Лабораторная работа №4

Моделирование сетей передачи данных

Хрусталев Влад Николаевич

Содержание

# 1 Цель работы

Основной целью работы является знакомство с NETEM — инструментом для тестирования производительности приложений в виртуальной сети, а также получение навыков проведения интерактивного и воспроизводимого экспериментов по измерению задержки и её дрожания (jitter) в моделируемой сети в среде 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 простейшую топологию, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8.
2. Проведите интерактивные эксперименты по добавлению/изменению задержки, джиттера, значения корреляции для джиттера и задержки, распределения времени задержки в эмулируемой глобальной сети.
3. Реализуйте воспроизводимый эксперимент по заданию значения задержки в эмулируемой глобальной сети. Постройте график.
4. Самостоятельно реализуйте воспроизводимые эксперименты по изменению задержки, джиттера, значения корреляции для джиттера и задержки, распределения времени задержки в эмулируемой глобальной сети. Постройте графики.

# 4 Выполнение лабораторной работы

Запустим виртуальную среду с mininet. Из основной ОС подключимся к виртуальной машине. В виртуальной машине mininet при необходимости исправим права запуска X-соединения. Скопируем значение куки (MIT magic cookie) своего пользователя mininet в файл для пользователя root (рис. 1).

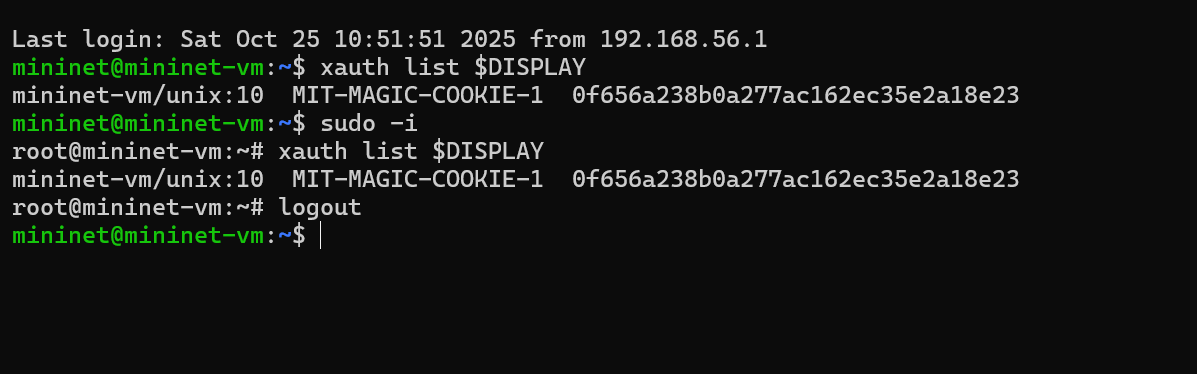


Рис. 1: Исправление прав запуска X-соединения

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

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



Рис. 2: Простейшая топология

На хостах h1 и h2 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 и h2-eth0 (рис. 3).

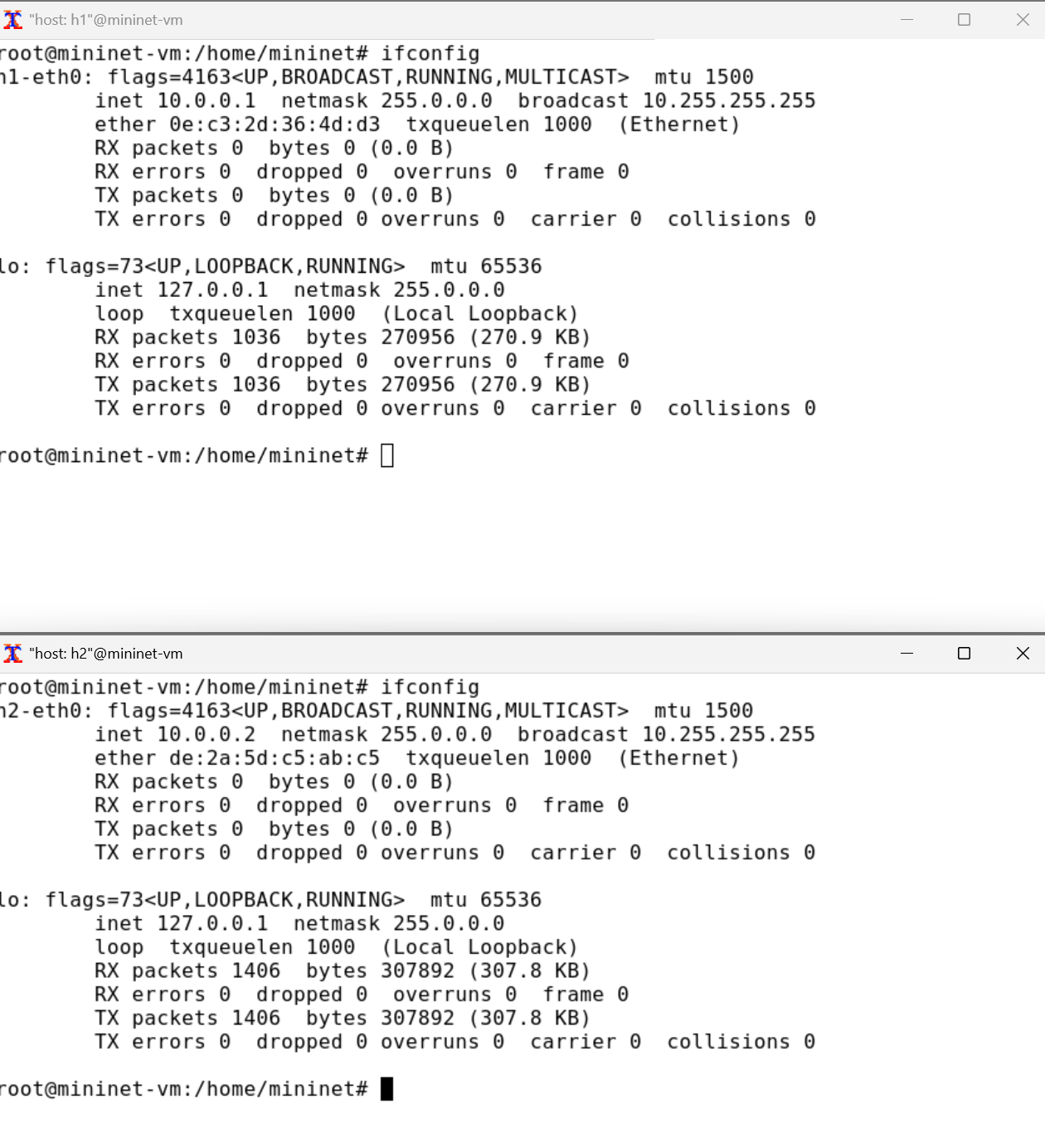


Рис. 3: ifconfig на хостах h1 и h2

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

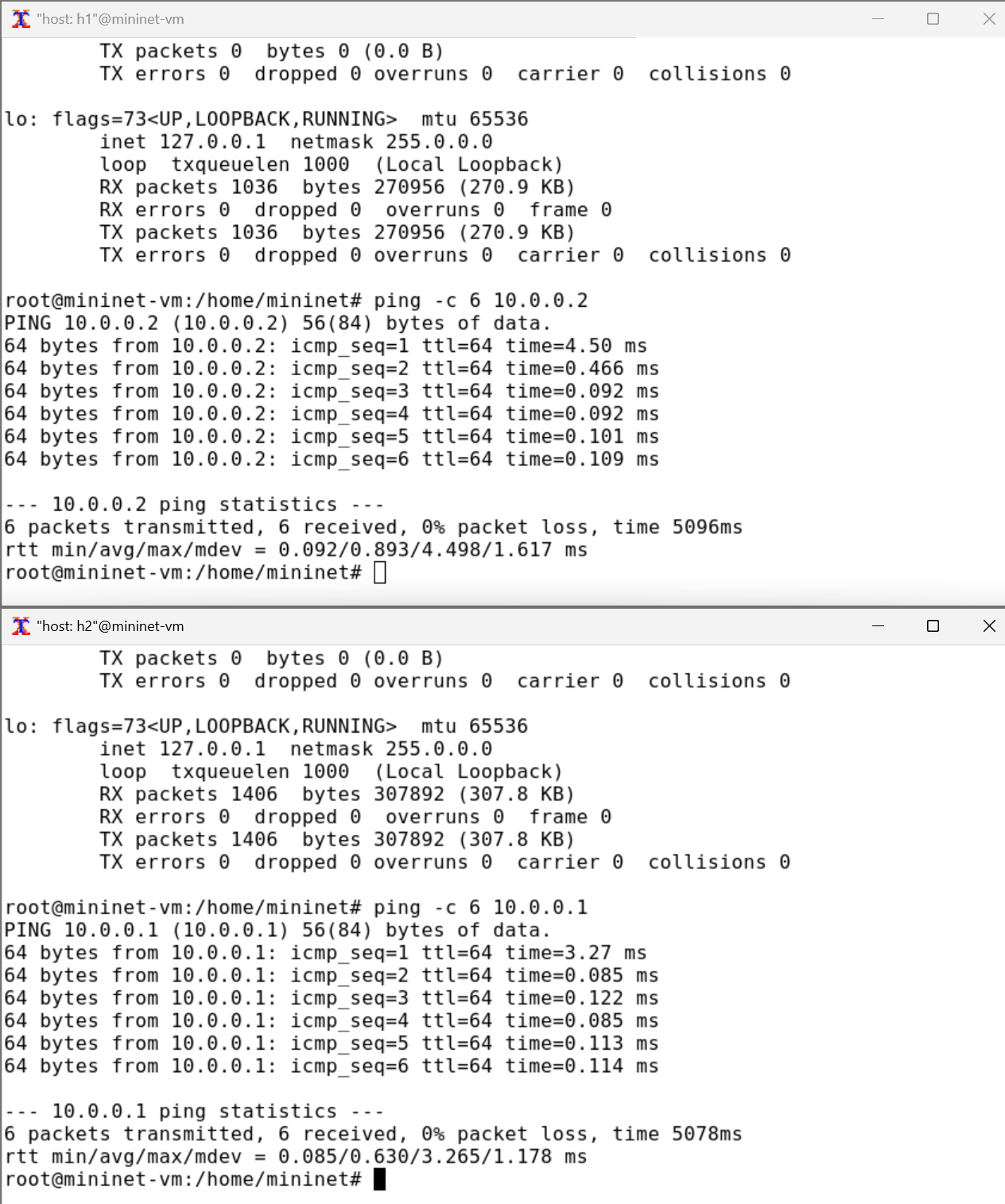


Рис. 4: Проверка подключения между хостами

Добавление/изменение задержки в эмулируемой глобальной сети

На хосте h1 добавим задержку в 100 мс к выходному интерфейсу (рис. 5).

sudo tc qdisc add dev h1-eth0 root netem delay 100ms

* sudo: выполнить команду с более высокими привилегиями;
* tc: вызвать управление трафиком Linux;
* qdisc: изменить дисциплину очередей сетевого планировщика;
* add: создать новое правило;
* dev h1-eth0: указать интерфейс, на котором будет применяться правило;
* netem: использовать эмулятор сети;
* delay 100ms: задержка ввода 100 мс.

Проверим, что соединение от хоста h1 к хосту h2 имеет задержку 100 мс, используя команду ping с параметром -c 6 с хоста h1

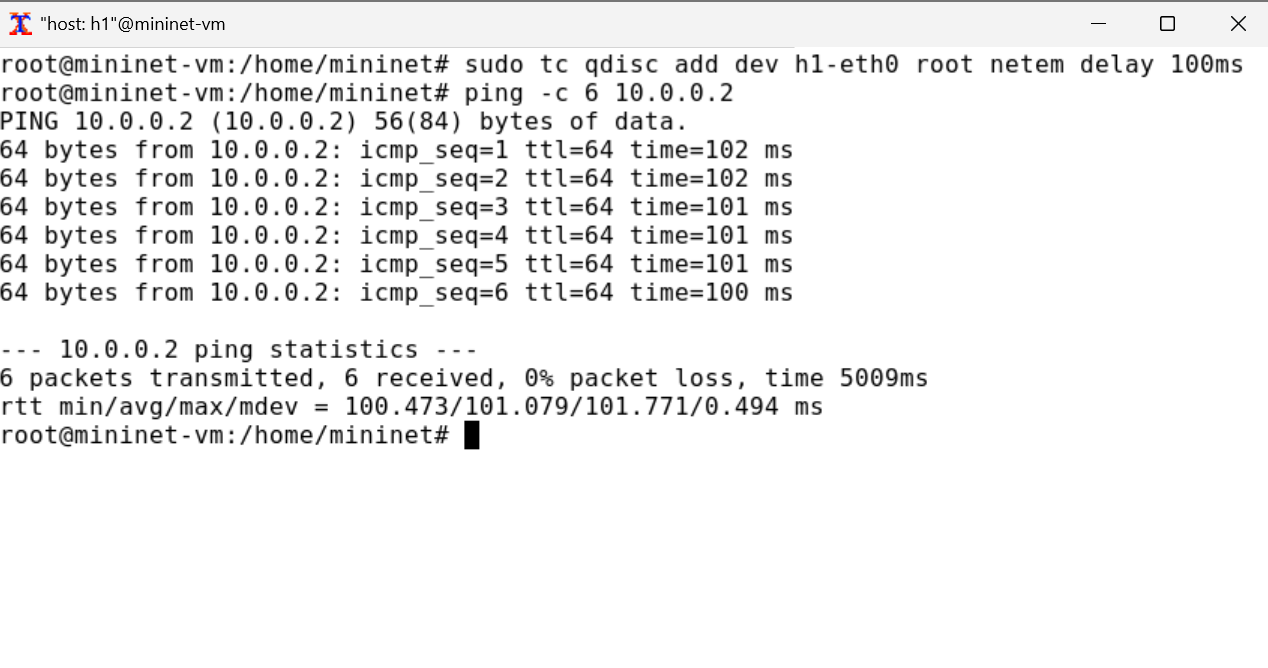


Рис. 5: Добавление задержки в 100мс

Для эмуляции глобальной сети с двунаправленной задержкой необходимо к соответствующему интерфейсу на хосте h2 также добавим задержку в 100 миллисекунд (рис. 6).

Проверим, что соединение между хостом h1 и хостом h2 имеет RTT в 200 мс (100 мс от хоста h1 к хосту h2 и 100 мс от хоста h2 к хосту h1), повторив команду ping с параметром -c 6 на терминале хоста h1.

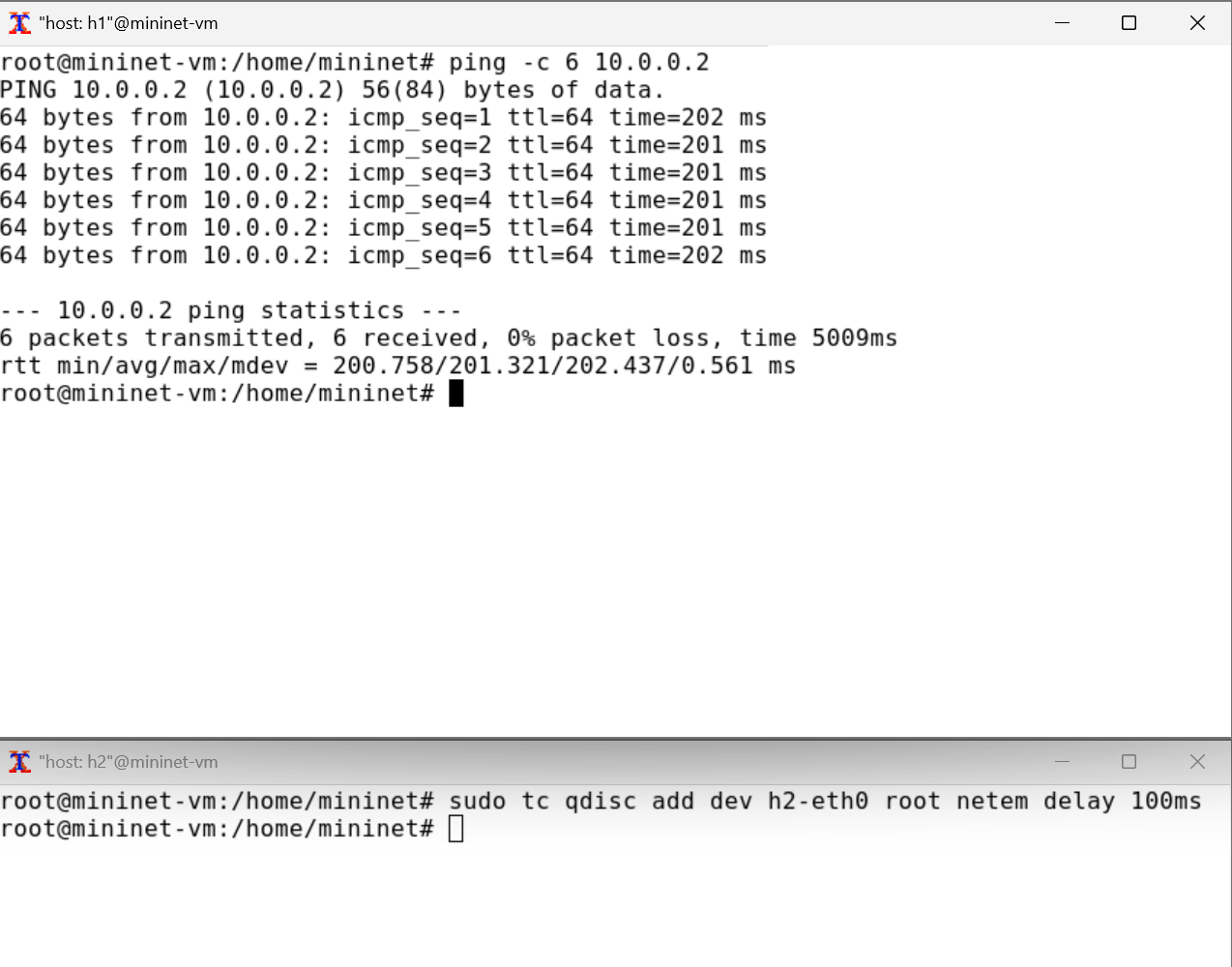


Рис. 6: Двунаправленная задержка соединения

Изменение задержки в эмулируемой глобальной сети

Изменим задержку со 100 мс до 50 мс для отправителя h1 и для получателя h2 (рис. 7).

Проверим, что соединение от хоста h1 к хосту h2 имеет задержку 100 мс, используя команду ping с параметром -c 6 с терминала хоста h1.

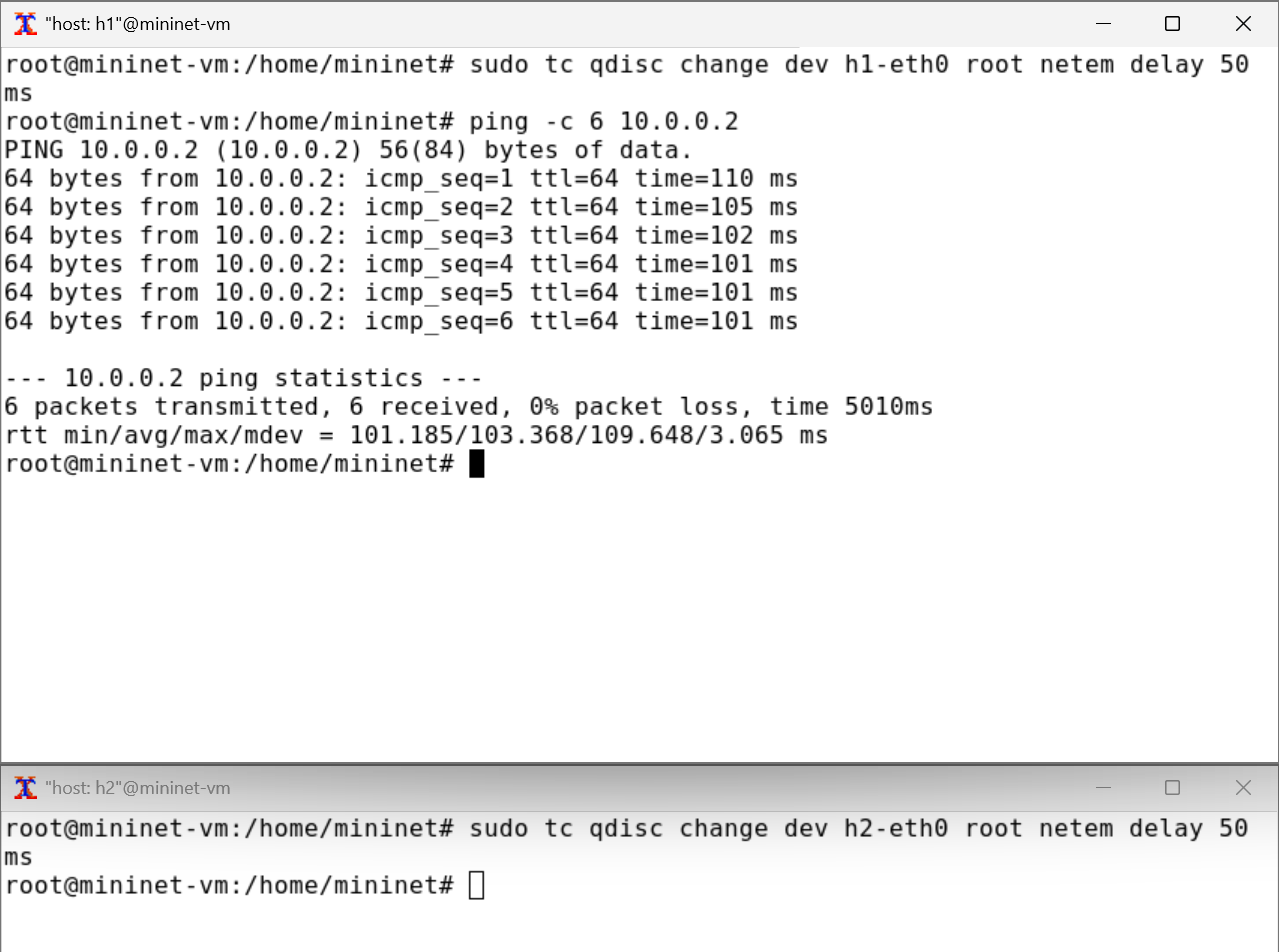


Рис. 7: Изменение задержки на 50мс

Восстановление исходных значений (удаление правил) задержки в эмулируемой глобальной сети

Восстановим конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса для отправителя h1 и для получателя h2. Проверим, что соединение между хостом h1 и хостом h2 не имеет явно установленной задержки, используя команду ping с параметром -c 6 с терминала хоста h1 (рис. 8).

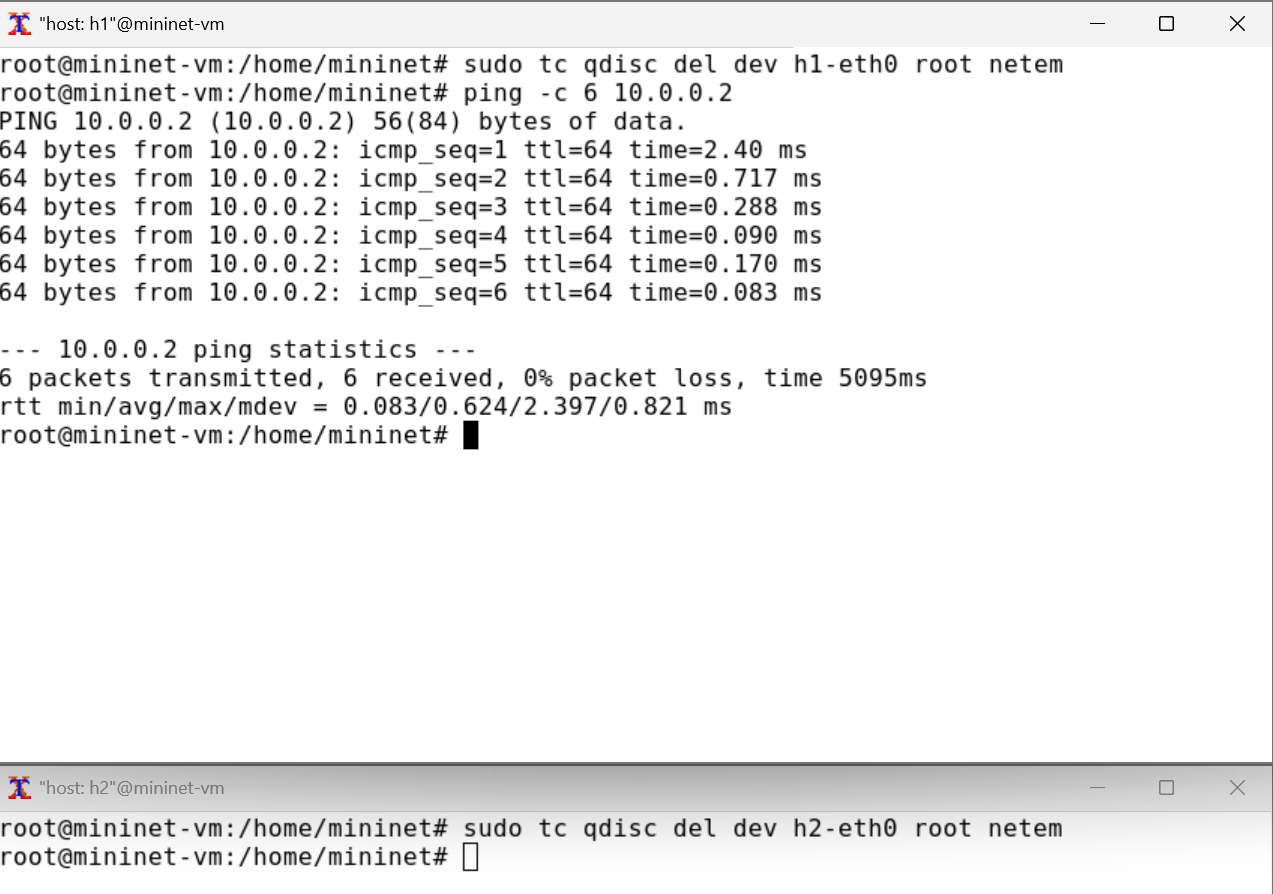


Рис. 8: Восстановление исходных значений задержки

Добавление значения дрожания задержки в интерфейс подключения к эмулируемой глобальной сети

Добавим на узле h1 задержку в 100 мс со случайным отклонением 10 мс. Проверим, что соединение от хоста h1 к хосту h2 имеет задержку 100 мс со случайным отклонением ±10 мс, используя в терминале хоста h1 команду ping с параметром -c 6. Восстановим конфигурацию интерфейса по умолчанию на узле h1 (рис. 9).



Рис. 9: Добавление значения дрожания задержки в интерфейс подключения

Добавление значения корреляции для джиттера и задержки в интерфейс подключения к эмулируемой глобальной сети

Добавим на интерфейсе хоста h1 задержку в 100 мс с вариацией ±10 мс и значением корреляции в 25%. Убедимся, что все пакеты, покидающие устройство h1 на интерфейсе h1-eth0, будут иметь время задержки 100 мс со случайным отклонением ±10 мс, при этом время передачи следующего пакета зависит от предыдущего значения на 25%. Используем для этого в терминале хоста h1 команду ping с параметром -c 20. Восстановим конфигурацию интерфейса по умолчанию на узле h1 (рис. 10).

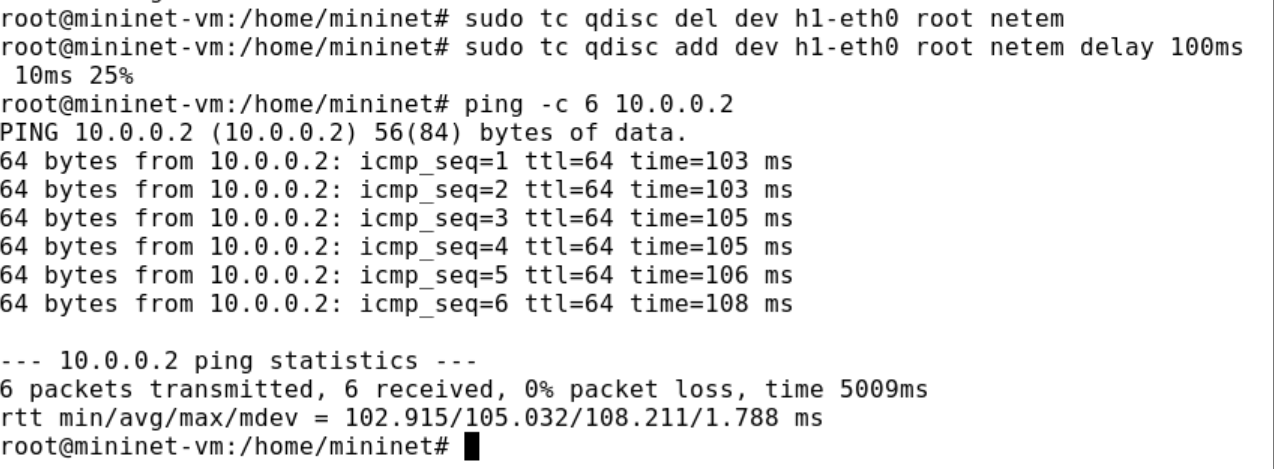


Рис. 10: Добавление значения корреляции для джиттера и задержки в интерфейс подключения

Распределение задержки в интерфейсе подключения к эмулируемой глобальной сети

Зададим нормальное распределение задержки на узле h1 в эмулируемой сети. Убедимся, что все пакеты, покидающие хост h1 на интерфейсе h1-eth0, будут иметь время задержки, которое распределено в диапазоне 100 мс ±20 мс. Используем для этого команду ping на терминале хоста h1 с параметром -c 10. Восстановим конфигурацию интерфейса по умолчанию на узле h1. Завершим работу mininet в интерактивном режиме (рис. 11).

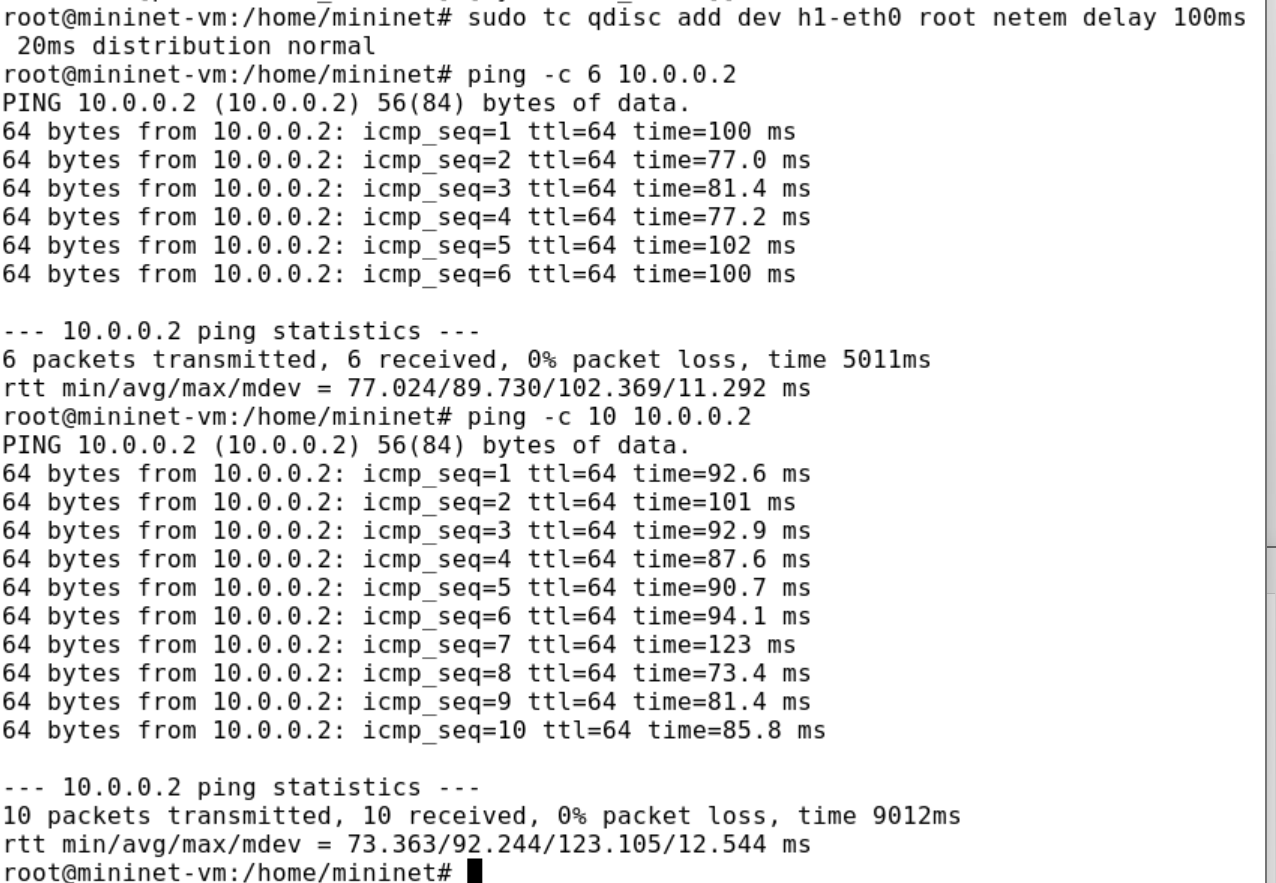


Рис. 11: Распределение задержки в интерфейсе подключения

Обновим репозитории программного обеспечения на виртуальной машине. Установим пакет geeqie для просмотра файлов png. Для каждого воспроизводимого эксперимента expname создадим свой каталог, в котором будут размещаться файлы эксперимента (рис. 12).

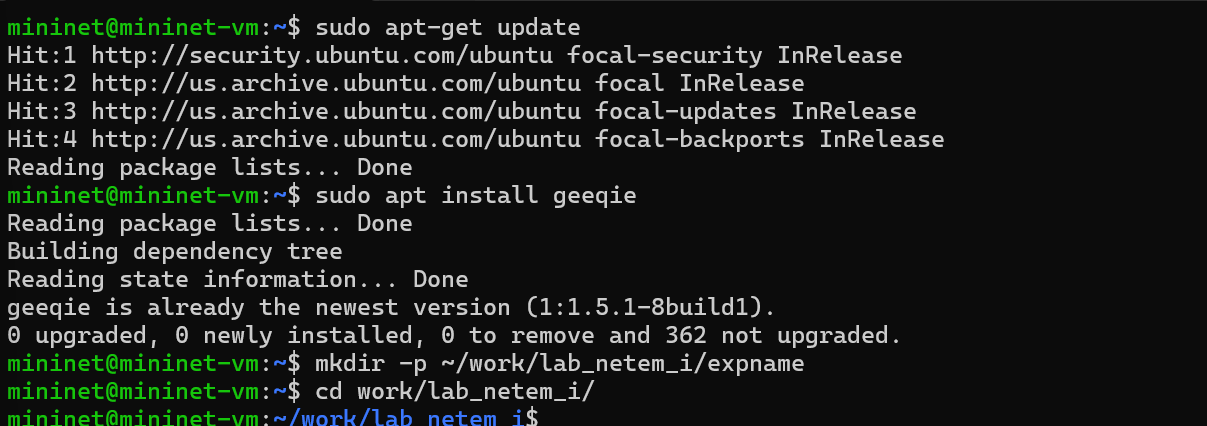


Рис. 12: Подготовка к производимому эксперементу

Создадим скрипт для эксперимента lab\_netem\_i.py (рис. 13).

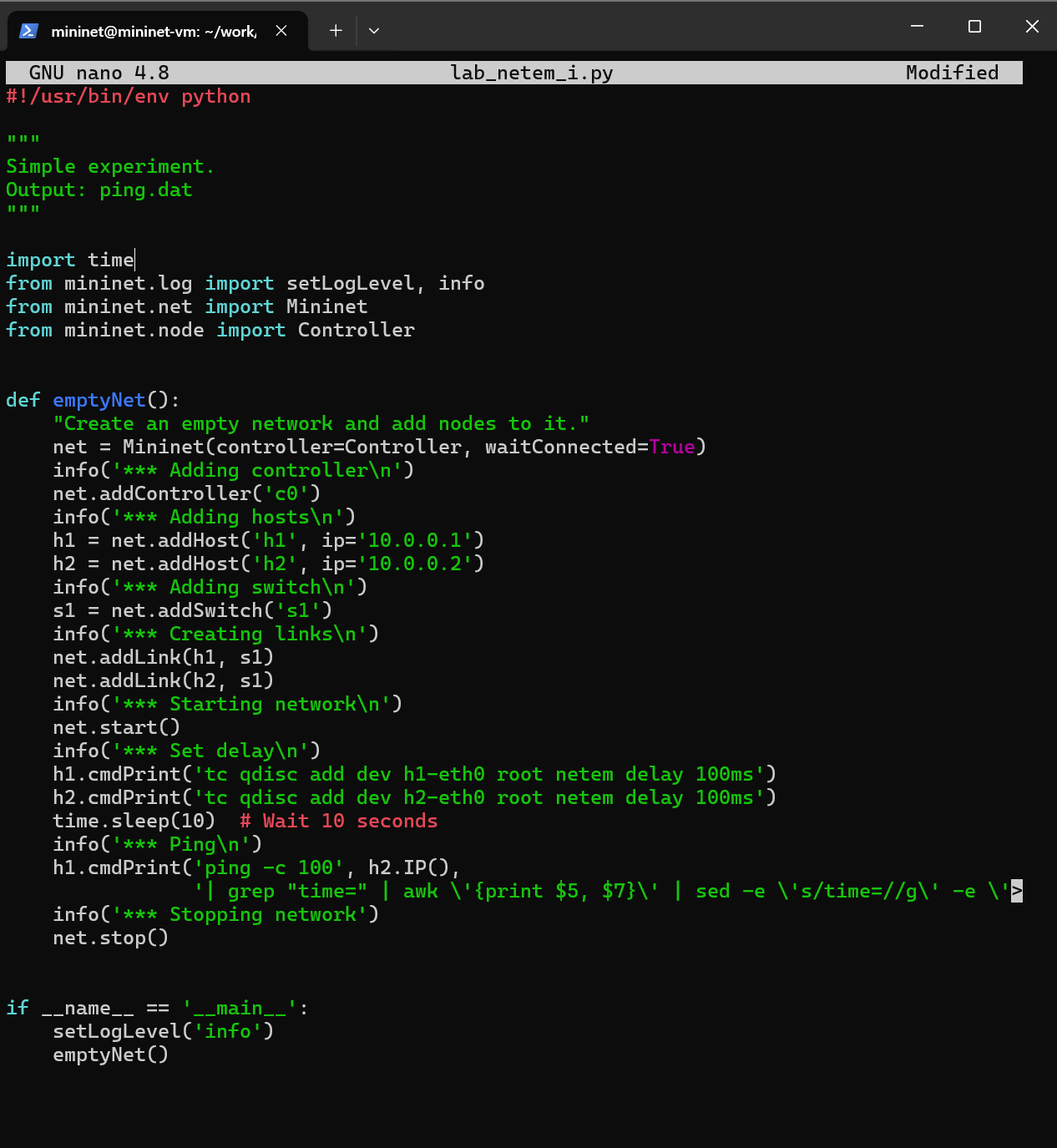


Рис. 13: Листинг lab\_netem\_i.py

Затем создадим скрипт для визуализации ping\_plot результатов эксперимента(рис. 14).

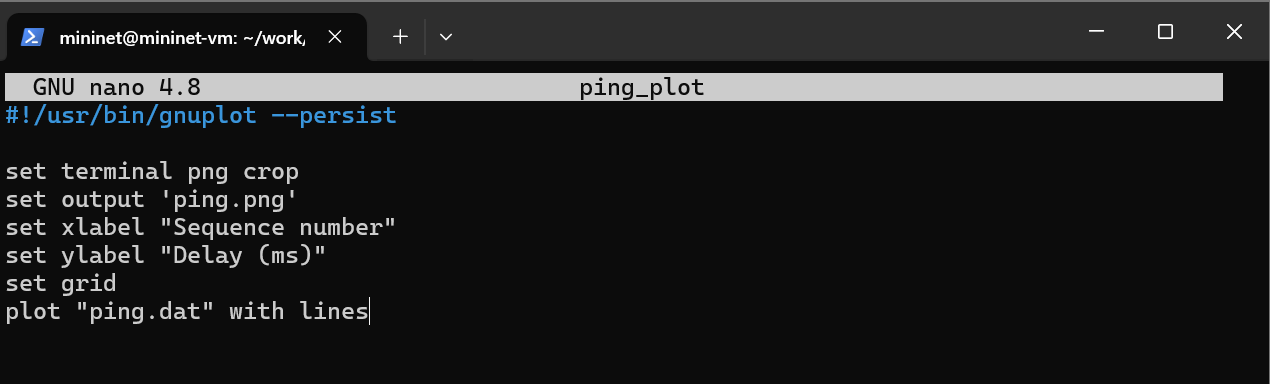


Рис. 14: Листинг ping\_plot

Зададим права доступа к файлу скрипта(рис. 15).

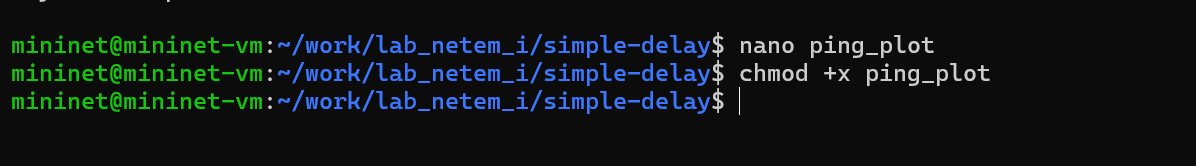


Рис. 15: Выдача прав выполнения дл ping\_plot

Создадим файла Makefile. Внутри файла Makefile поместим скрипт для управления процессом проведения эксперимента.(рис. 16).



Рис. 16: Листинг Makefile

Выполним эксперимент(рис. 17).

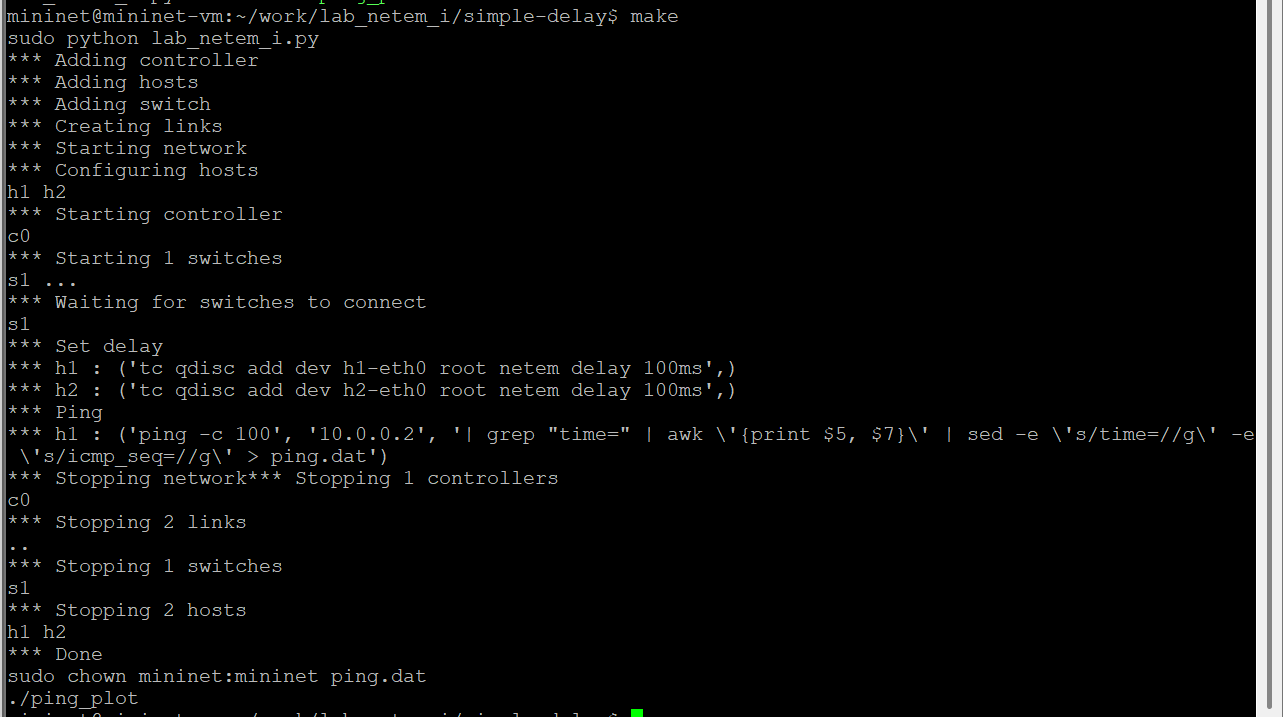


Рис. 17: Запуск эксперемента

Просмотрим построенный в результате выполнения скриптов график(рис. 18).

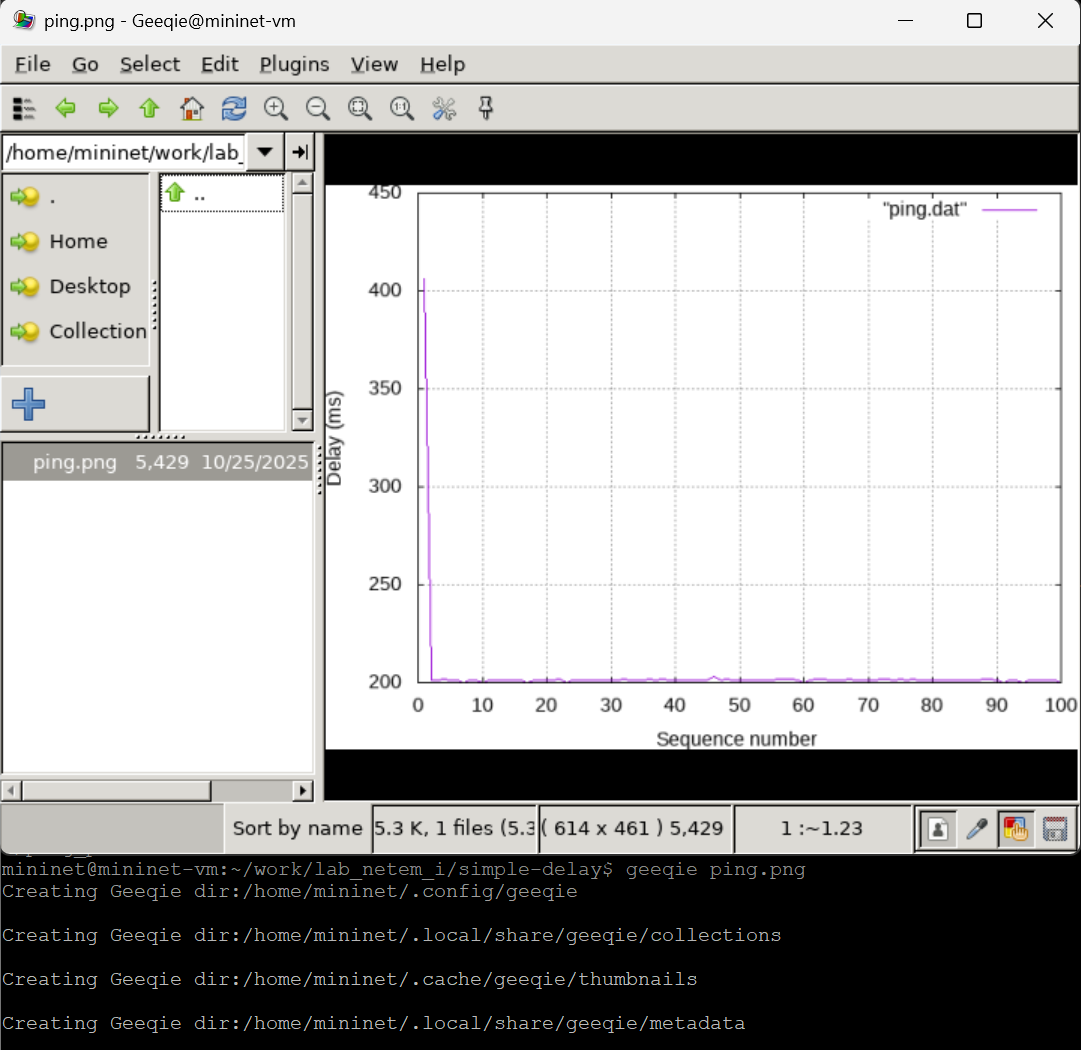


Рис. 18: Просмотр графика

Из файла ping.dat удалим первую строку и заново построим график(рис. 19).

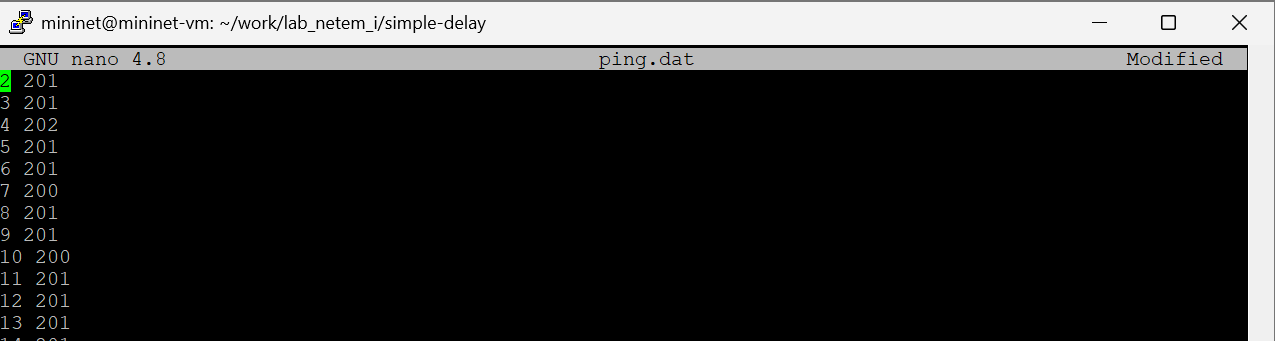


Рис. 19: Удаление первой строчки из файла ping.dat

Просмотрим заново построенный график (рис. 20).

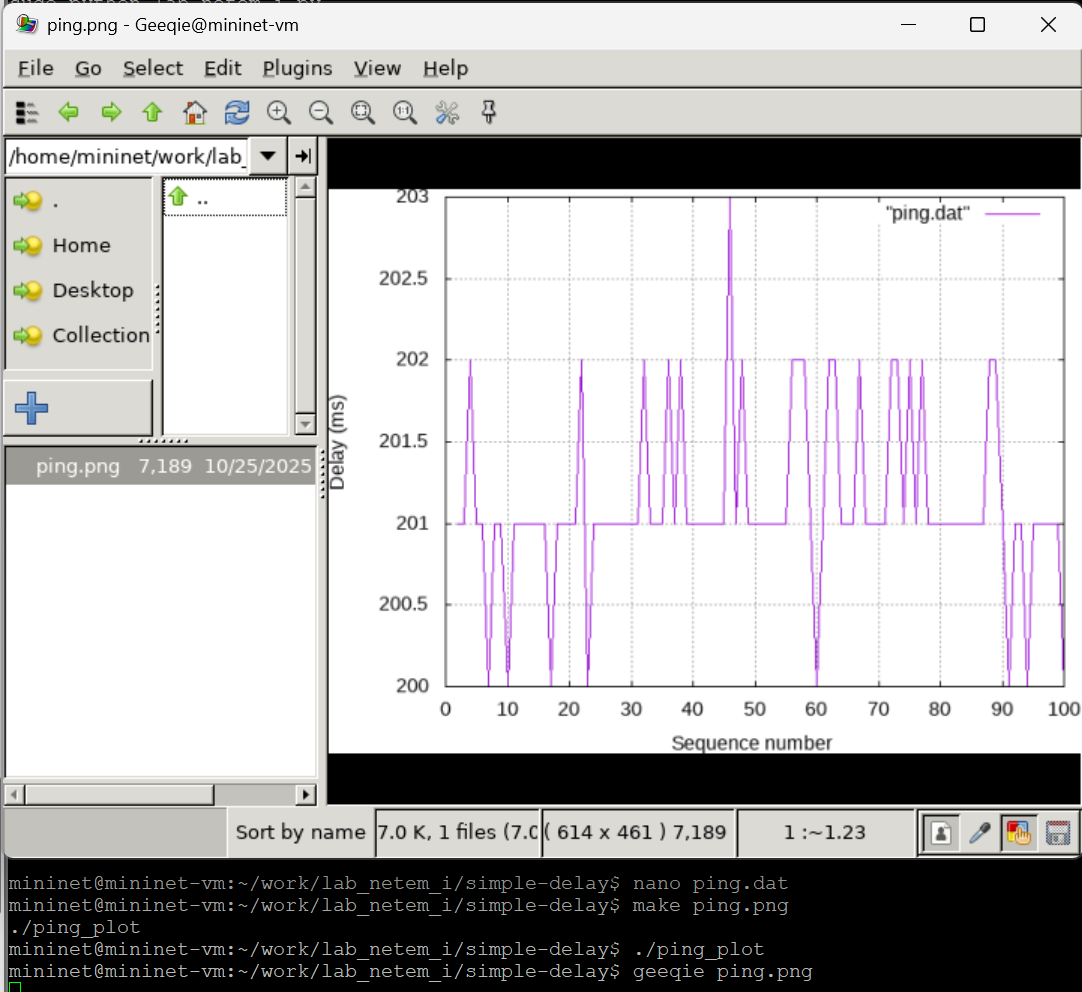


Рис. 20: Просмотр графика

Разработаем скрипт для вычисления на основе данных файла ping.dat минимального, среднего, максимального и стандартного отклонения времени приёма-передачи. Протестируем его Также добавим правило запуска скрипта в Makefile (рис. 21; 22; 23; ).

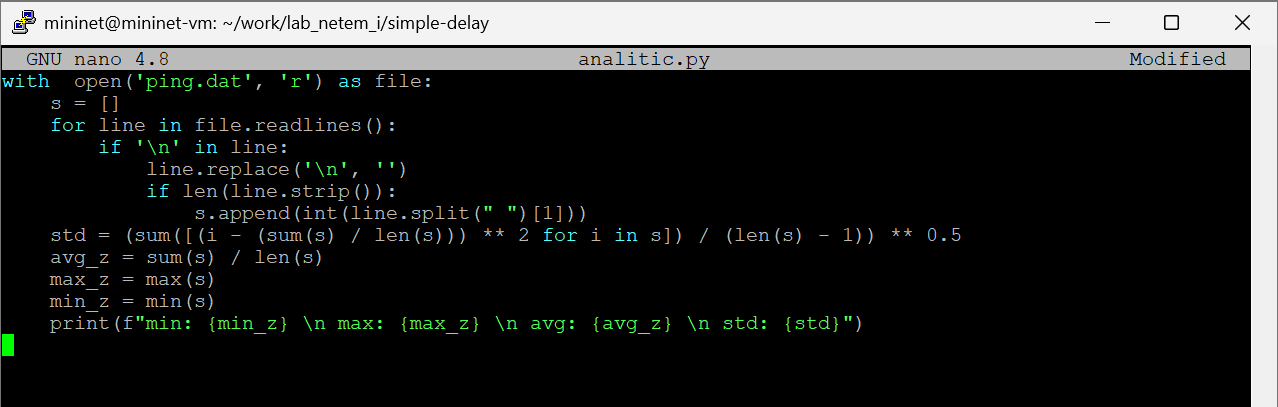


Рис. 21: Разработка скрипта для вычисления на основе данных файла ping.dat статистических данных

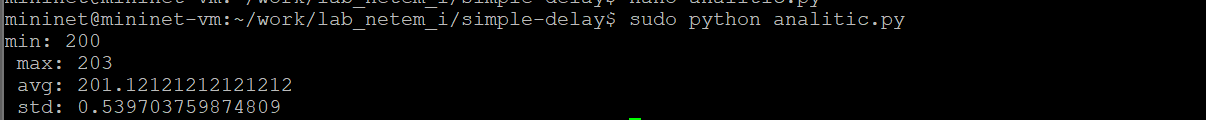


Рис. 22: Тестирование разработанного скрипта



Рис. 23: Добавление правила запуска скрипта в Makefile

Очистим каталог от результатов проведения экспериментов.(рис. 24).

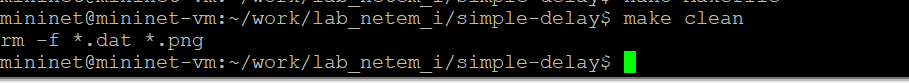


Рис. 24: Очистка результатов эксперемента

Самостоятельно реализуем воспроизводимые эксперименты по изменению задержки, джиттера, значения корреляции для джиттера и задержки, распределения времени задержки в эмулируемой глобальной сети. Построим графики. Вычислим минимальное, среднее, максимальное и стандартное отклонение времени приёма-передачи для каждого случая (рис. 25 - рис. 36):

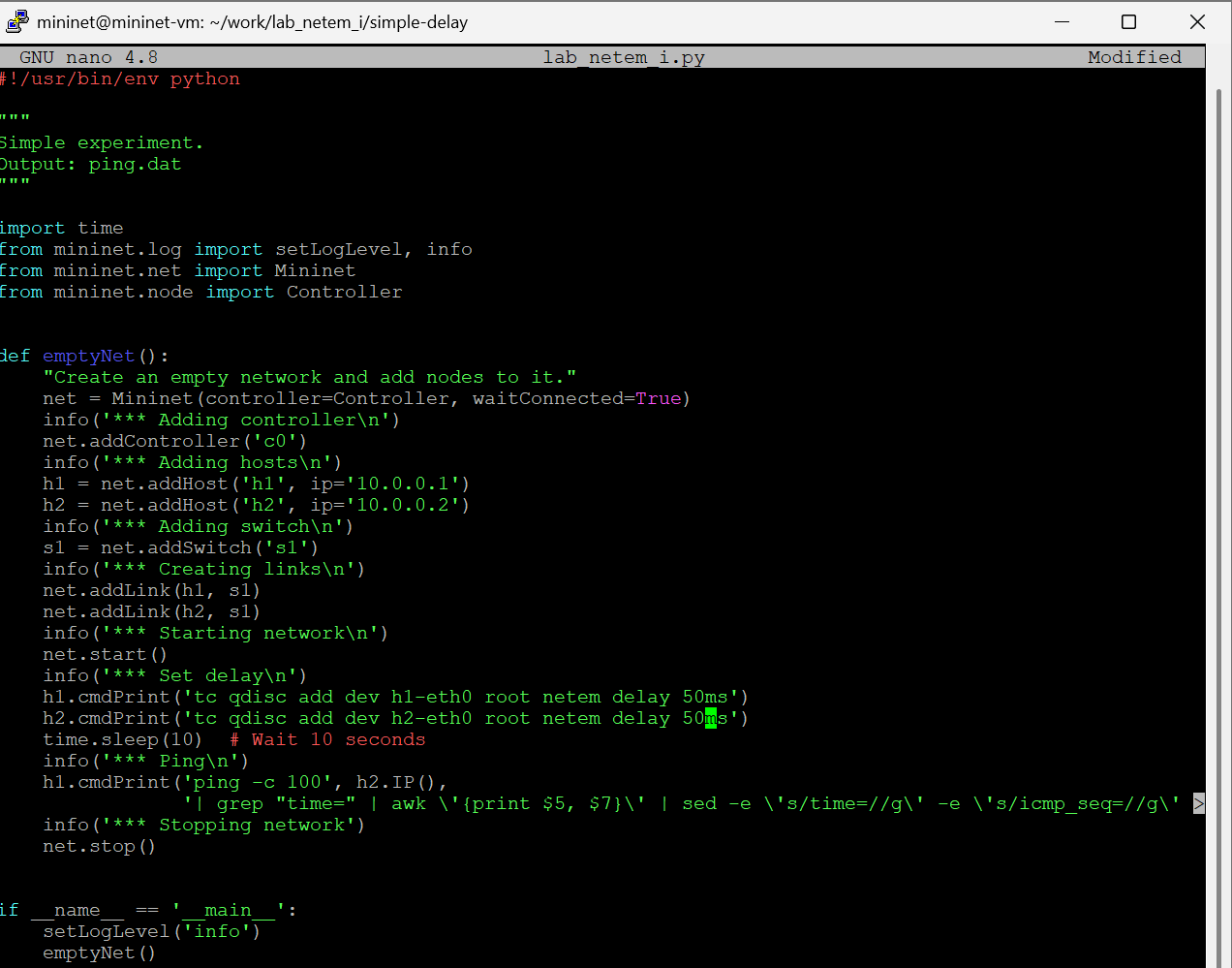


Рис. 25: ЛИСТИНГ|Воспроизводимый эксперимент по изменению задержки

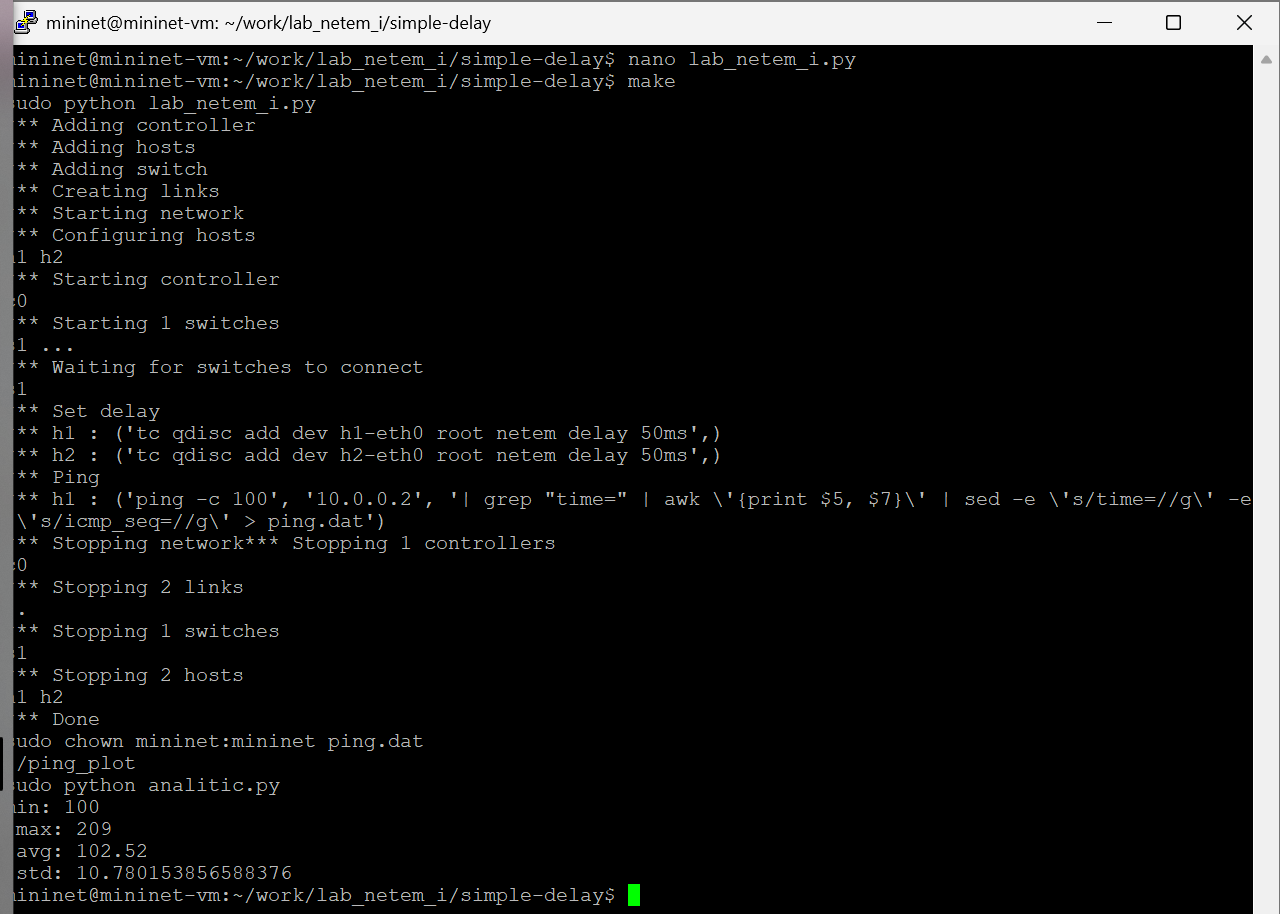


Рис. 26: ВЫВОД|Воспроизводимый эксперимент по изменению задержки

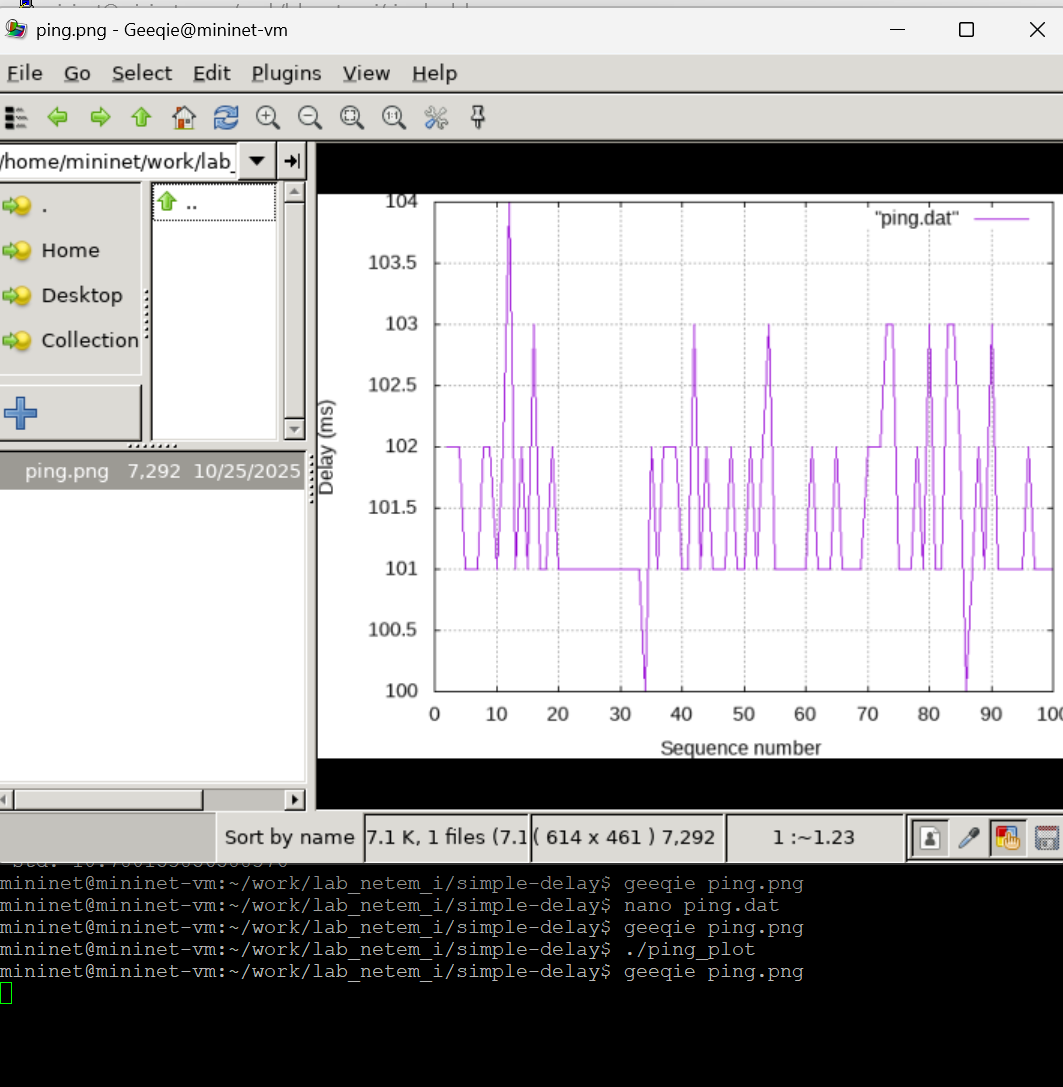


Рис. 27: ГРАФИК|Воспроизводимый эксперимент по изменению задержки

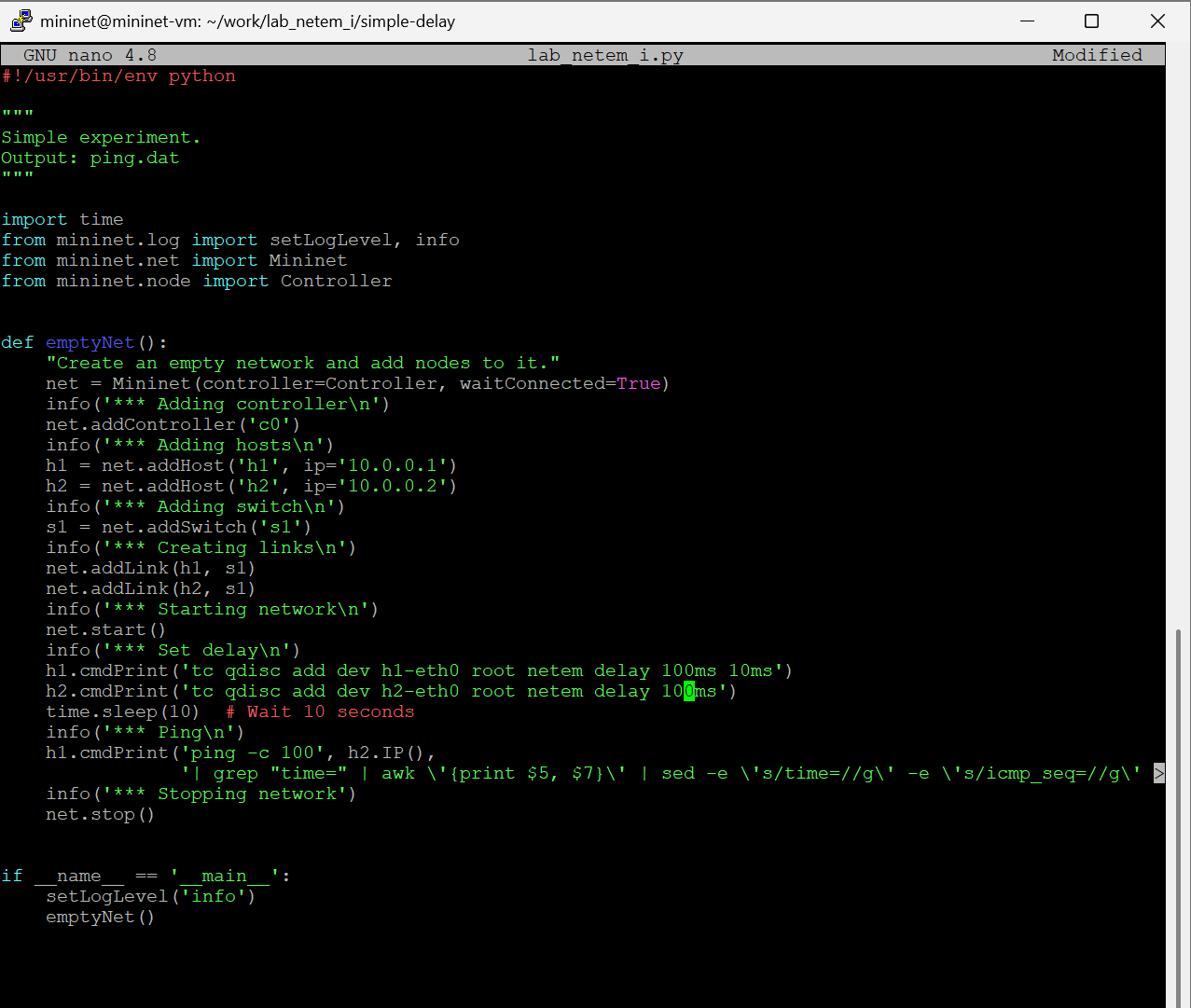


Рис. 28: ЛИСТИНГ|Воспроизводимый эксперимент по изменению джиттера

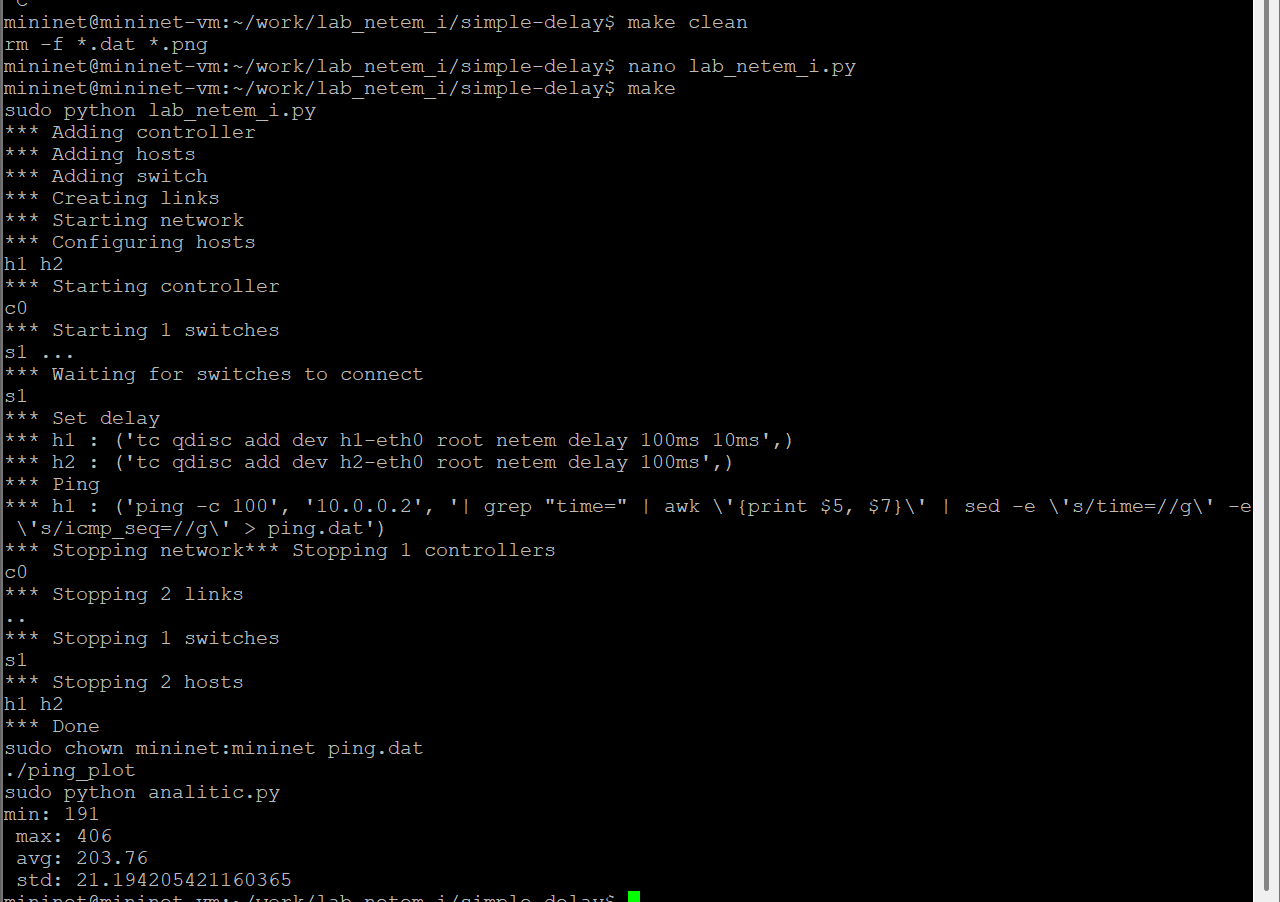


Рис. 29: ВЫВОД|Воспроизводимый эксперимент по изменению джиттера

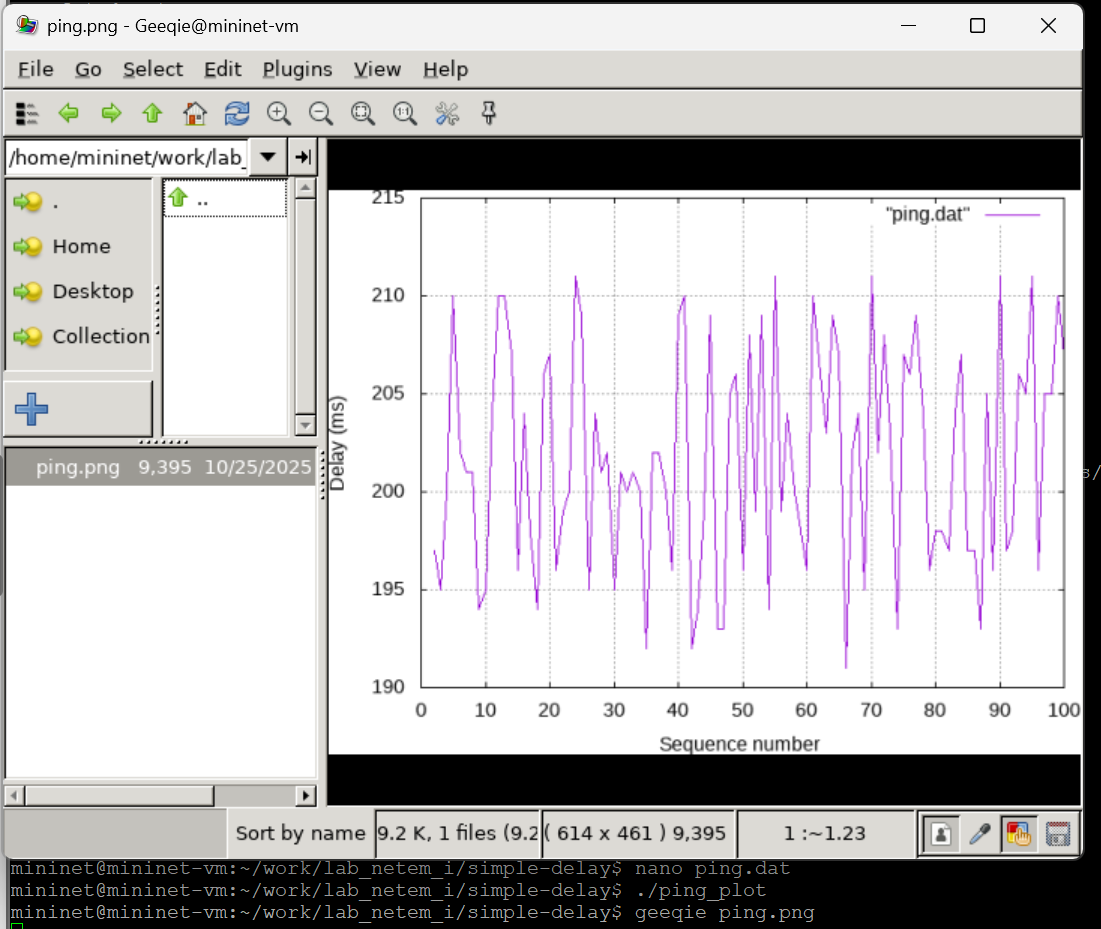


Рис. 30: ГРАФИК|Воспроизводимый эксперимент по изменению джиттера



Рис. 31: ЛИСТИНГ|Воспроизводимый эксперимент по изменению значения корреляции для джиттера и задержки

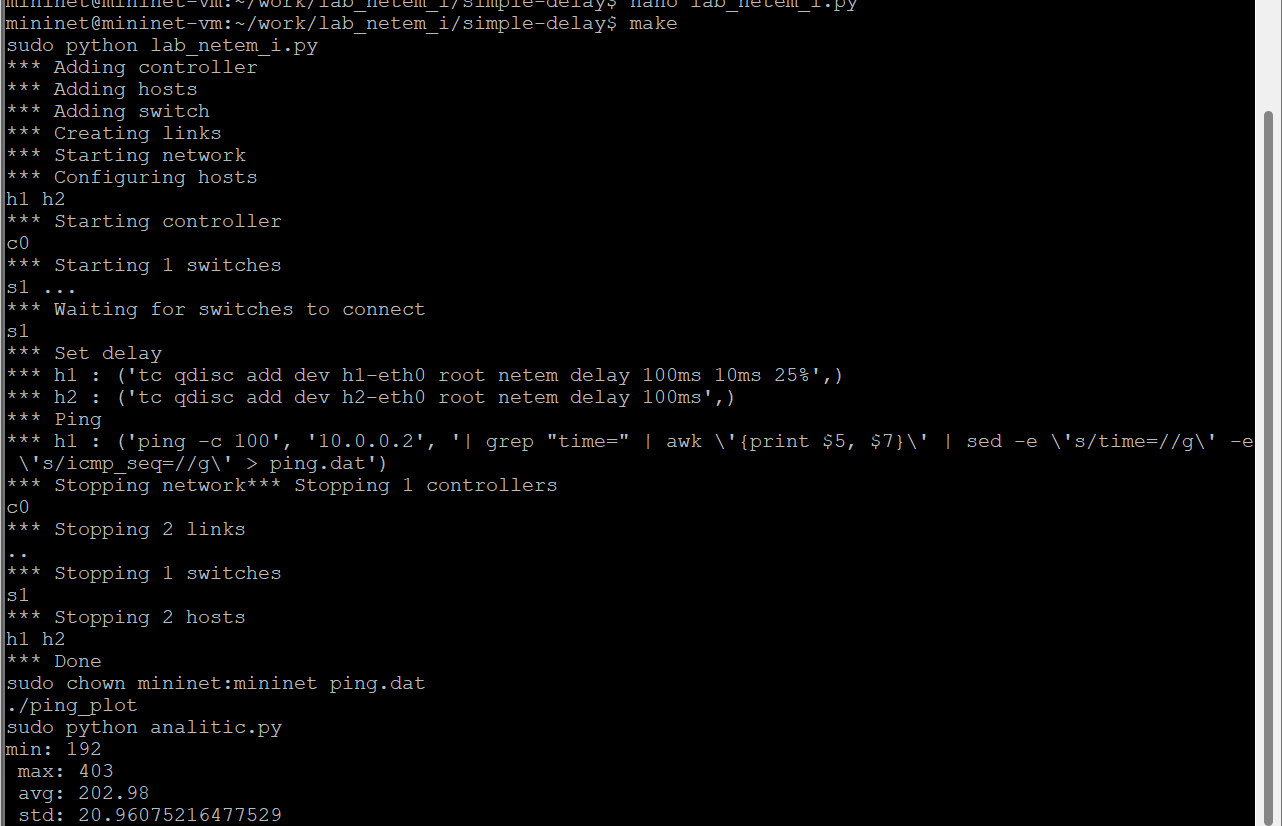


Рис. 32: ВЫВОД|Воспроизводимый эксперимент по изменению значения корреляции для джиттера и задержки



Рис. 33: ГРАФИК|Воспроизводимый эксперимент по изменению значения корреляции для джиттера и задержки

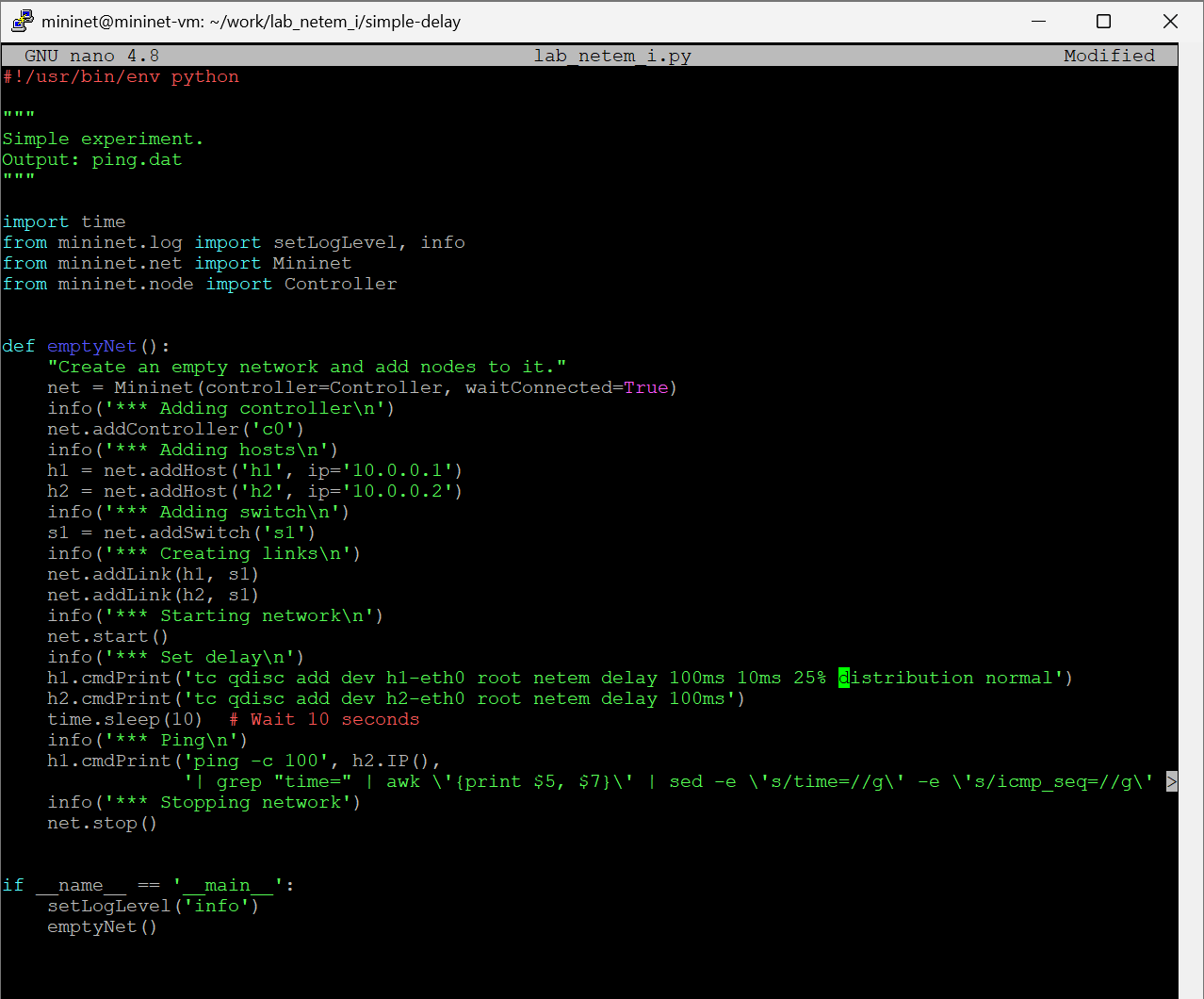


Рис. 34: ЛИСТИНГ|Воспроизводимый эксперимент по изменению распределения времени задержки в эмулируемой глобальной сети

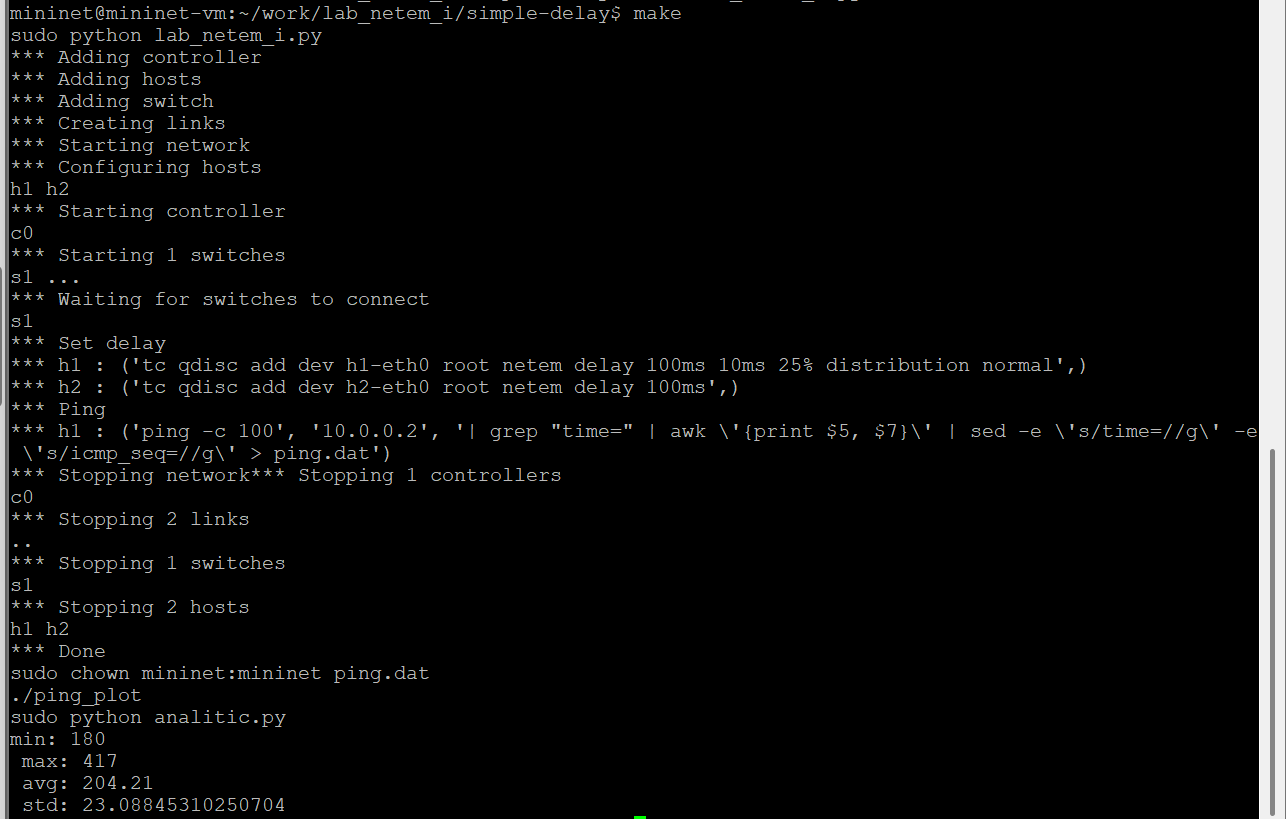


Рис. 35: ВЫВОД|Воспроизводимый эксперимент по изменению распределения времени задержки в эмулируемой глобальной сети

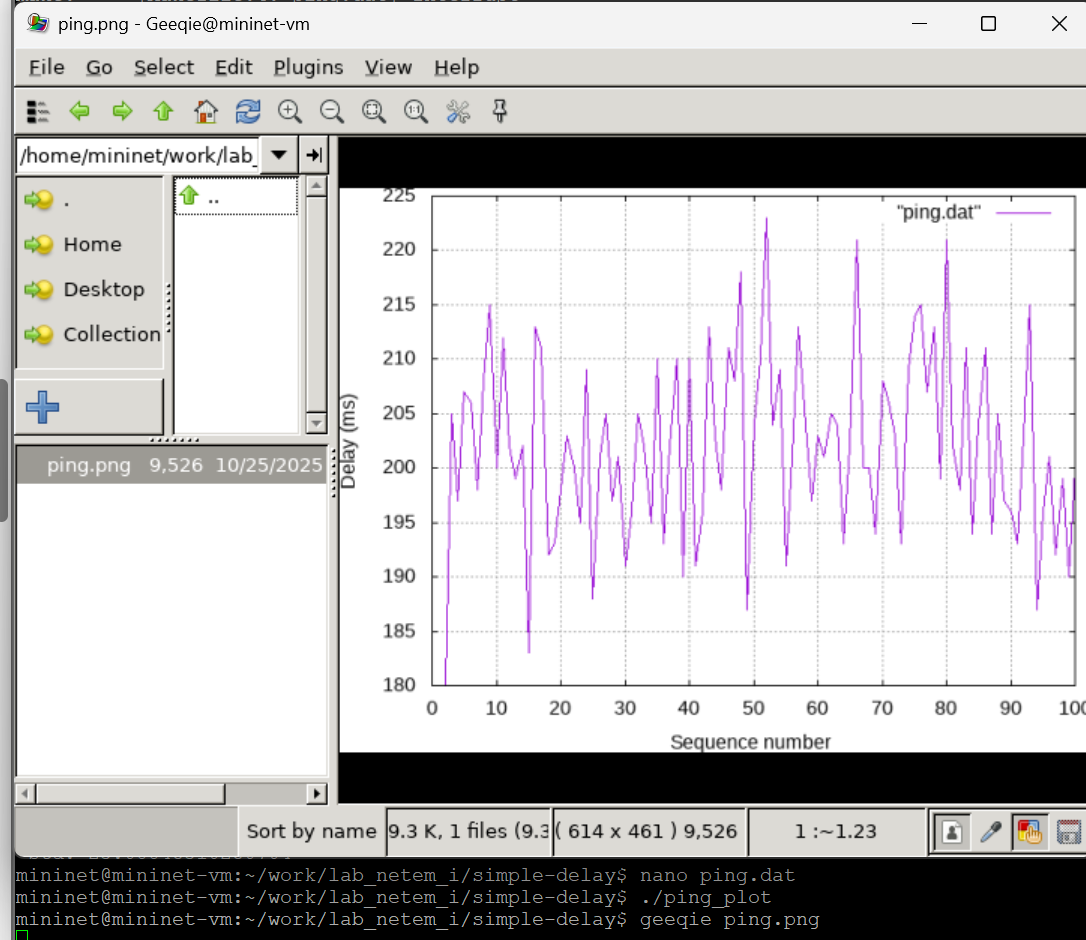


Рис. 36: ГРАФИК|Воспроизводимый эксперимент по изменению распределения времени задержки в эмулируемой глобальной сети

Так как все эти эксперементы желательно по заданию распределить по папкам, то сделаем копию где мы делали эксперемент и перенесём известные нам уже программы.(рис. 37).

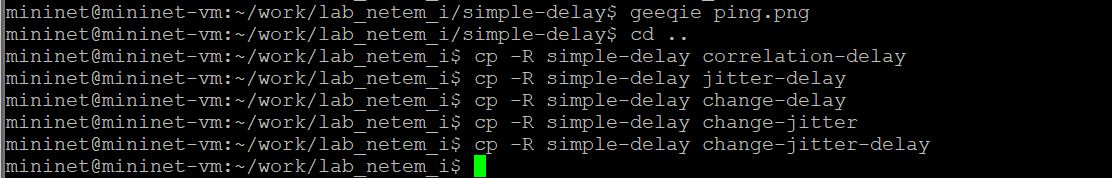


Рис. 37: Создание папок под эксперементы

# 5 Выводы

В результате выполнения данной лабораторной работы я познакомилcя с NETEM – инструментом для тестирования производительности приложений в виртуальной сети, а также получил навыки проведения интерактивного и воспроизводимого экспериментов по измерению задержки и её дрожания (jitter) в моделируемой сети в среде Mininet.

# Список литературы

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