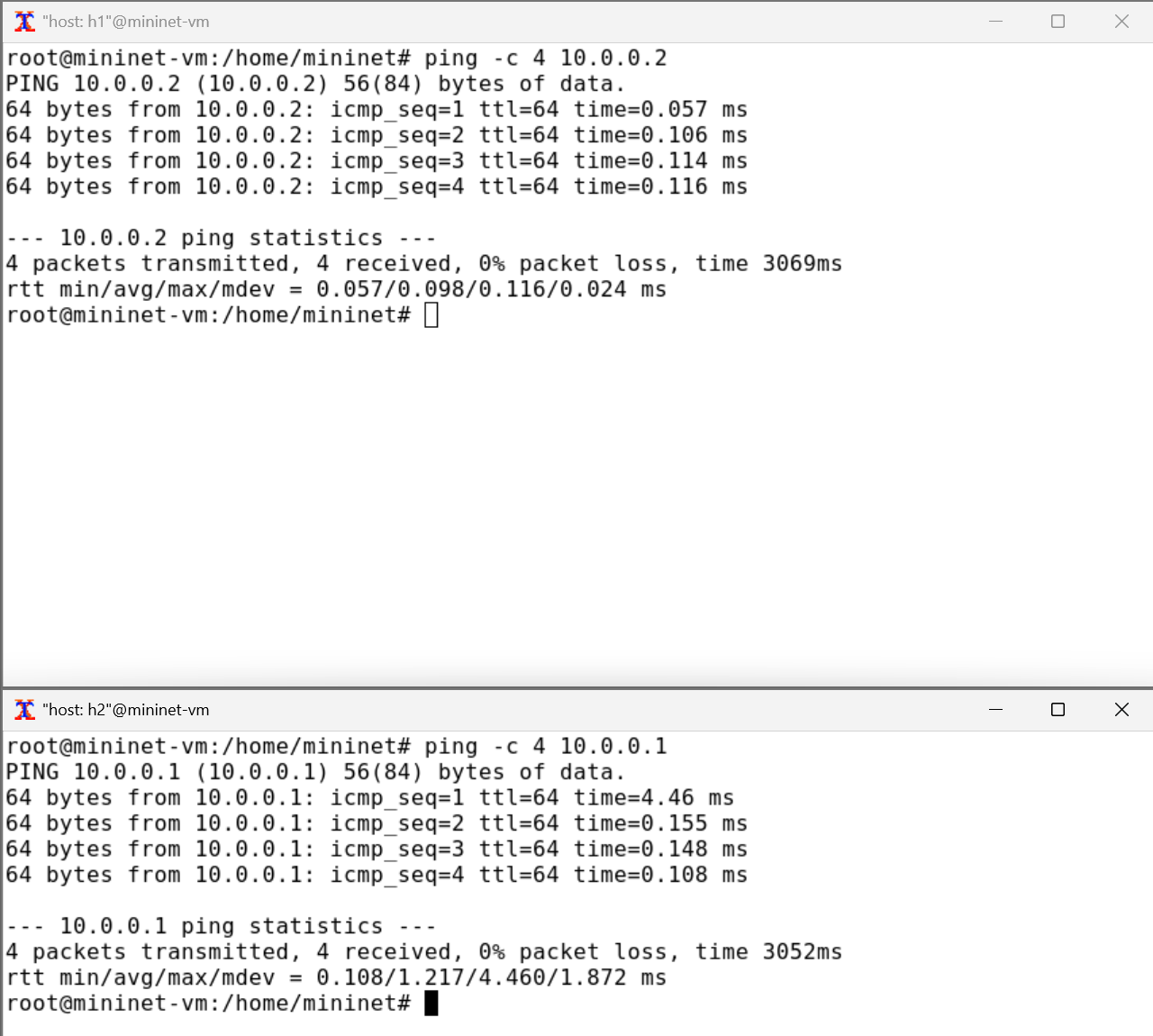


Рис. 4: Тестирование соединения между h1 и h2

В терминале хоста h2 запустим iPerf3 в режиме сервера: iperf3 -s. В терминале хоста h1 запустим iPerf3 в режиме клиента: iperf3 -c 10.0.0.2. После завершения работы iPerf3 на хосте h1 остановите iPerf3 на хосте h2, нажав Ctrl + c. (рис. 5). Как видно скорости 7-11 Gbits/sec

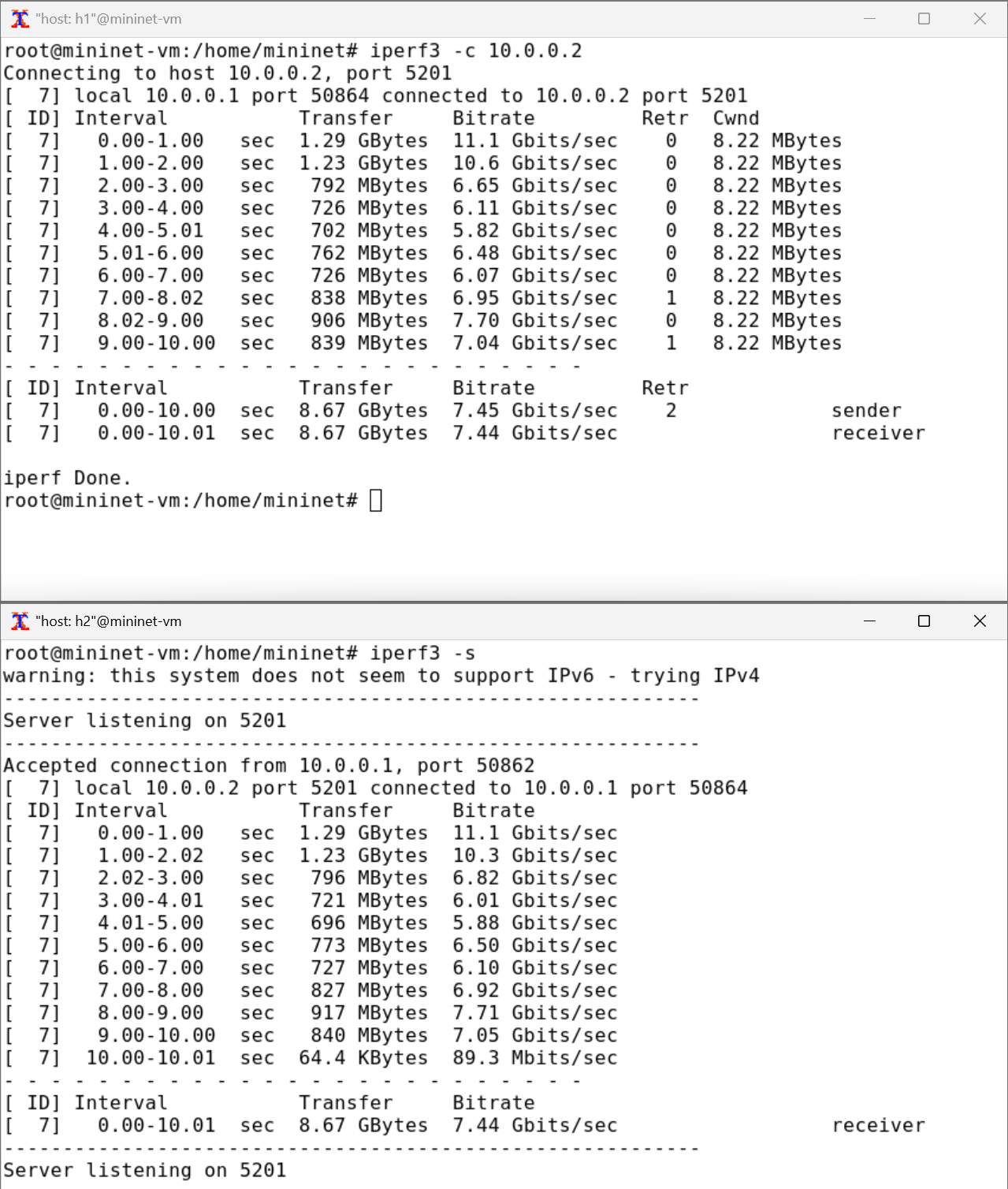


Рис. 5: Проверка изначальной пропускной способности

## 4.2 Интерактивные эксперименты

### 4.2.1 Ограничение скорости на конечных хостах

Команду tc можно применить к сетевому интерфейсу устройства для формирования исходящего трафика.Требуется ограничить скорость отправки данных с конечного хоста с помощью фильтра Token Bucket Filter (tbf).

1. Измените пропускную способность хоста h1, установив пропускную способность на 10 Гбит/с на интерфейсе h1-eth0 и параметры TBF-фильтра: sudo tc qdisc add dev h1-eth0 root tbf rate 10gbit burst 5000000 limit 15000000.

Здесь: - sudo: включить выполнение команды с более высокими привилегиями безопасности; - tc: вызвать управление трафиком Linux; - qdisc: изменить дисциплину очередей сетевого планировщика; - add (добавить): создать новое правило; - dev h1-eth0 root: интерфейс, на котором будет применяться правило; - tbf: использовать алгоритм Token Bucket Filter; - rate: указать скорость передачи (10 Гбит/с); - burst: количество байтов, которое может поместиться в корзину (5000000); - limit: размер очереди в байтах (15000000).

1. Фильтр tbf требует установки значения всплеска при ограничении скорости. Это значение должно быть достаточно высоким, чтобы обеспечить установленную скорость. Она должна быть не ниже указанной частоты, делённой на HZ, где HZ — тактовая частота, настроенная как параметр ядра, и может быть извлечена с помощью следующей команды:

egrep '^CONFIG\_HZ\_[0-9]+' /boot/config-`uname -r`

Для расчёта значения всплеска (burst) необходимо скорость передачи (10 Гбит/с или 10 Gbps = 10,000,000,000 bps) разделить на полученное таким образом значение HZ (на хосте h1 HZ = 250): Burst = 10,000,000,000 / 250 = 40,000,000 bits = 40,000,000 / 8 bytes = 5,000,000 bytes.

1. C помощью Iperf3 проверим, что значение пропускной способности изменилось

* В терминале хоста h2 запустим iPerf3 в режиме сервера: iperf3 -s
* В терминале хоста h2 запустим iPerf3 в режиме клиента: iperf3 -c 10.0.0.2
* После завершения работы iPerf3 на хосте h1 остановим iPerf3 на хосте h2, нажав Ctrl + c.

Получилась скорость передачи 7-9 Gbits/sec что соответсвует заданным ограничениям.

1. Удалим модифицированную конфигурацию на хосте h1: sudo tc qdisc del dev h1-eth0 root (рис. 6)

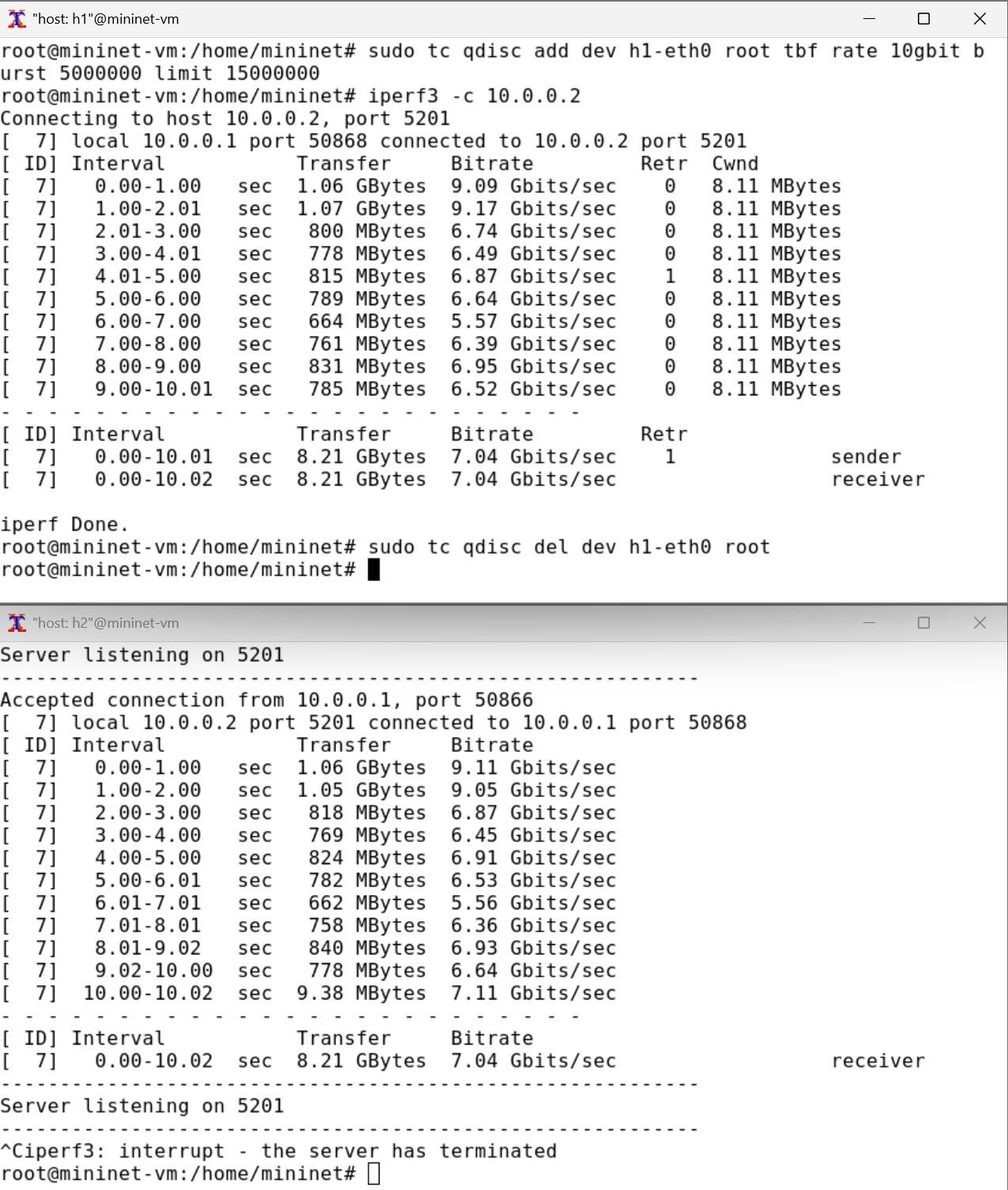


Рис. 6: Ограничение скорости на конечных хостах и тест

### 4.2.2 Ограничение скорости на коммутаторах

При ограничении скорости на интерфейсе s1-eth2 коммутатора s1 все сеансы связи между коммутатором s1 и коммутатором s2 будут фильтроваться в соответствии с применяемыми правилами.

1. Примените правило ограничения скорости tbf с параметрами rate=10gbit, burst=5,000,000, limit=15,000,000 к интерфейсу s1-eth2 коммутатора s1, который соединяет его с коммутатором s2: sudo tc qdisc add dev s1-eth2 root tbf rate 10gbit burst 5000000 limit 15000000.
2. Проверьте конфигурацию с помощью инструмента iperf3 для измерения пропускной способности(7)

* В терминале хоста h2 запустите iPerf3 в режиме сервера: iperf3 -s
* В терминале хоста h2 запустите iPerf3 в режиме клиента: iperf3 -c 10.0.0.2
* После завершения работы iPerf3 на хосте h1 остановите iPerf3 на хосте h2, нажав Ctrl + c.

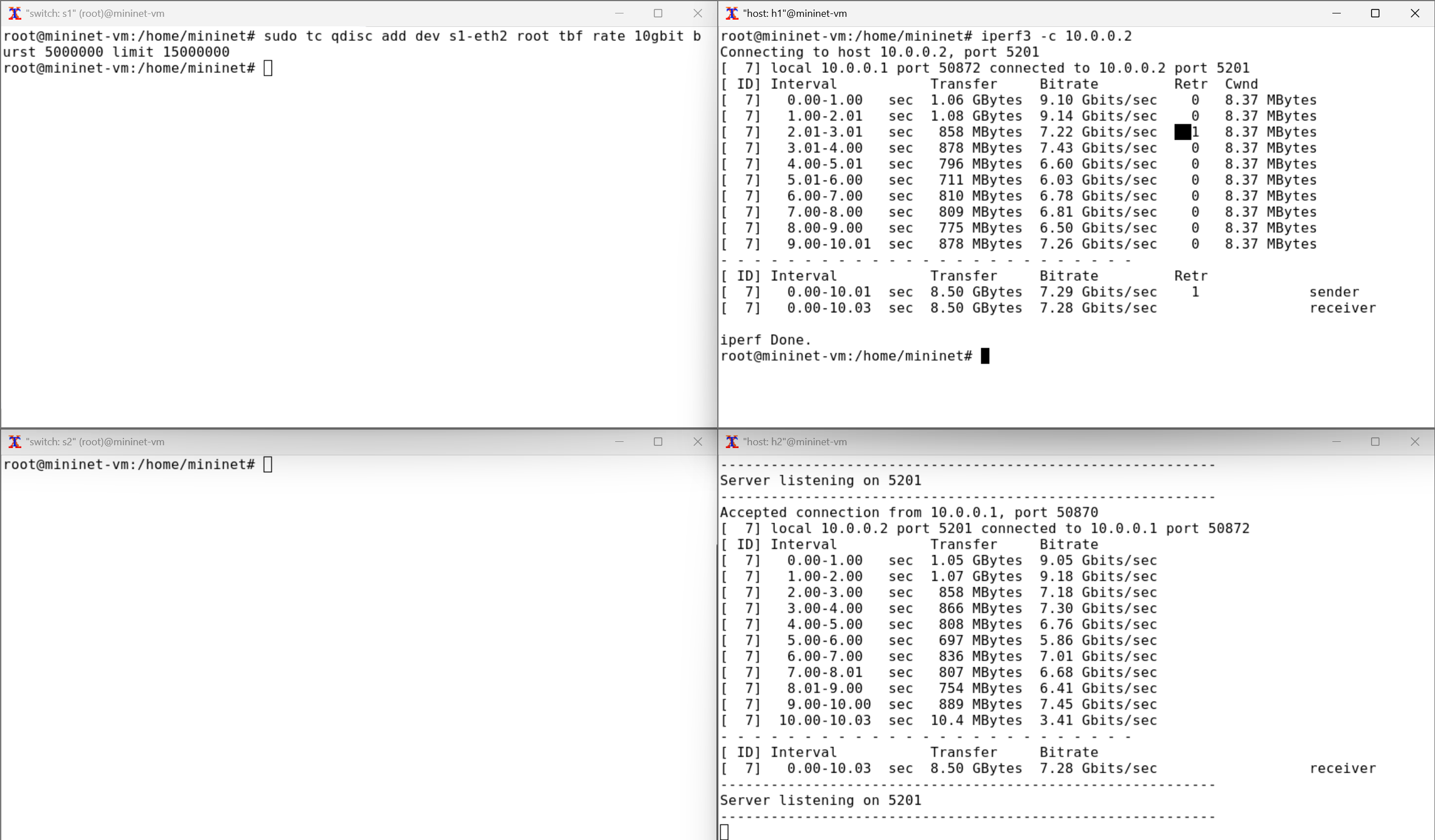


Рис. 7: Ограничение скорости на коммутаторе и тест

Скорость передачи в пределах 6-9 Gbits/sec что соответсвует правилу.

1. Удалите модифицированную конфигурацию на коммутаторе s1: sudo tc qdisc del dev s1-eth2 root (рис. 8)

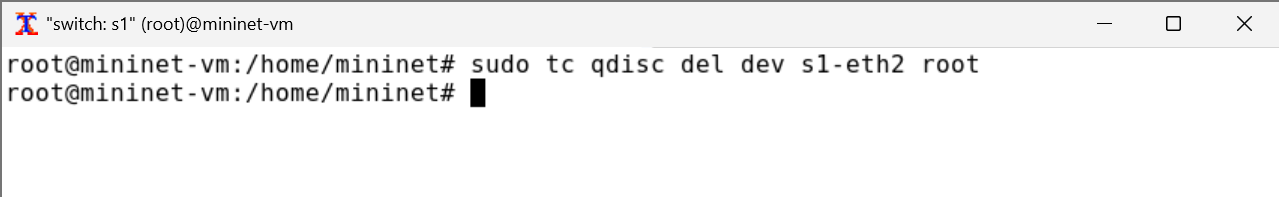


Рис. 8: Удалние модифицированной конфигурации на коммутаторе s1

### 4.2.3 Объединение NETEM и TBF

NETEM используется для изменения задержки, джиттера, повреждения пакетов и т.д. TBF может использоваться для ограничения скорости. Утилита tc позволяет комбинировать несколько модулей. При этом первая дисциплина очереди (qdisc1) присоединяется к корневой метке, последующие дисциплины очереди можно прикрепить к своим родителям, указав правильную метку.

1. Объедините NETEM и TBF, введя на интерфейсе s1-eth2 коммутатора s1 задержку, джиттер, повреждение пакетов и указав скорость: sudo tc qdisc add dev s1-eth2 root handle 1: netem delay 10ms.

Здесь ключевое слово handle задаёт дескриптор подключения, имеющий смысл очерёдности подключения разных дисциплин qdisc.

1. Убедитесь, что соединение от хоста h1 к хосту h2 имеет заданную задержку. Для этого запустите команду ping с параметром -c 4 с терминала хоста h1 (рис. 9)

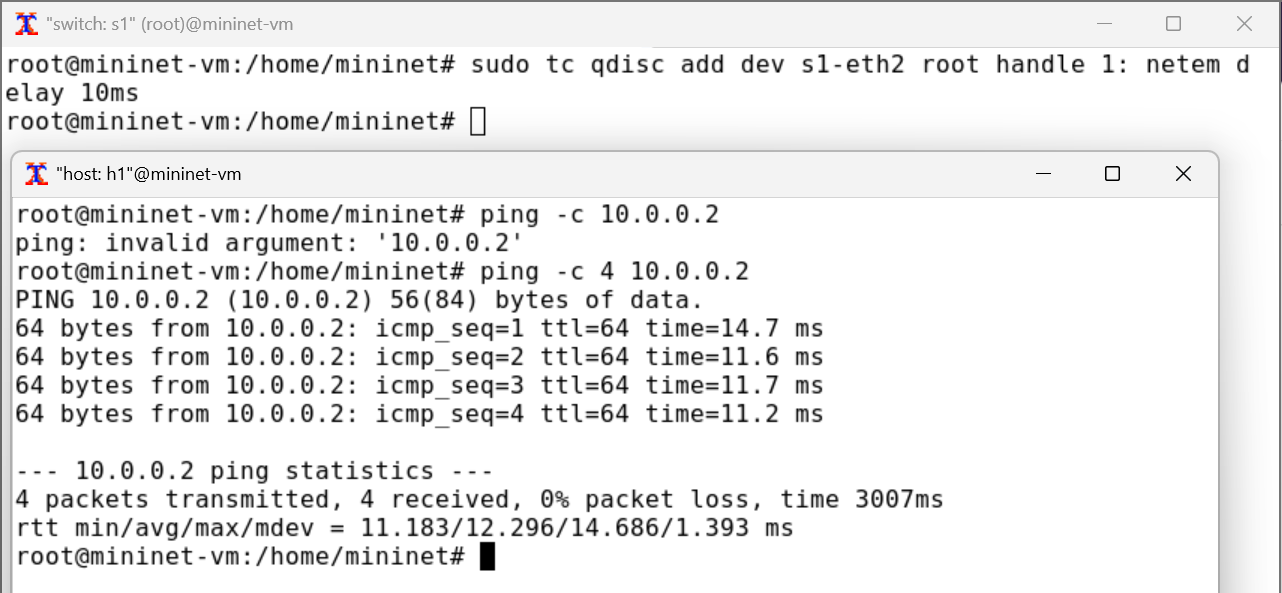


Рис. 9: Добавление NETEM правила задержки пакетов

Как мы видим что задержка от 10ms что вызвало наше правило

1. Добавьте второе правило на коммутаторе s1, которое задаёт ограничение скорости с помощью tbf с параметрами rate=2gbit, burst=1,000,000, limit=2,000,000: sudo tc qdisc add dev s1-eth2 parent 1: handle 2: tbf rate 2gbit burst 1000000 limit 2000000.
2. Проверьте конфигурацию с помощью инструмента iperf3 для измерения пропускной способности(рис. 10)

* В терминале хоста h2 запустите iPerf3 в режиме сервера: iperf3 -s
* В терминале хоста h2 запустите iPerf3 в режиме клиента: iperf3 -c 10.0.0.2
* После завершения работы iPerf3 на хосте h1 остановите iPerf3 на хосте h2, нажав Ctrl + c. В отчёте зафиксируйте результат работы iPerf3 на данном этапе проведения эксперимента.

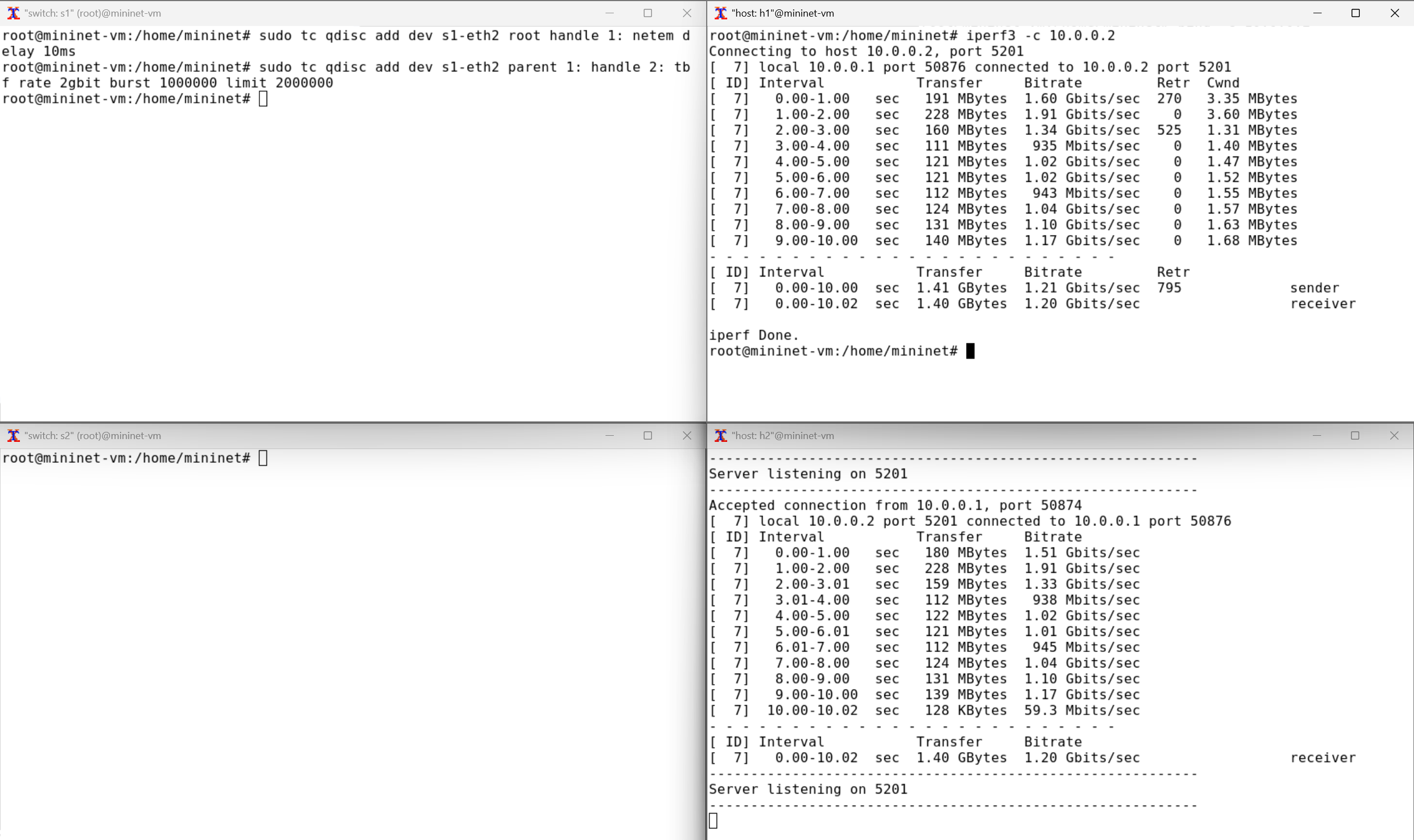


Рис. 10: Добавление 2ого правила ограничения скорости и тест

получился битрейт 0.9 - 1.9 Gbits/sec что также соответсвует заданным правиом ограничения скрости.

1. Удалите модифицированную конфигурацию на коммутаторе s1: sudo tc qdisc del dev s1-eth2 root (рис. 11)



Рис. 11: Удалине модифицированной конфигурации на коммутаторе s1

## 4.3 Воспроизведение экспериментов

1. Для воспроизводлимого эксперемента создам каталог ~/work/lab\_tbf\_i, в нём создаим файлы для воспроивзодимого эксперемента (рис. 12).
2. Далее перенёс полученную структуры по папкам ~/work/lab\_tbf\_i/exp1, ~/work/lab\_tbf\_i/exp2, ~/work/lab\_tbf\_i/exp3 соответсвующим нашим эксперементам.

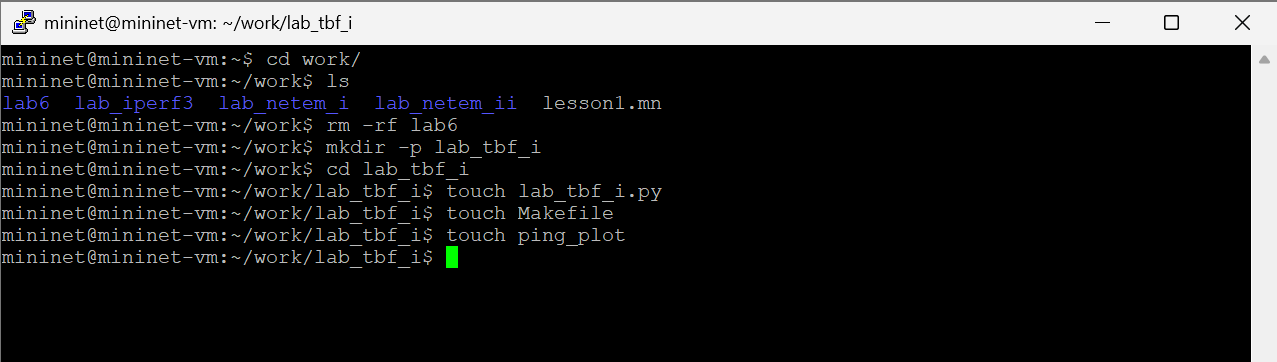


Рис. 12: Создание необходимых файлов

Для начала воспроизведём эксперент “Ограничение скорости на конечных хоста”. Создадим и заполним файл exp1/lab\_tbf\_i.py(рис. 13), заполним файлы Makefile для выполнения эксперемента и ping\_plot для построения графиков (рис. 14)

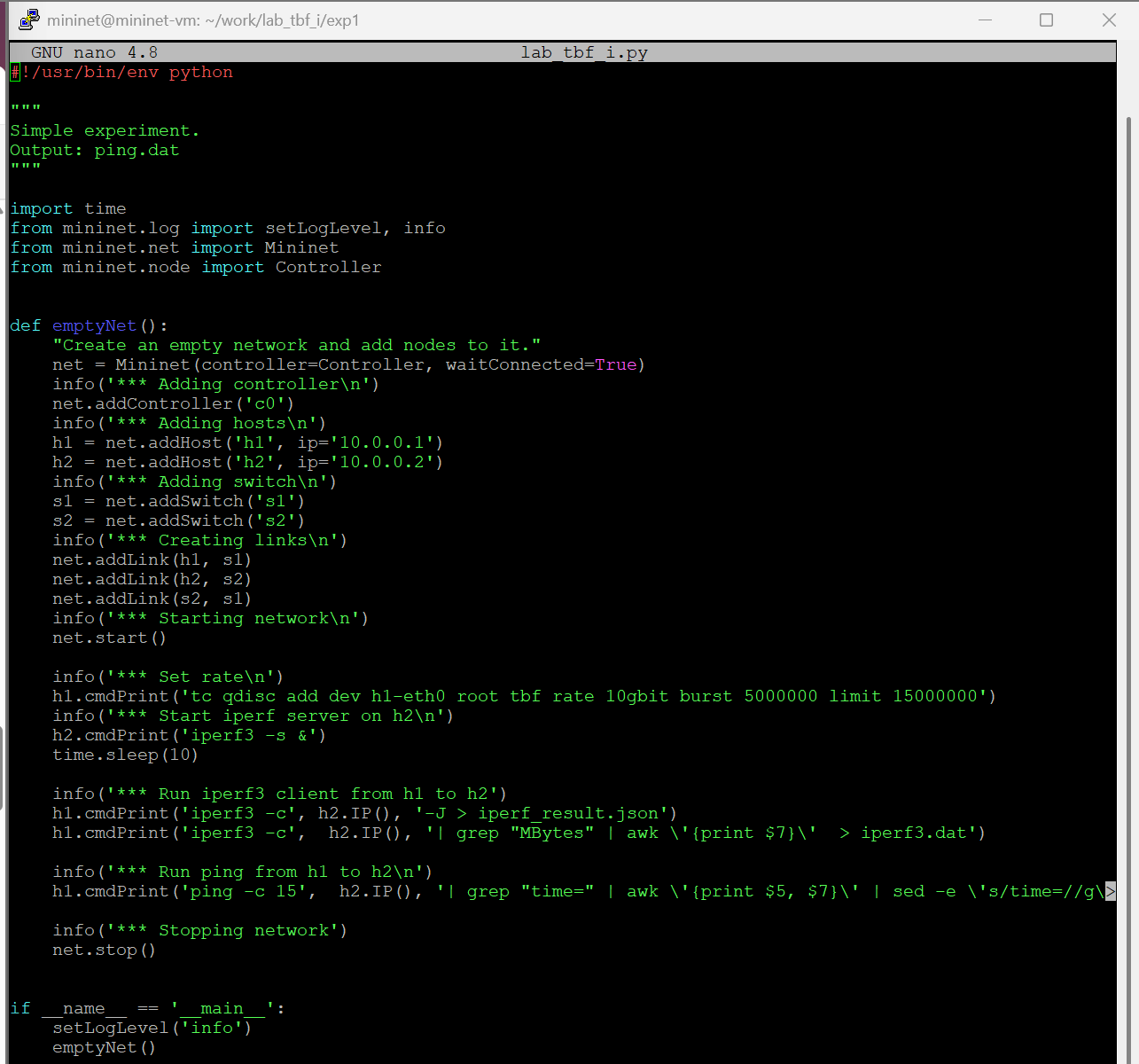


Рис. 13: Лиcтинг exp1/lab\_tbf\_i.py

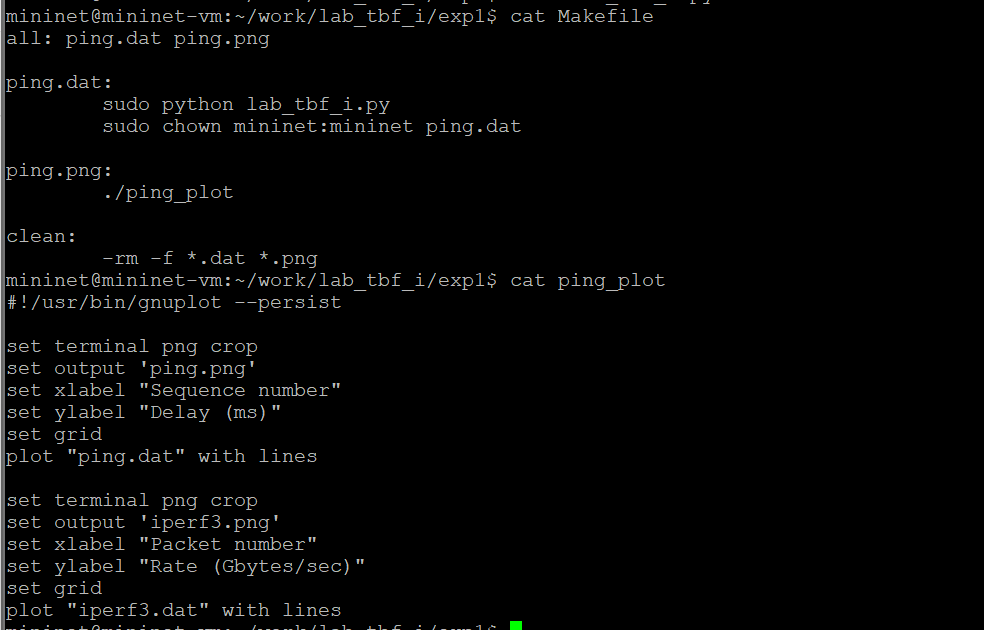


Рис. 14: Листинг Makefile и ping\_plot

Запустим эксперемент exp1 командой make(рис. 15)

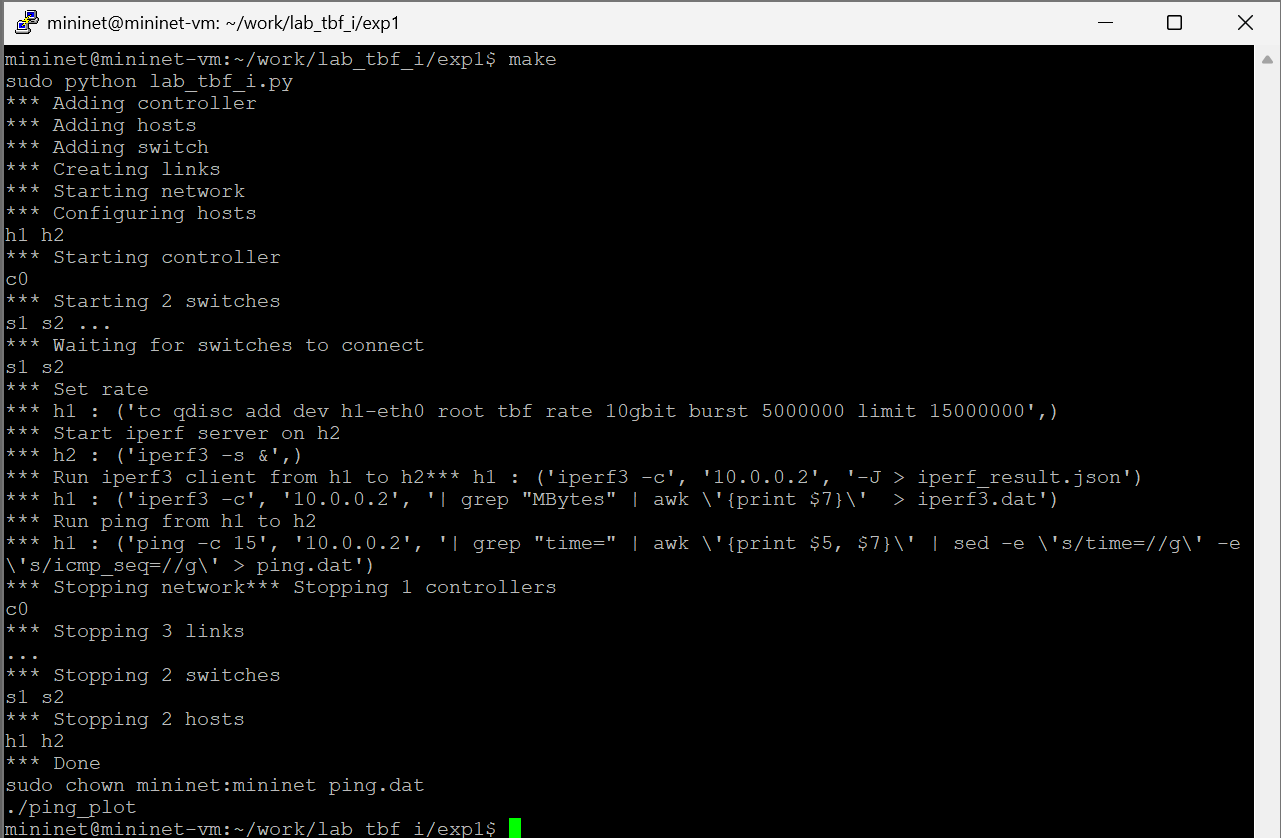


Рис. 15: Запуск exp1

Просмотрим полученные графики iperf3.png (рис. 16) и ping.png(рис. 17)

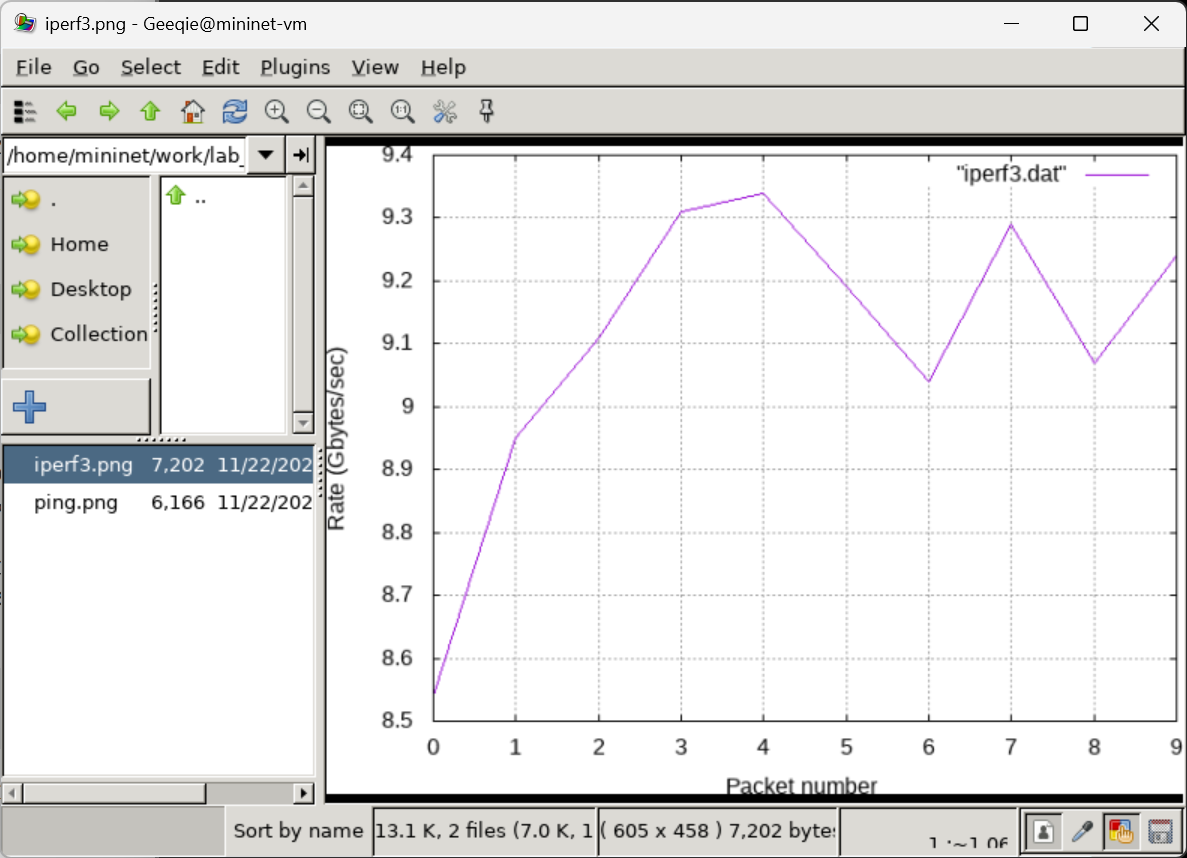


Рис. 16: График iperf3.png из exp1

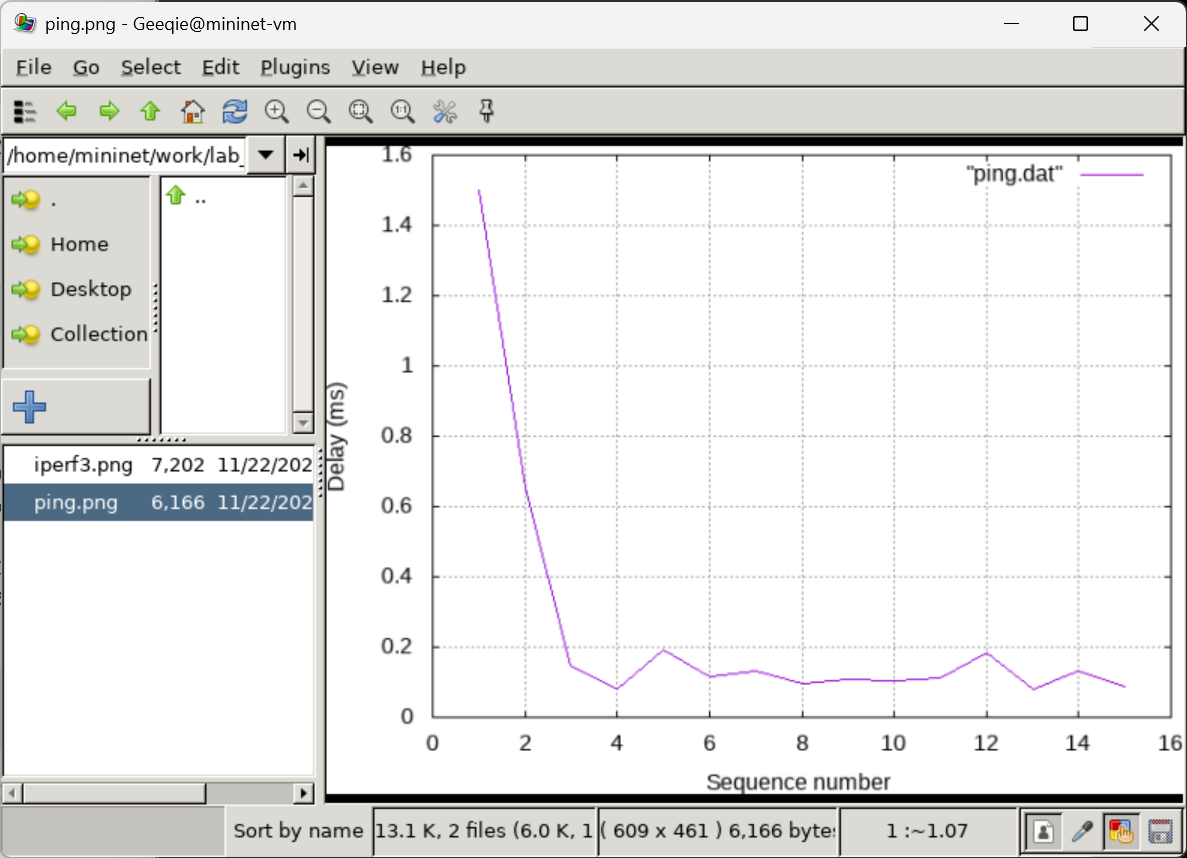


Рис. 17: График ping.png из exp1

По аналогии отредактируем файл exp2/lab\_tbf\_i.py для эксперемента ” Ограничение скорости на коммутаторах”(рис. 18). Остальные файлы оставим такими же как в первом эксперементе.

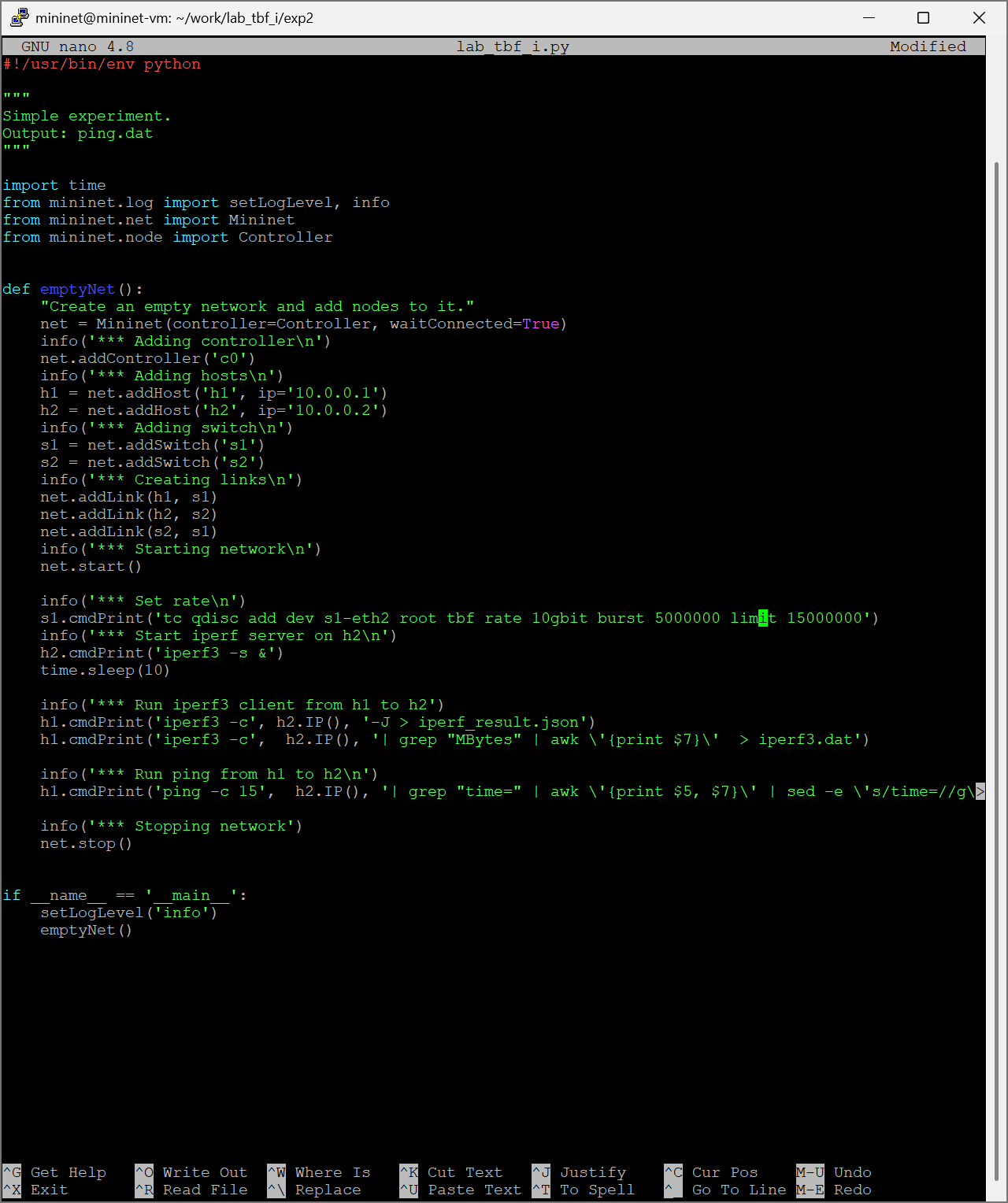


Рис. 18: Лиcтинг exp2/lab\_tbf\_i.py

Запустим эксперемент exp2 командой make(рис. 19)

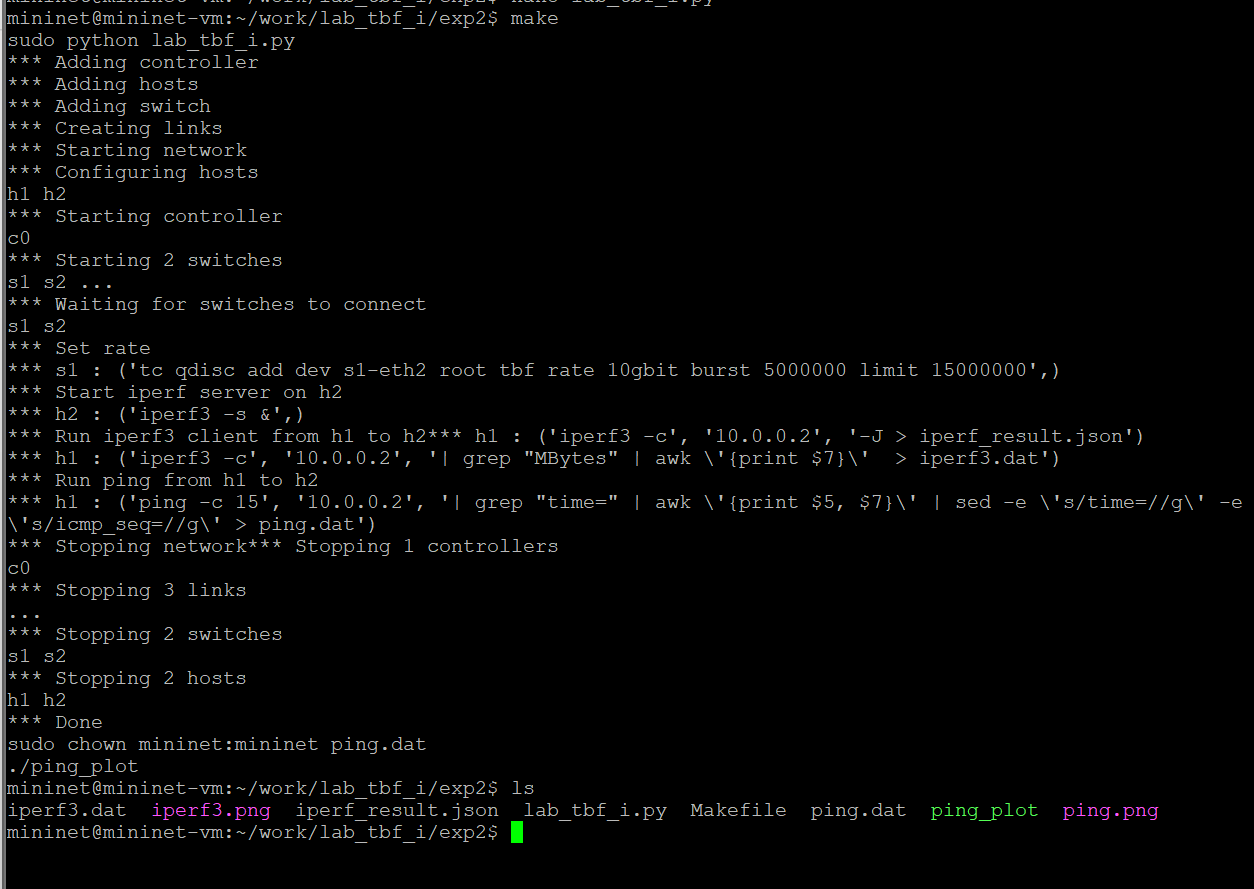


Рис. 19: Запуск exp2

Просмотрим полученные графики iperf3.png (рис. 20) и ping.png(рис. 21)



Рис. 20: График iperf3.png из exp2

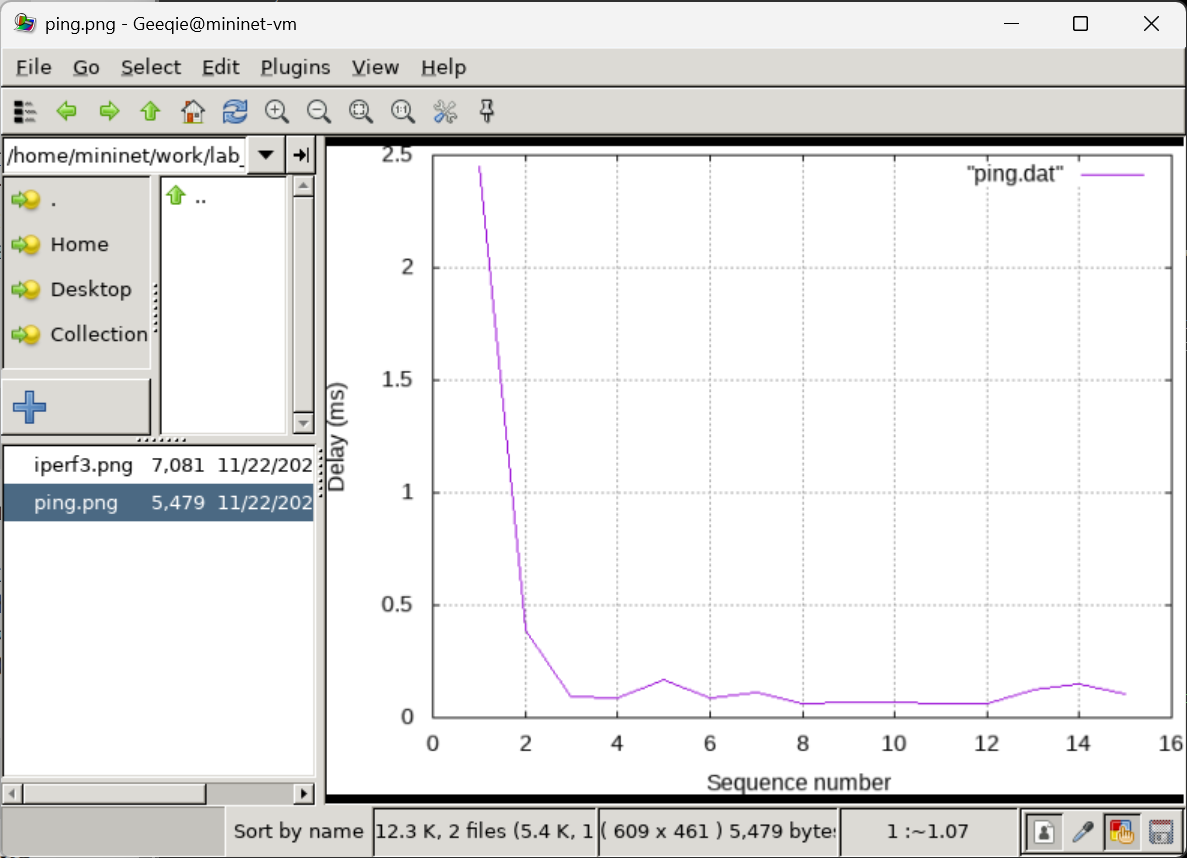


Рис. 21: График ping.png из exp2

Таким же образом проведём эксперемент “Объединение NETEM и TBF”. Отредактируем exp3/lab\_tbf\_i.py(рис. 22)

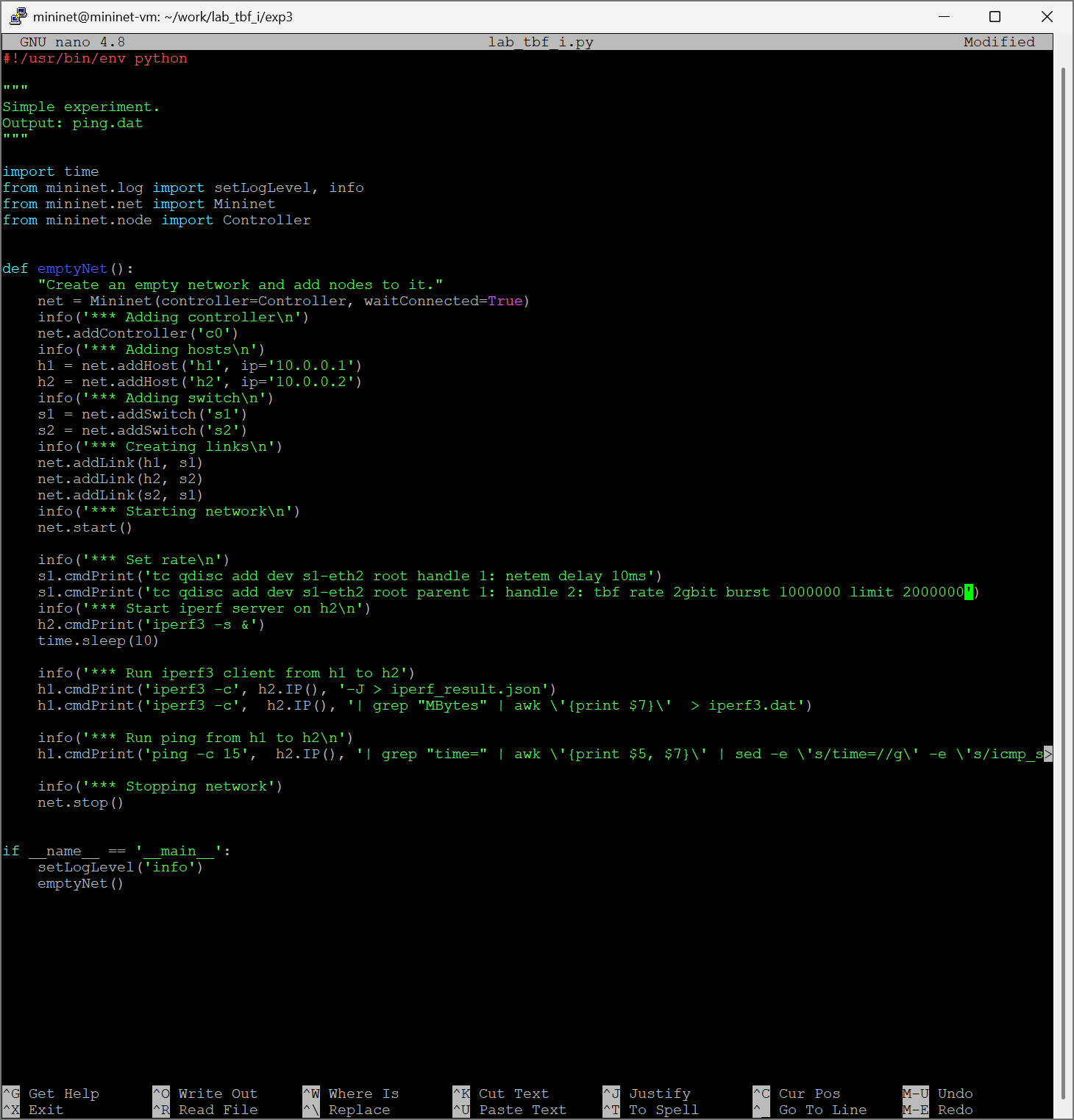


Рис. 22: Лиcтинг exp3/lab\_tbf\_i.py

Запустим эксперемент exp3 командой make(рис. 23)

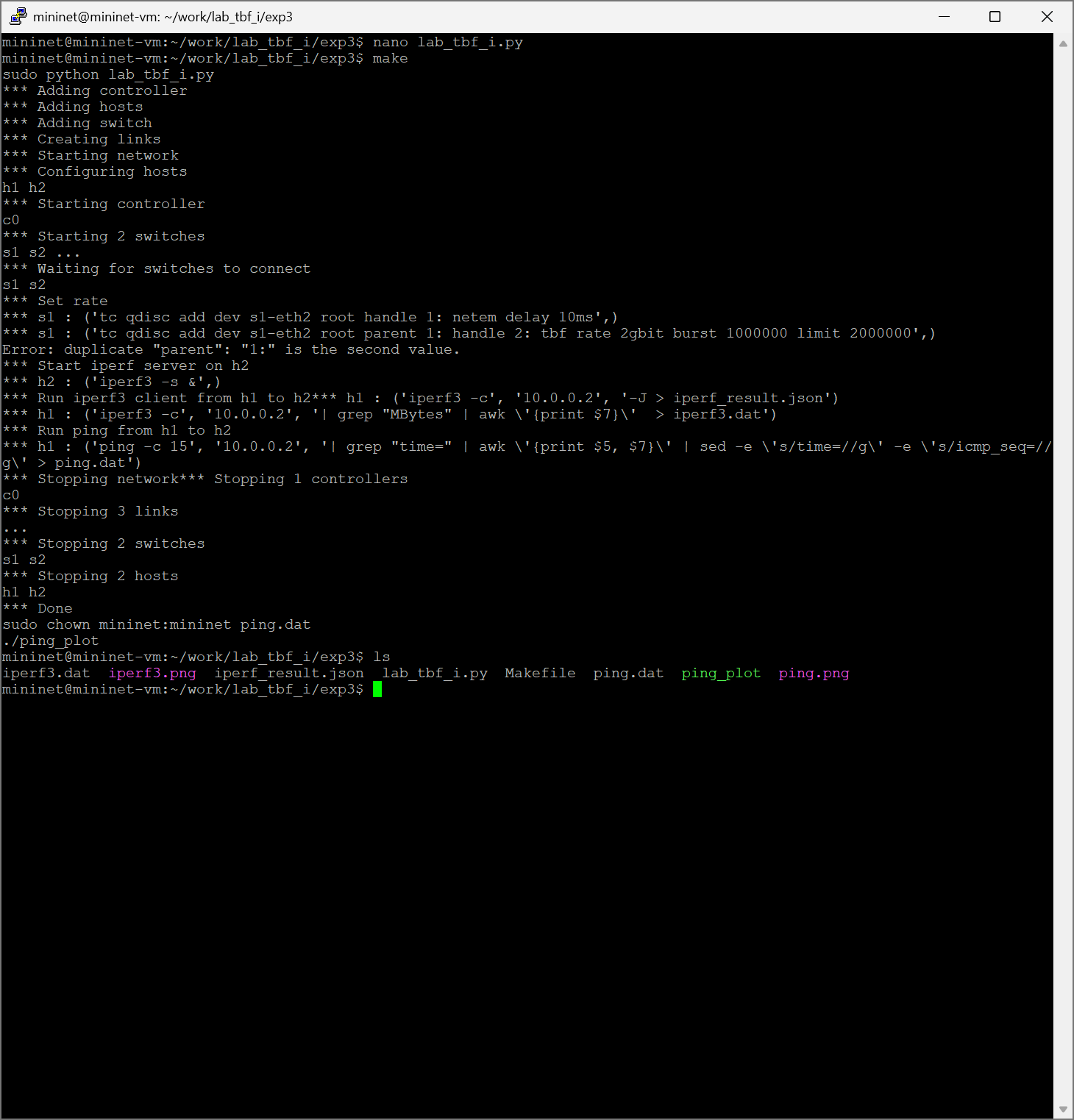


Рис. 23: Запуск exp3

Просмотрим полученные графики iperf3.png (рис. 24) и ping.png(рис. 25)

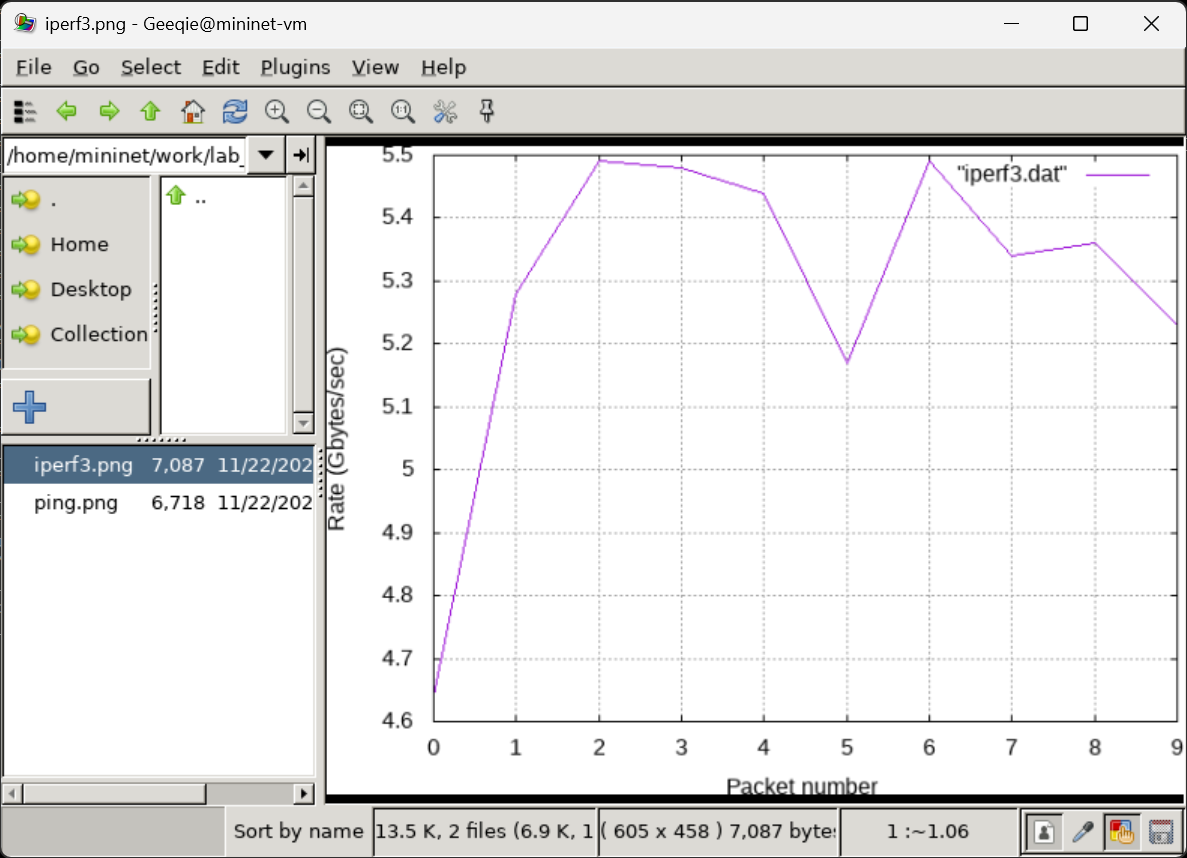


Рис. 24: График iperf3.png из exp3

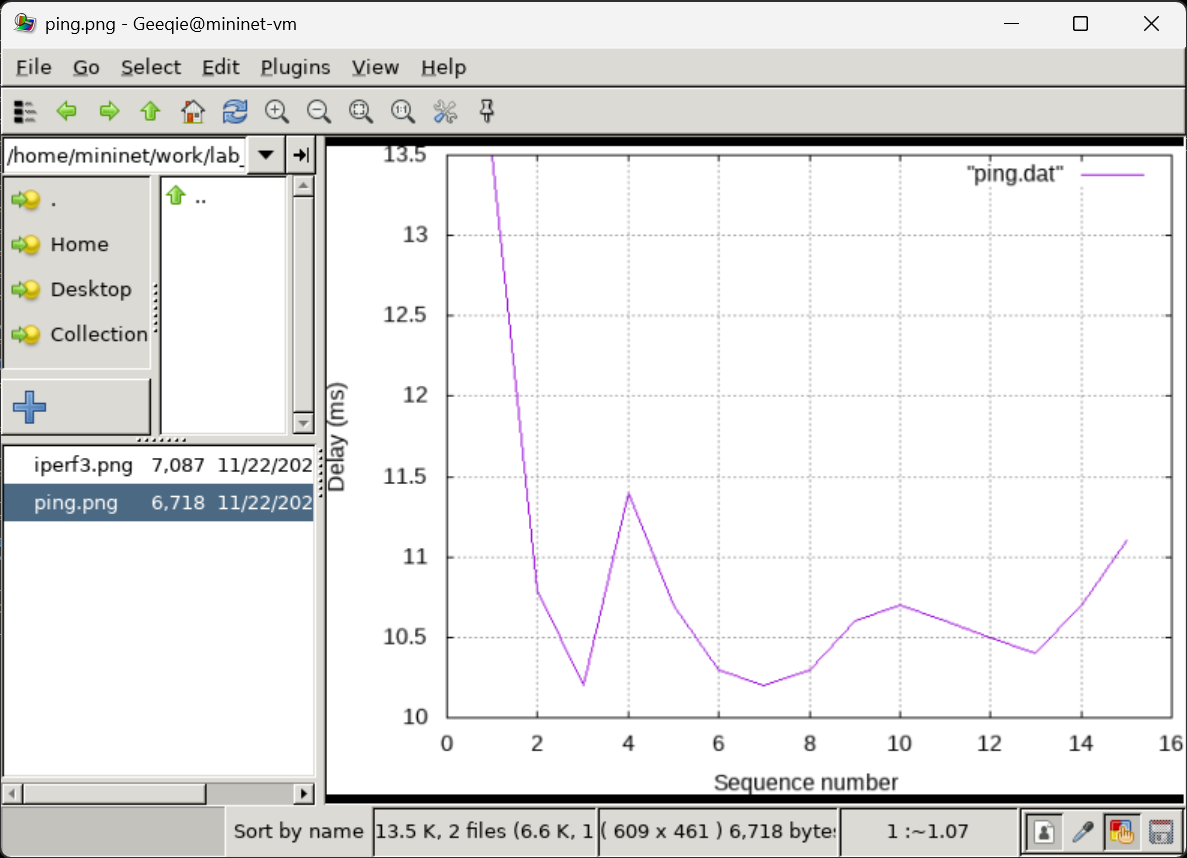


Рис. 25: График ping.png из exp1

# 5 Выводы

В результате выполнения работы я познакомился с принципами работы Token Bucket Filter, а также получили навыки моделирования и исследования поведения трафика посредством проведения интерактивного и воспроизводимого экспериментов в Mininet.

# Список литературы

1. Mininet [Электронный ресурс]. Mininet Project Contributors. URL: <http://mininet.org/> (дата обращения: 07.10.2025).