Отчёт по лабораторной работе №4

Дисциплина: Моделирование сетей передачи данных

Боровиков Даниил Александрович НПИбд-01-22

Содержание

# 1 Цель работы

Основной целью работы является знакомство с NETEM [1]. — 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 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.

# 3 Выполнение лабораторной работы

В виртуальной машине mininet исправим права запуска X-соединения (рис. 1):

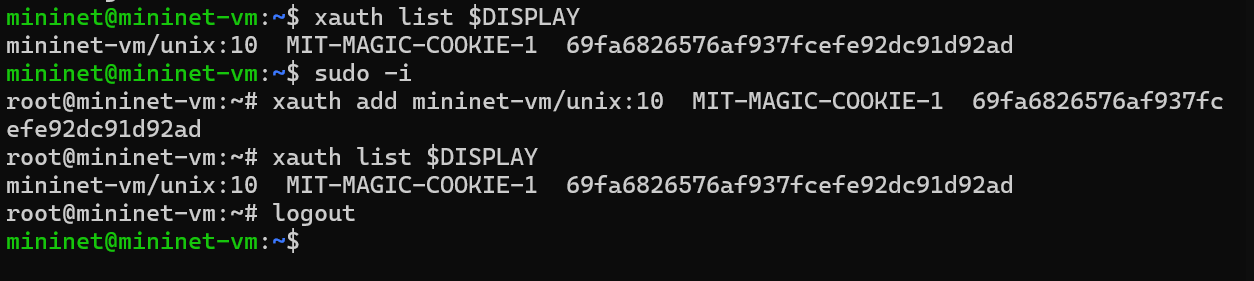


Рис. 1: Исправление прав запуска X-соединения в виртуальной машине mininet

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

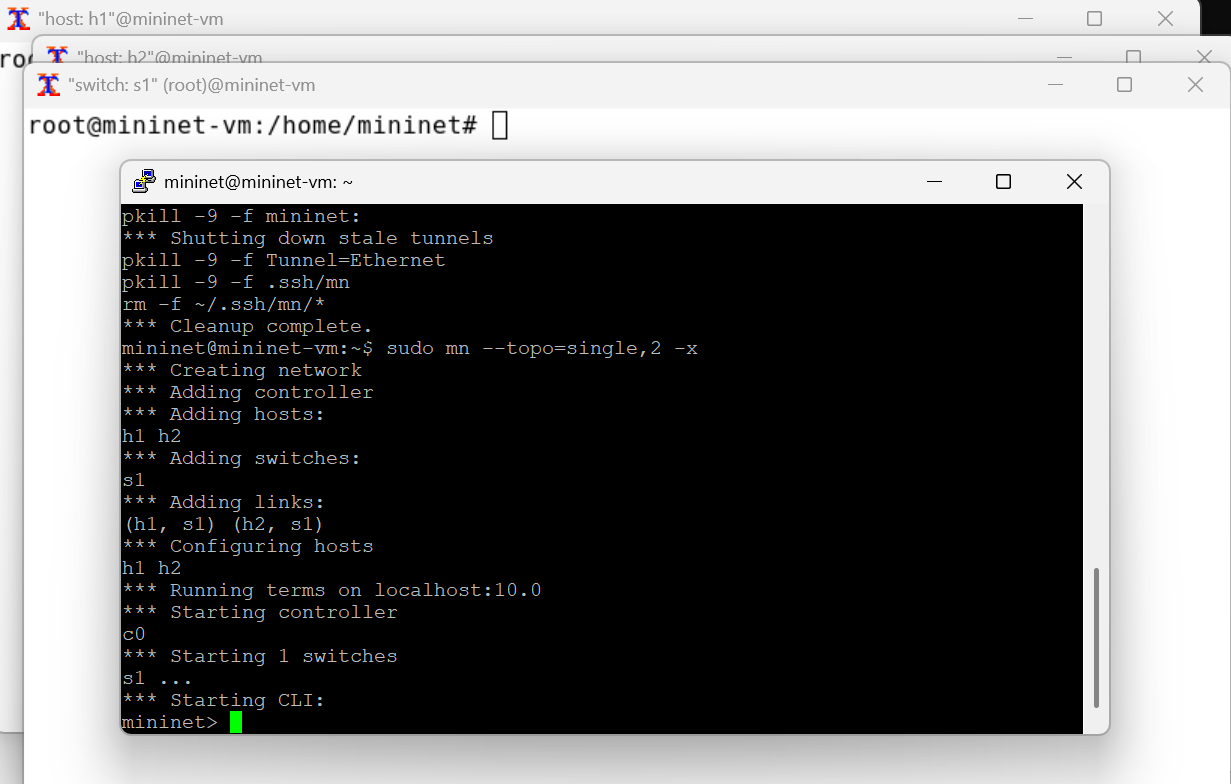


Рис. 2: Создание простейшей топологии

На хостах h1 и h2 введём 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):

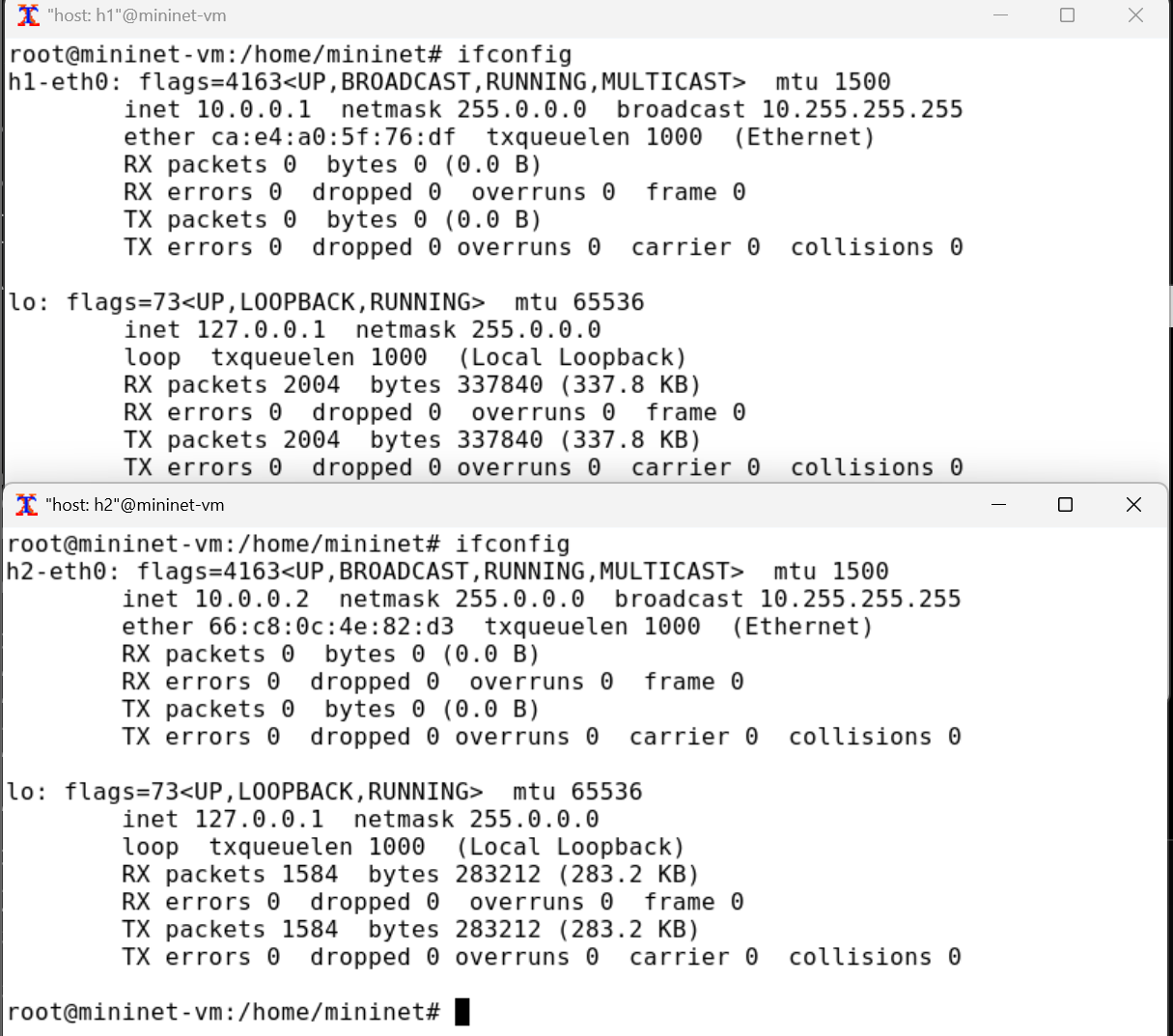


Рис. 3: Отображение информации их сетевых интерфейсов и IP-адресов

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

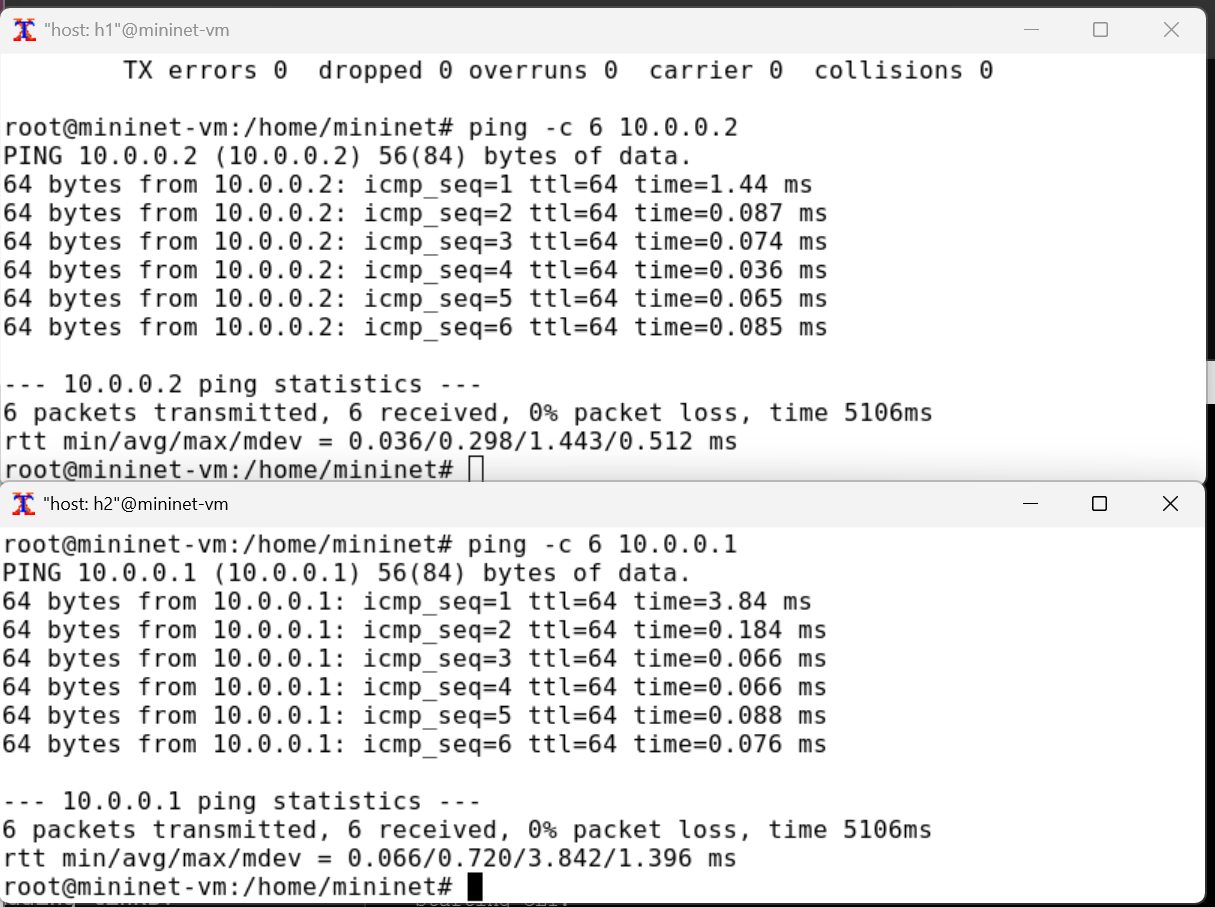


Рис. 4: Проверка подключения между хостами h1 и h2

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

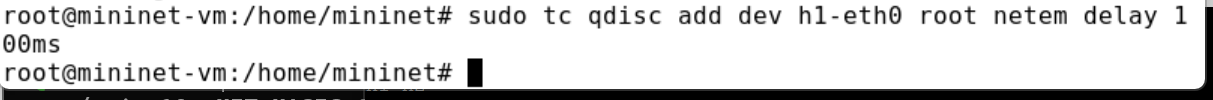


Рис. 5: Добавление задержки в 100 мс к выходному интерфейсу на хосте h1

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

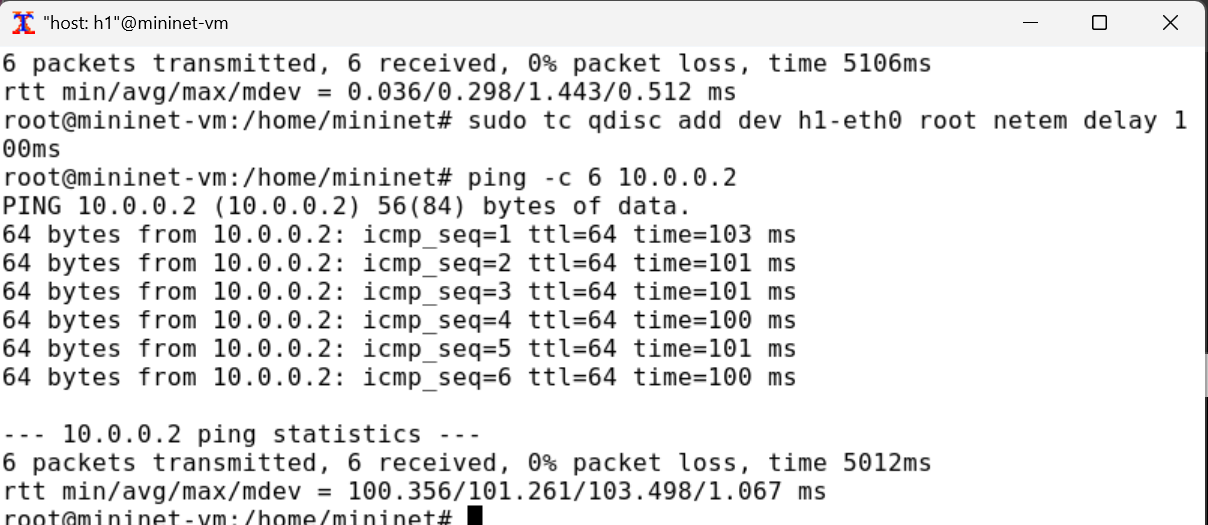


Рис. 6: Проверка

Для эмуляции глобальной сети с двунаправленной задержкой необходимо к соответствующему интерфейсу на хосте h2 также добавить задержку в 100 миллисекунд (рис. 7):

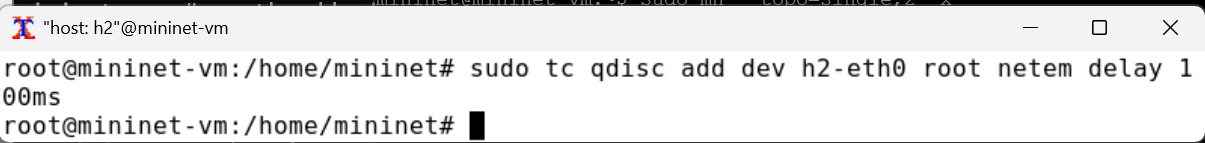


Рис. 7: Добавление задержки в 100 мс к выходному интерфейсу на хосте h2

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. 8):

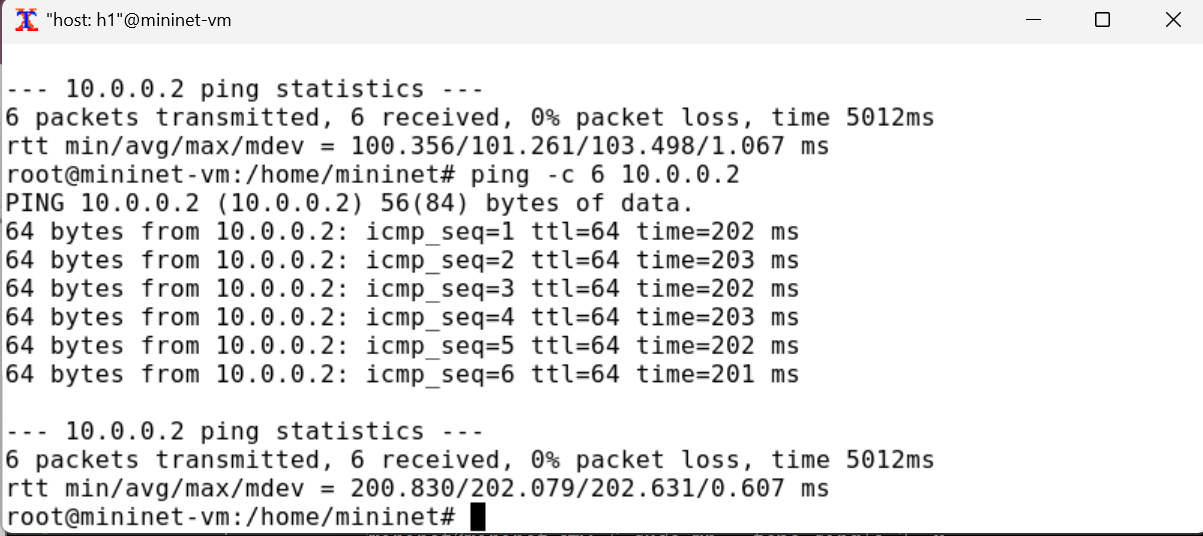


Рис. 8: Проверка

Изменим задержку со 100 мс до 50 мс для отправителя h1 и для получателя h2 (рис. 9):

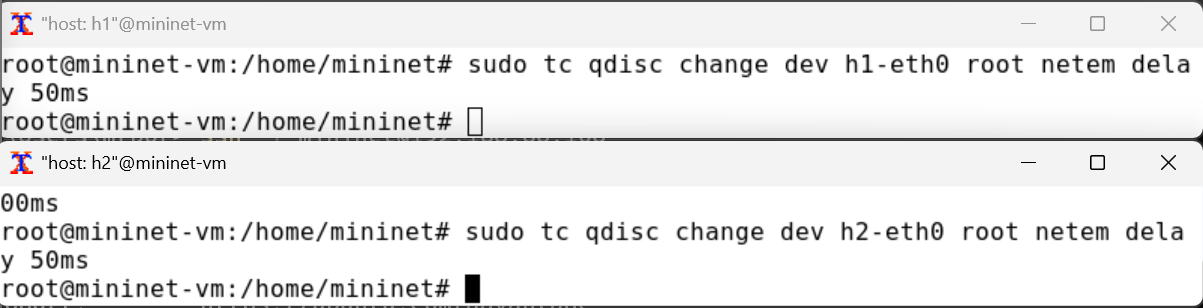


Рис. 9: Изменение задержки со 100 мс до 50 мс

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. 10):

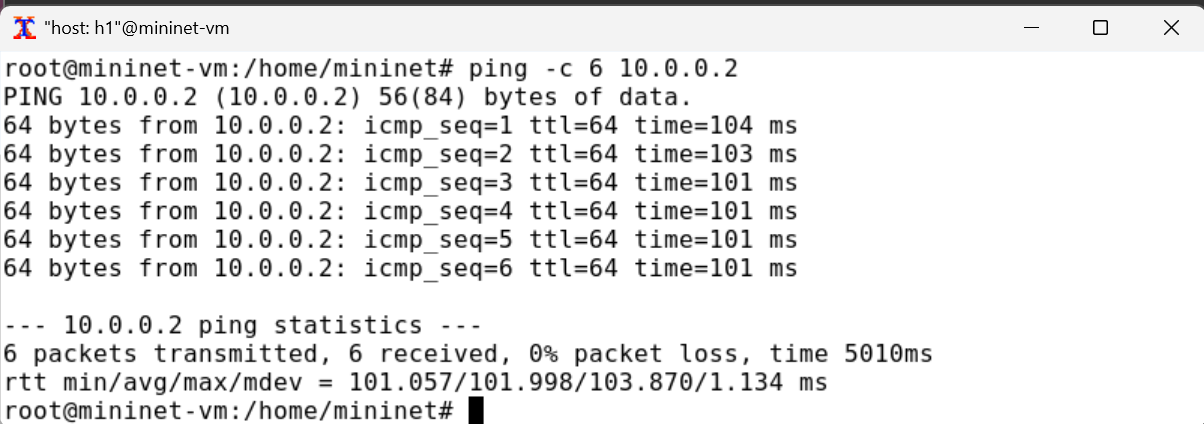


Рис. 10: Проверка

Восстановим конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса (рис. 11):

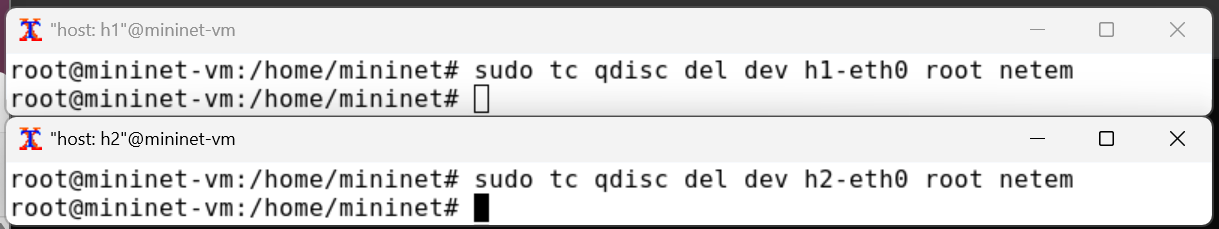


Рис. 11: Восстановление конфигураций по умолчанию

Добавим на узле h1 задержку в 100 мс со случайным отклонением 10 мс (рис. 12):

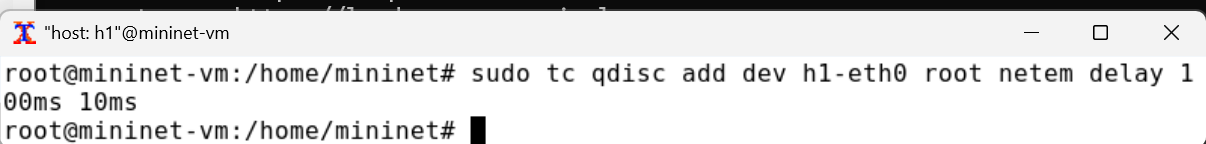


Рис. 12: Добавление на узле h1 задержки в 100 мс со случайным отклонением 10 мс

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 (рис. 13):

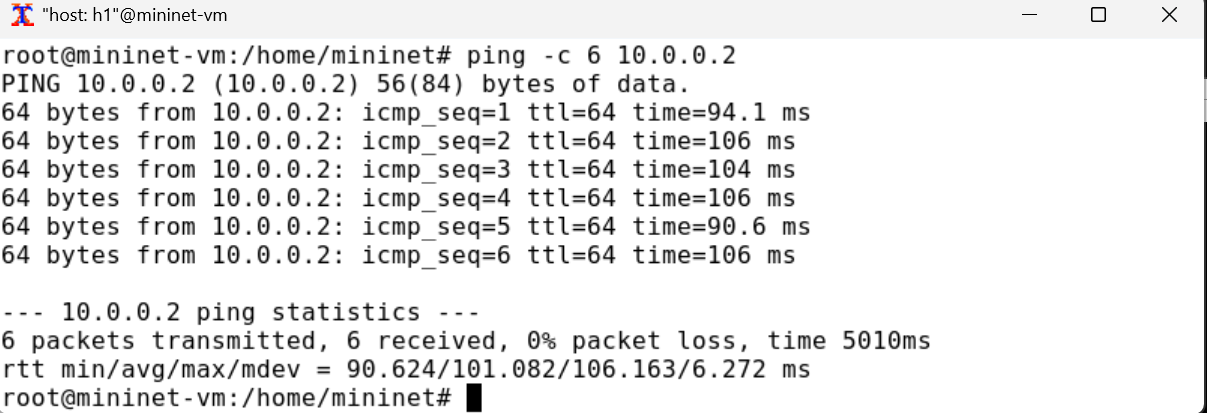


Рис. 13: Проверка

Восстановим конфигурацию интерфейса по умолчанию на узле h1 (рис. 14):

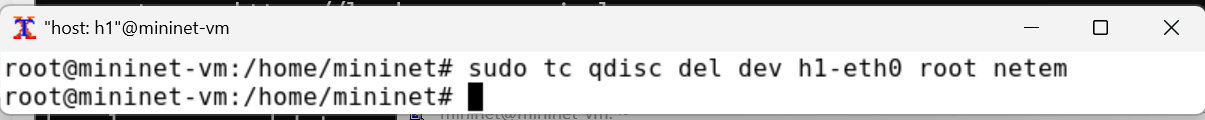


Рис. 14: Восстановление конфигурации интерфейса по умолчанию

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 (рис. 15):

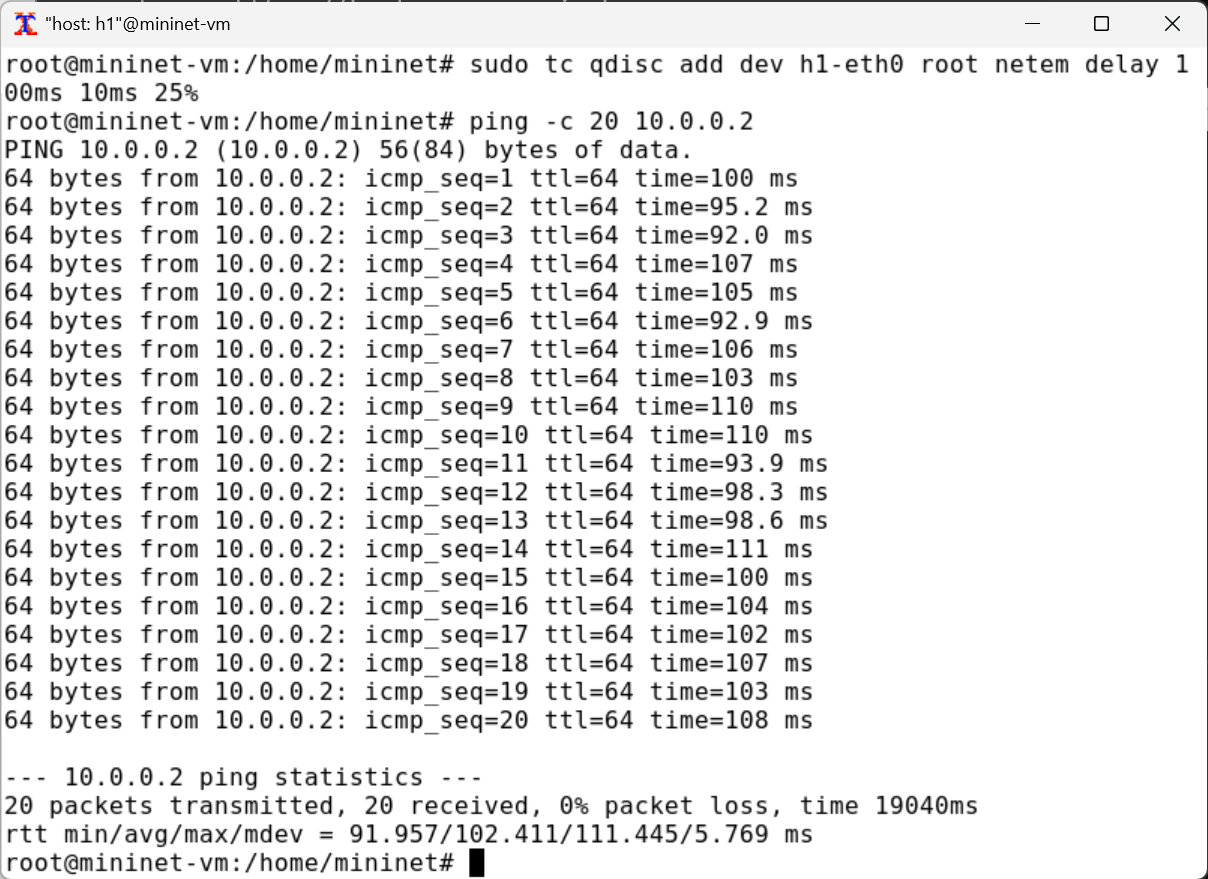


Рис. 15: Проверка

Восстановим конфигурацию интерфейса по умолчанию на узле h1 (рис. 16):



Рис. 16: Восстановление конфигурации интерфейса по умолчанию

Зададим нормальное распределение задержки на узле h1 в эмулируемой сети (рис. 17):

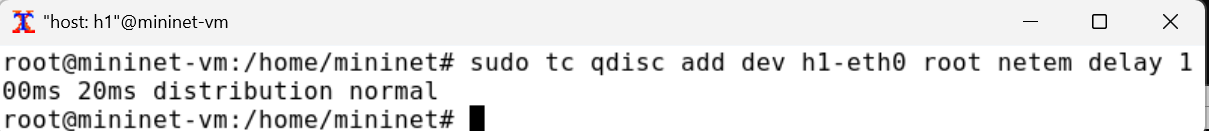


Рис. 17: Настройка нормального распределения задержки на узле h1 в эмулируемой сети

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 (рис. 18):

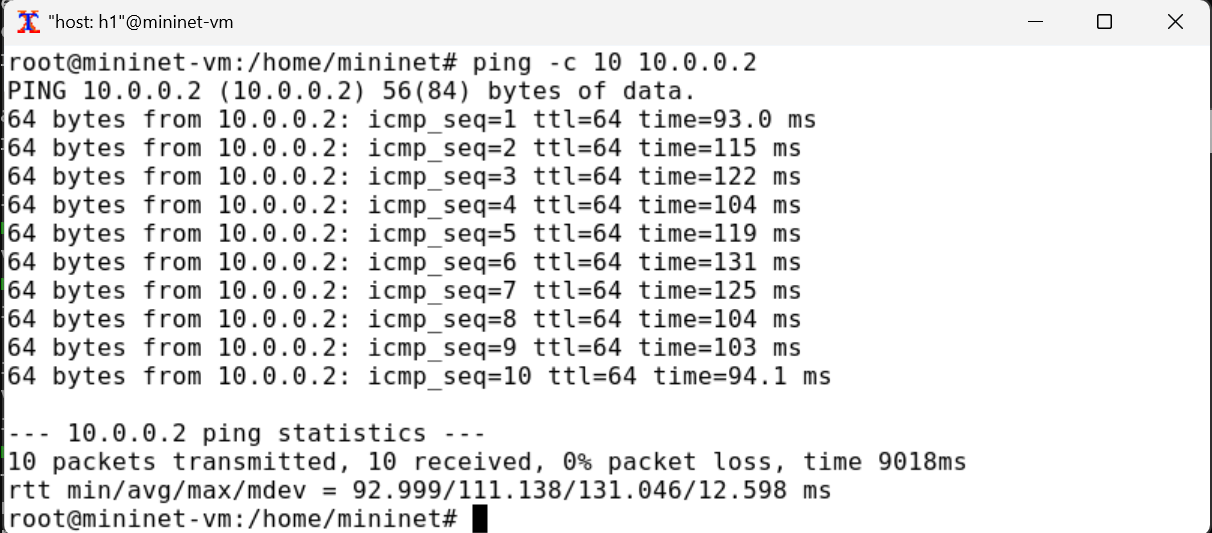


Рис. 18: Проверка

Восстановим конфигурацию интерфейса по умолчанию на узле h1 (рис. 19):

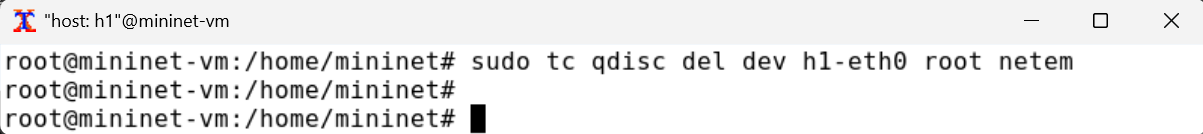


Рис. 19: Восстановление конфигурации интерфейса по умолчанию

Завершим работу mininet в интерактивном режиме (рис. 20):

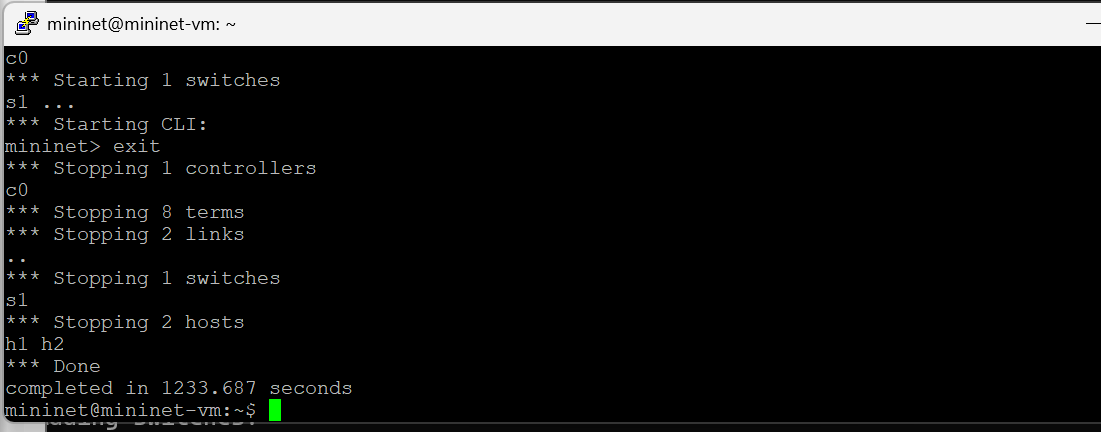


Рис. 20: Завершение работы mininet в интерактивном режиме

Обновим репозитории программного обеспечения на виртуальной машине (рис. 21):

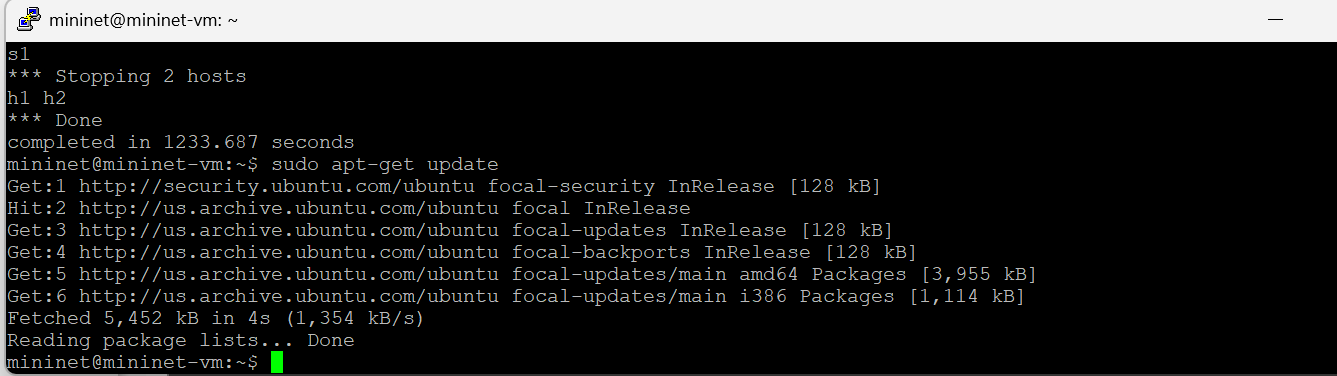


Рис. 21: Обновление репозиториев программного обеспечения на втртуальной машине

Установим пакет geeqie для просмотра файлов png (рис. 22):

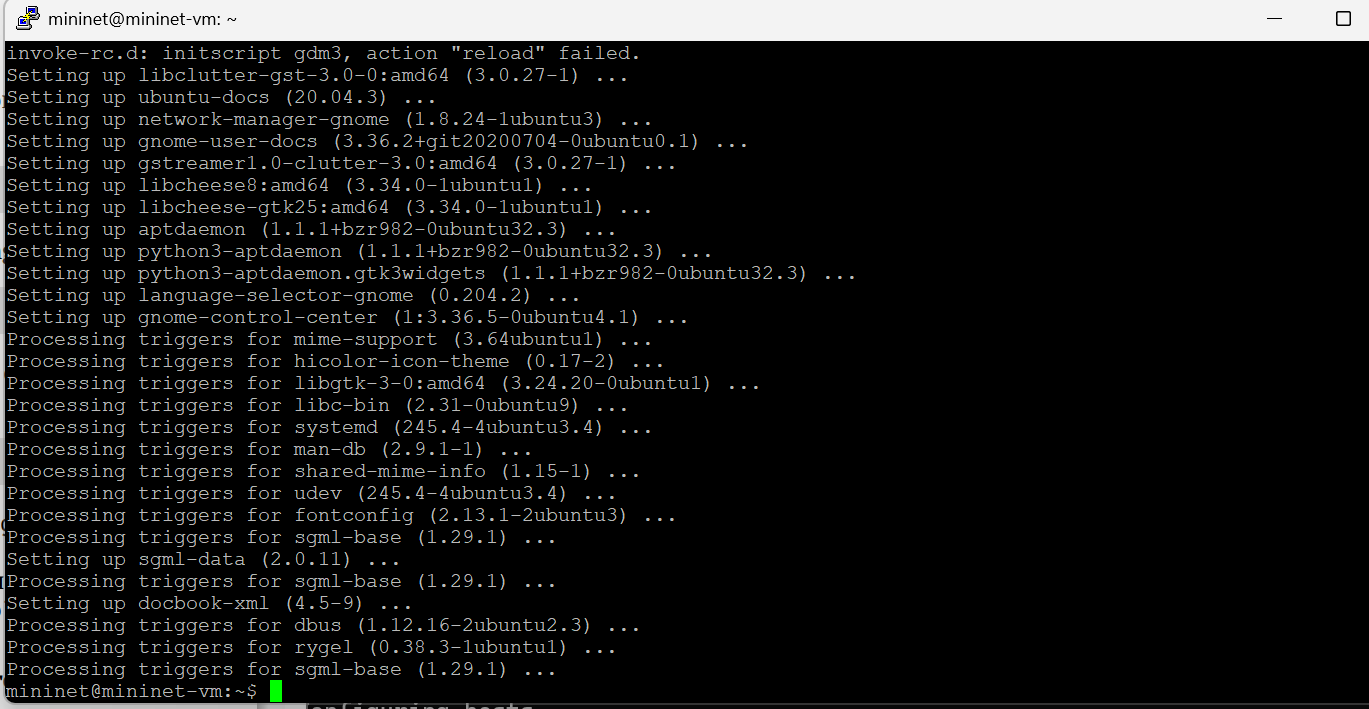


Рис. 22: Установка пакета geeqie

Для каждого воспроизводимого эксперимента expname создадим свой каталог, в котором будут размещаться файлы эксперимента (рис. 23):

Создание нового каталога

Рис. 23: Создание нового каталога

В виртуальной среде mininet в своём рабочем каталоге с проектами создадим каталог simple-delay и перейдём в него (рис. 24):

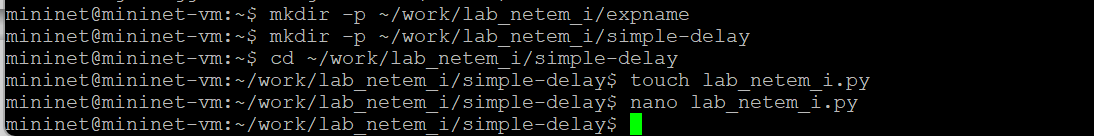


Рис. 24: Создание каталога simple-delay

Создадим скрипт для эксперимента lab\_netem\_i.py (рис. 25):

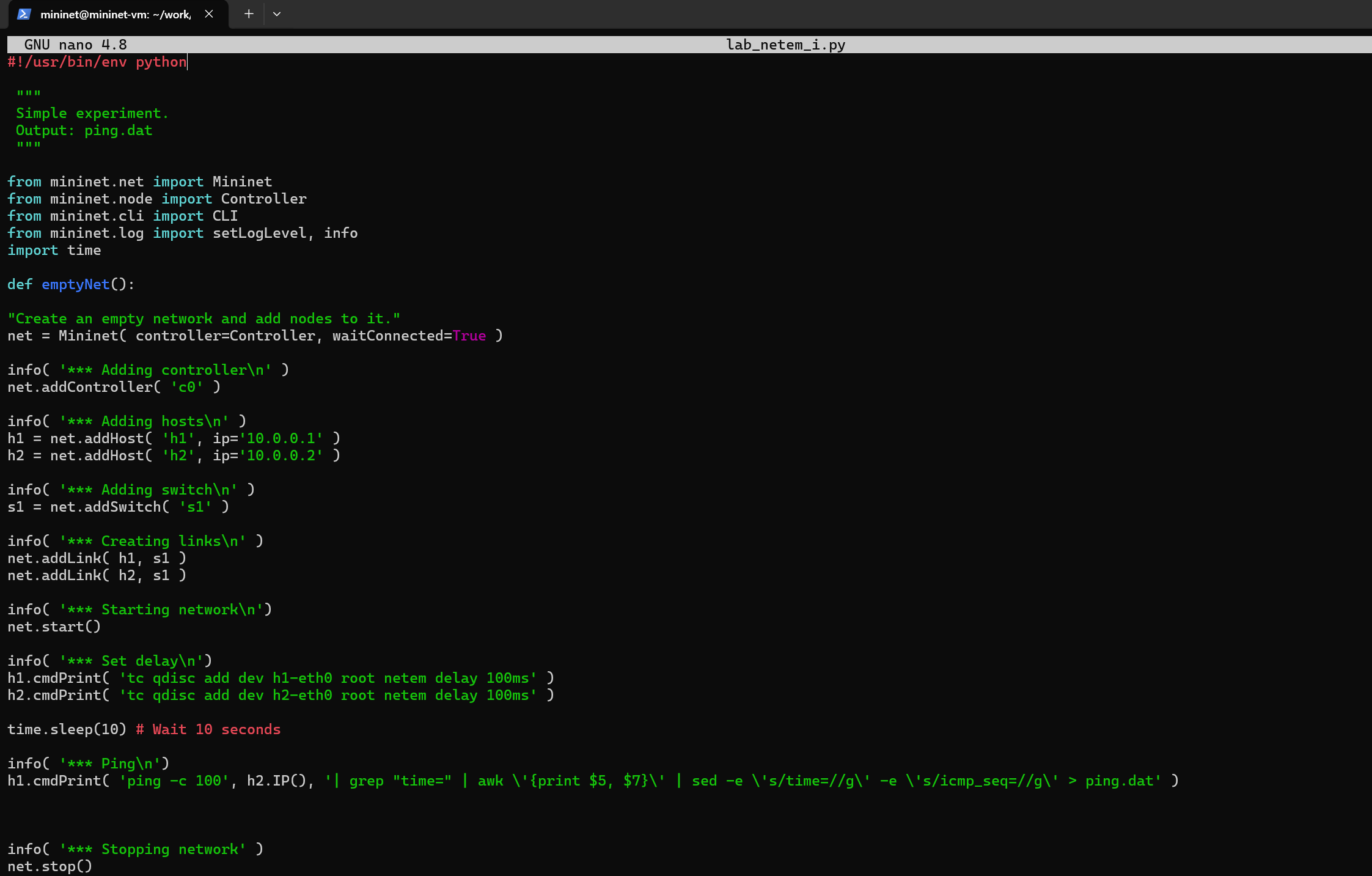


Рис. 25: Создание скрипта lab\_netem\_i.py для эксперимента

Создадим файл ping\_plot (рис. 26):

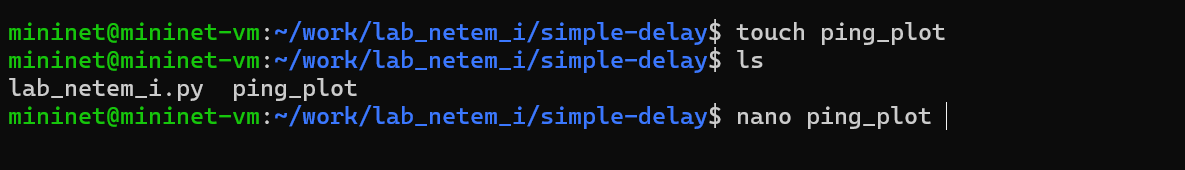


Рис. 26: Создание файла ping\_plot

Затем создадим скрипт для визуализации ping\_plot результатов эксперимента (рис. 27):

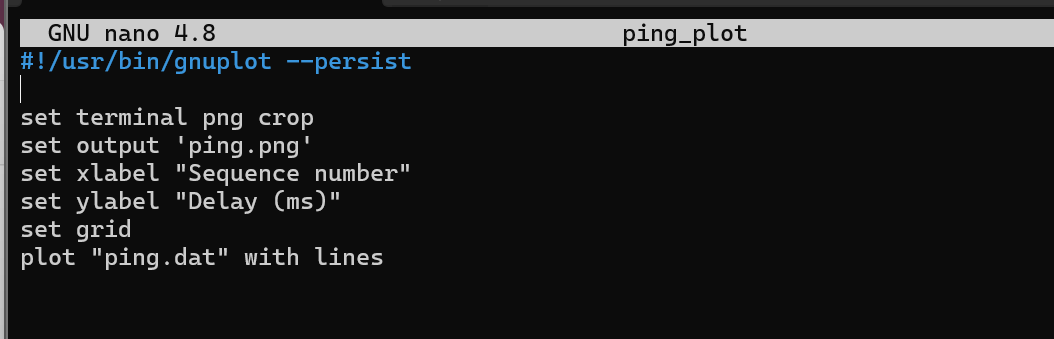


Рис. 27: Создание скрипта ping\_plot для визуализации результатов эксперимента

Зададим права доступа к файлу скрипта (рис. 28):

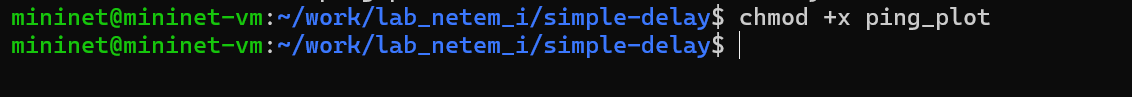


Рис. 28: Настройка прав доступа к файлу скрипта

Создадим файла Makefile (рис. 29):

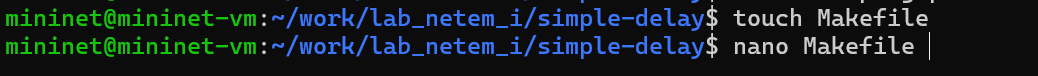


Рис. 29: Создание файла Makefile

Внутри файла Makefile поместим скрипт для управления процессом проведения эксперимента (рис. 30):



Рис. 30: Добавления скрипта в Makefile для управления процессом проведения эксперимента

Выполним эксперимент (рис. 31):

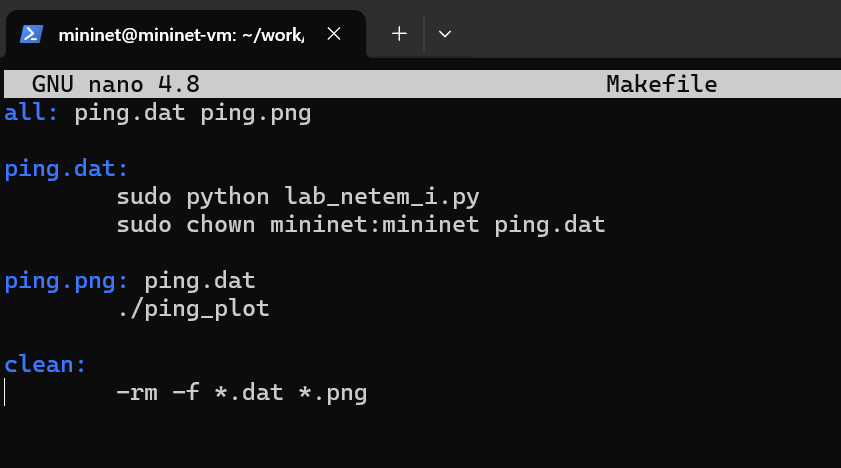


Рис. 31: Выполнение эксперимента

Просмотрим построенный в результате выполнения скриптов график (рис. 32):

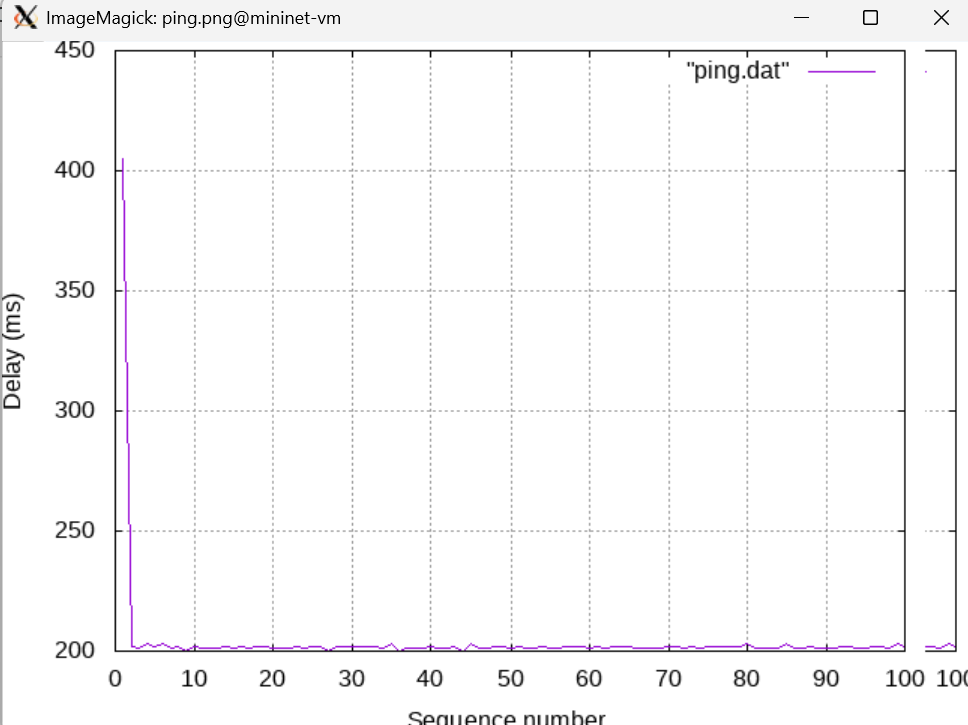


Рис. 32: Просмотр графика

Из файла ping.dat удалим первую строку и заново построим график (рис. 33 - рис. 34 ):



Рис. 33: Удаление первой строчки из файла ping.dat

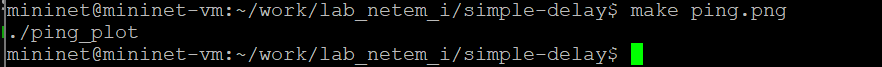


Рис. 34: Повторное построение графика

Просмотрим заново построенный график (рис. 35):

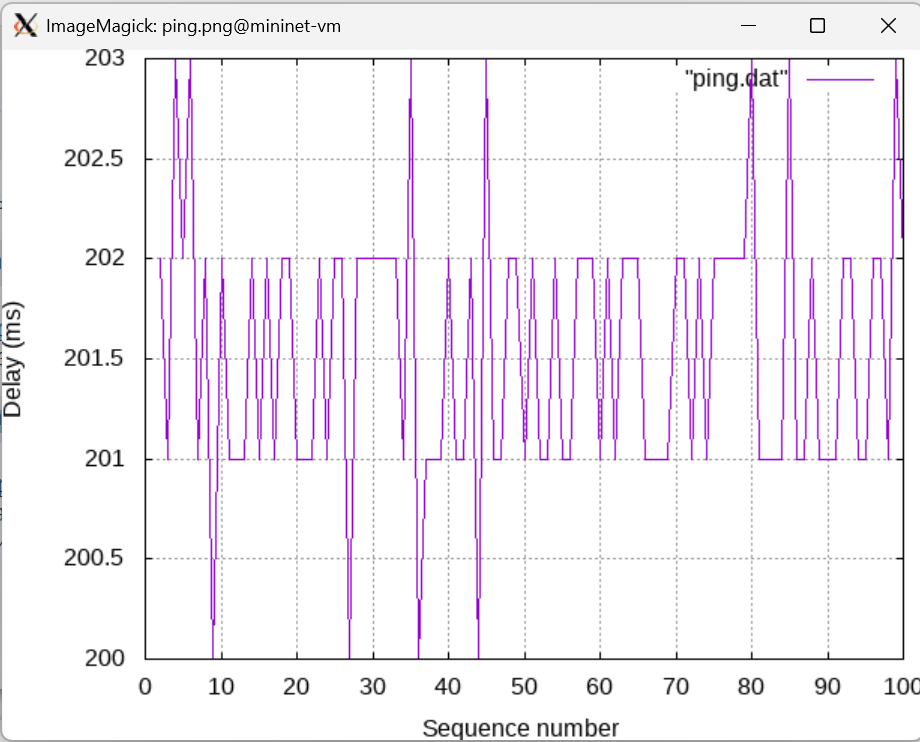


Рис. 35: Просмотр графика

Разработаем скрипт для вычисления на основе данных файла ping.dat минимального, среднего, максимального и стандартного отклонения времени приёма-передачи. Также добавим правило запуска скрипта в Makefile (рис. 36 - рис. 38):

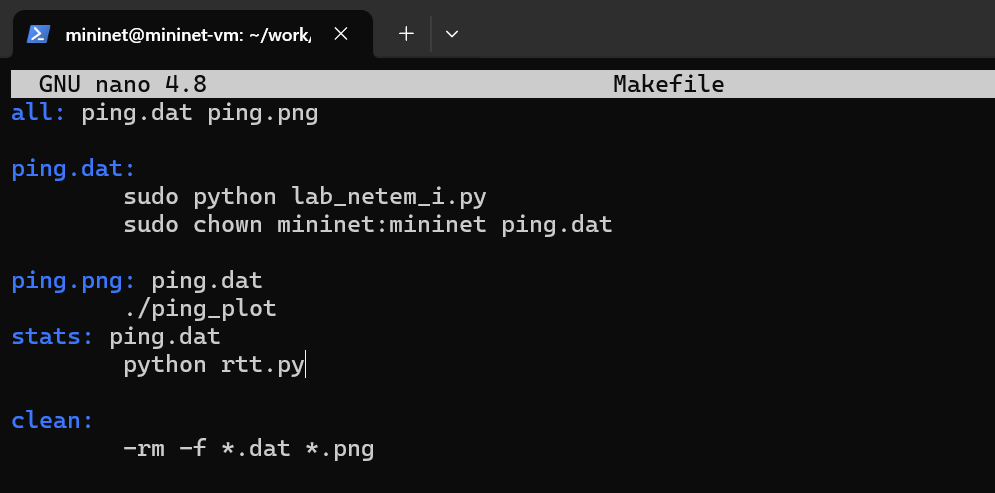


Рис. 36: Разработка скрипта для вычисления на основе данных файла ping.dat минимального, среднего, максимального и стандартного отклонения времени приёма-передачи



Рис. 37: Добавление правила запуска скрипта в Makefil

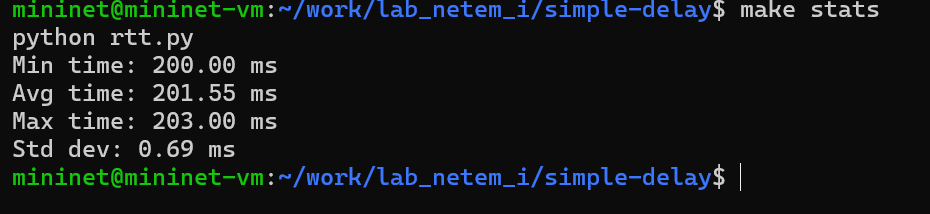


Рис. 38: Проверка

Очистим каталог от результатов проведения экспериментов.

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. 39 - рис. 50):

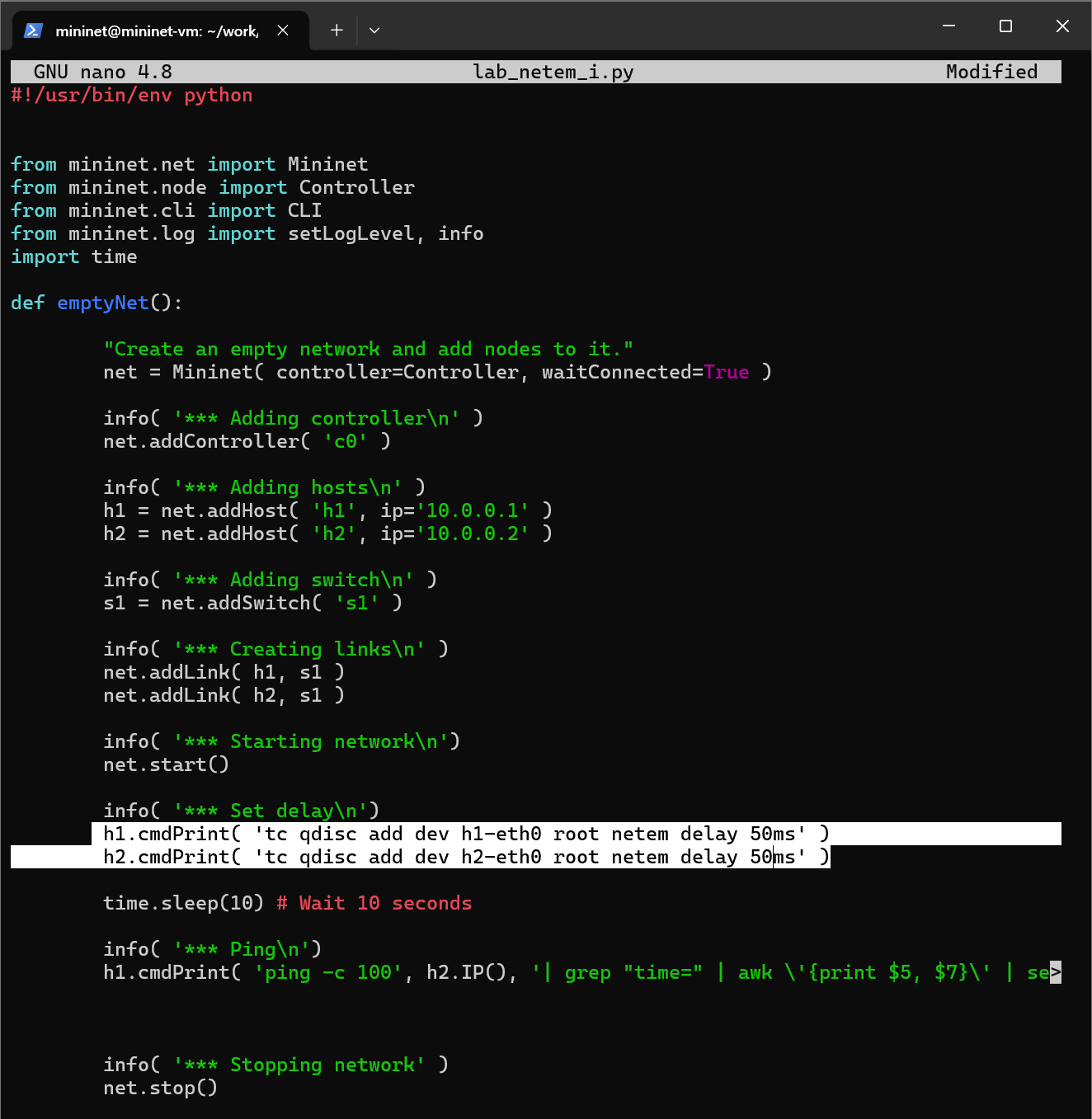


Рис. 39: Воспроизводимый эксперимент по изменению задержки

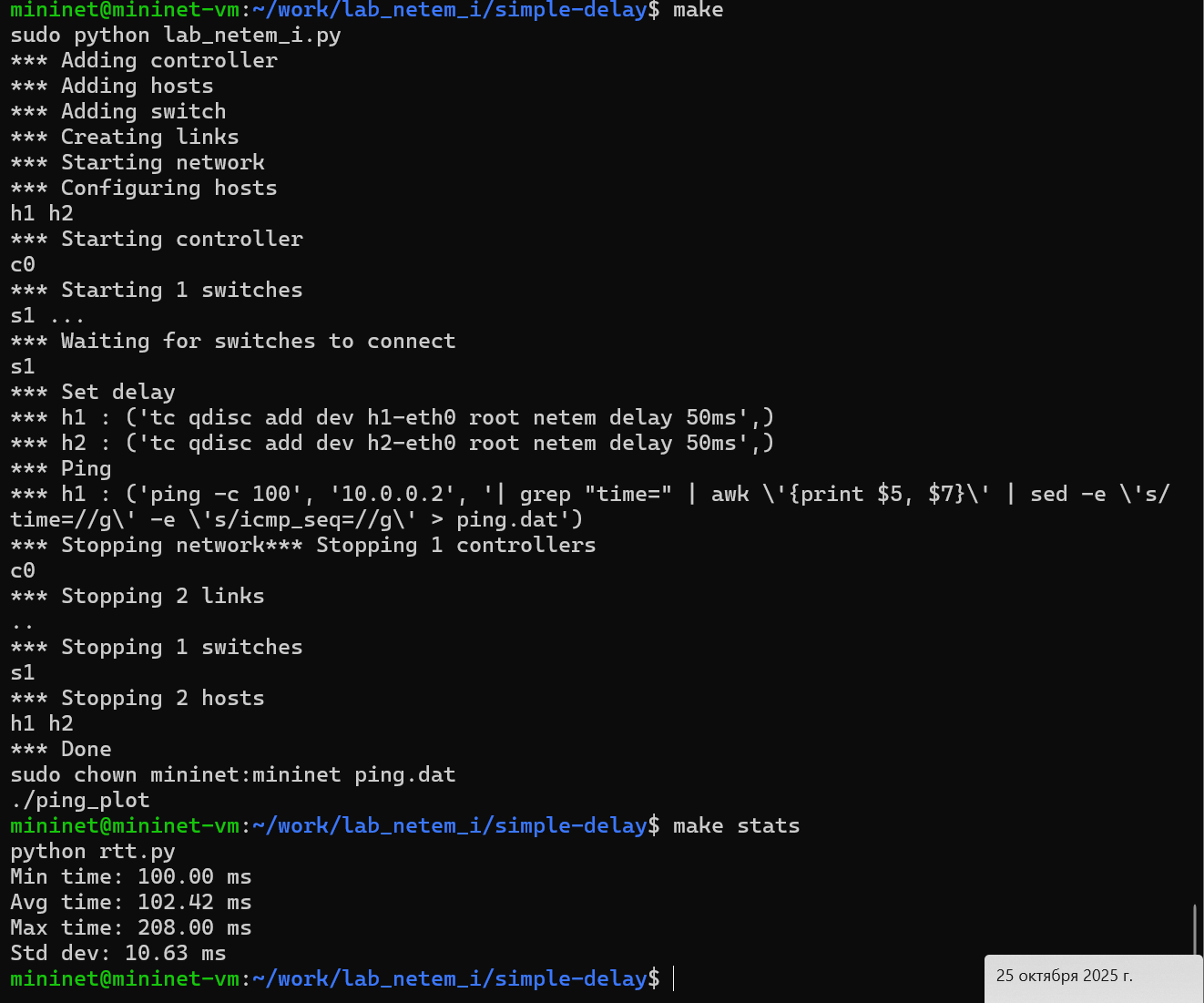


Рис. 40: Воспроизводимый эксперимент по изменению задержки

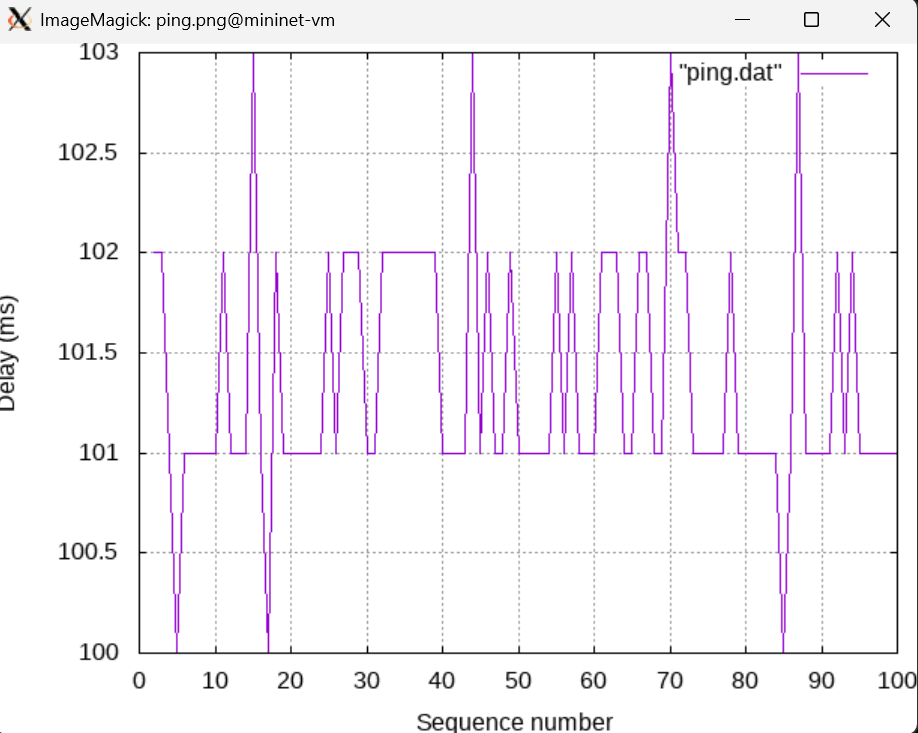


Рис. 41: Просмотр графика

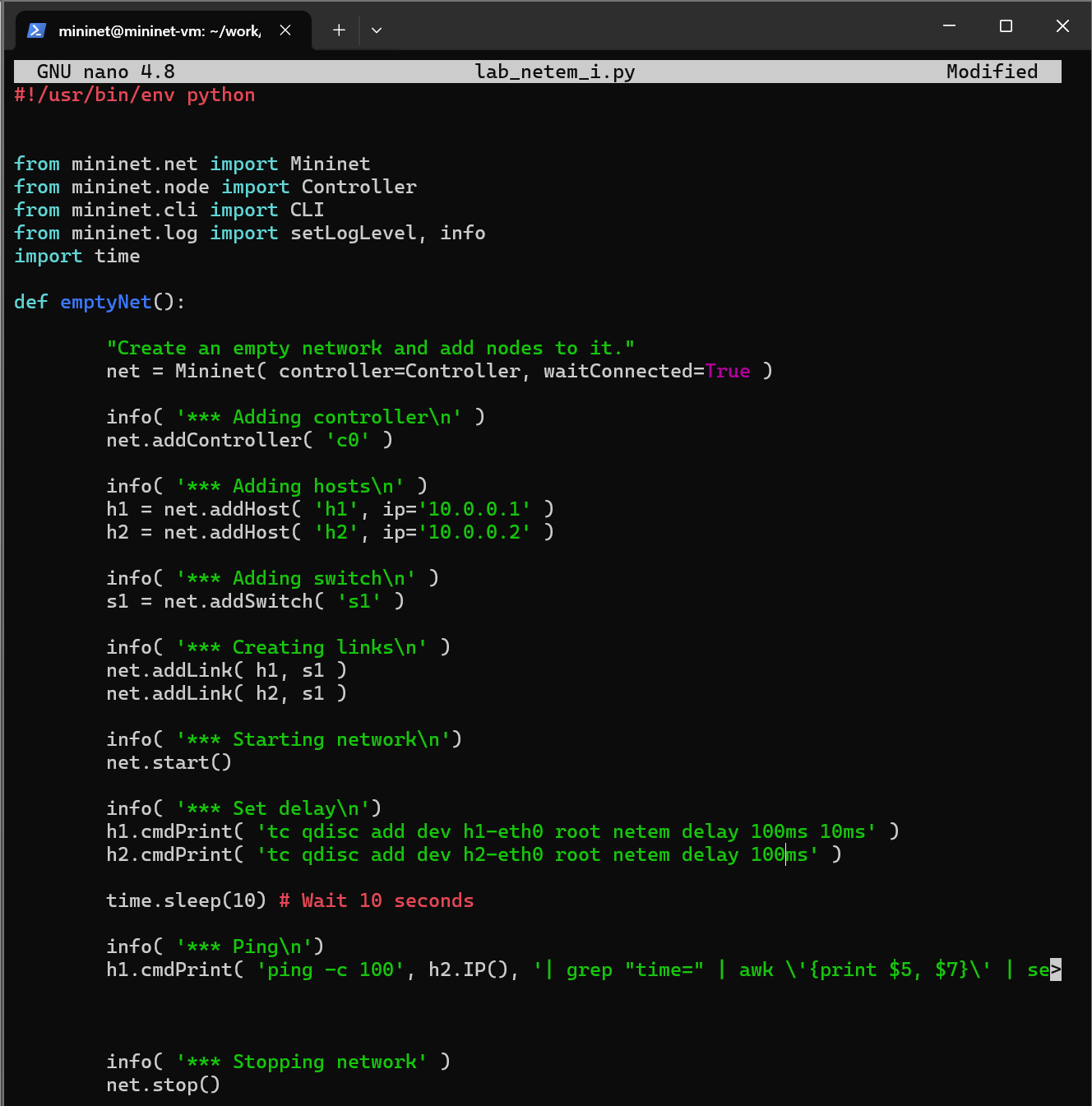


Рис. 42: Воспроизводимый эксперимент по изменению джиттера

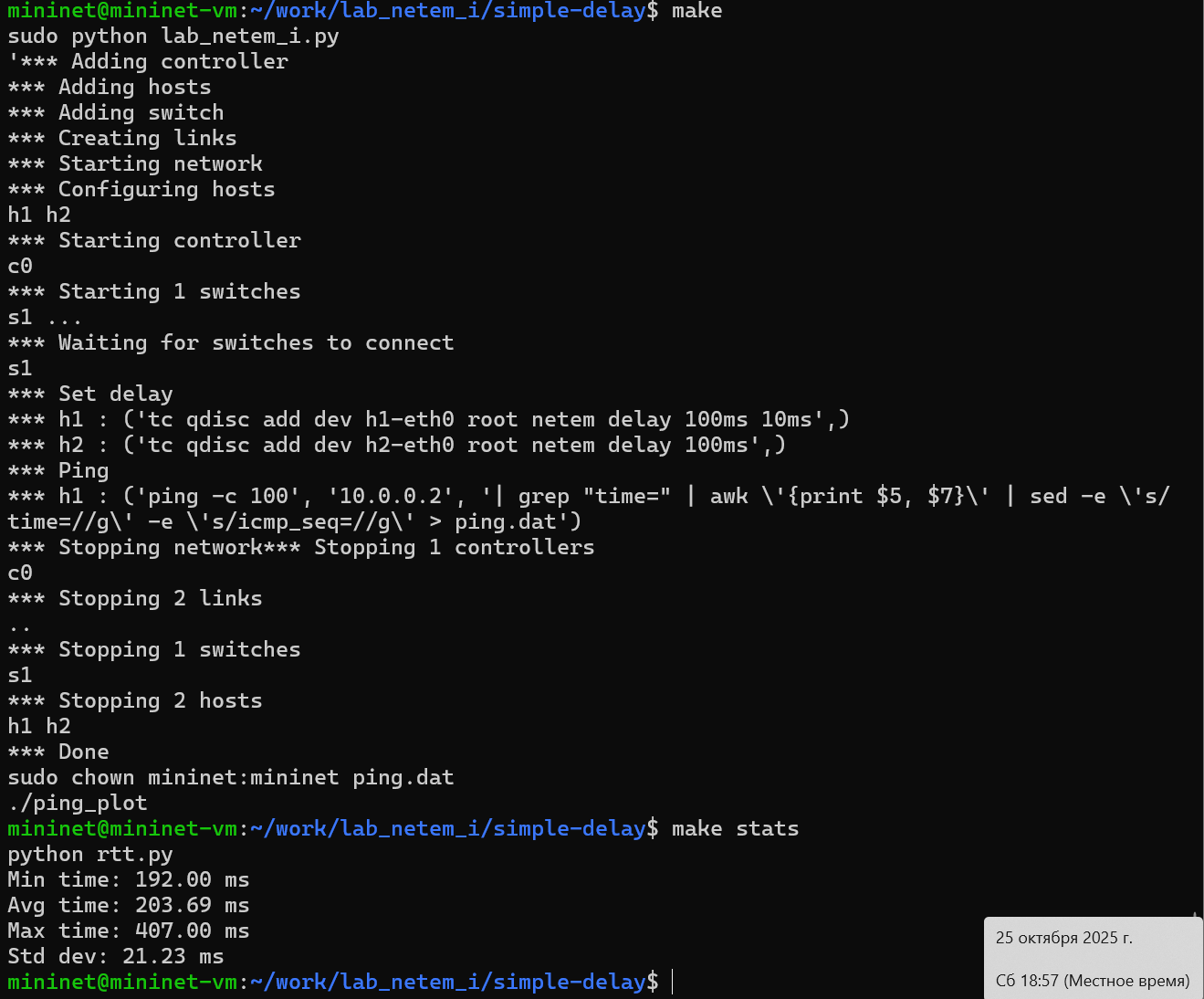


Рис. 43: Воспроизводимый эксперимент по изменению джиттера

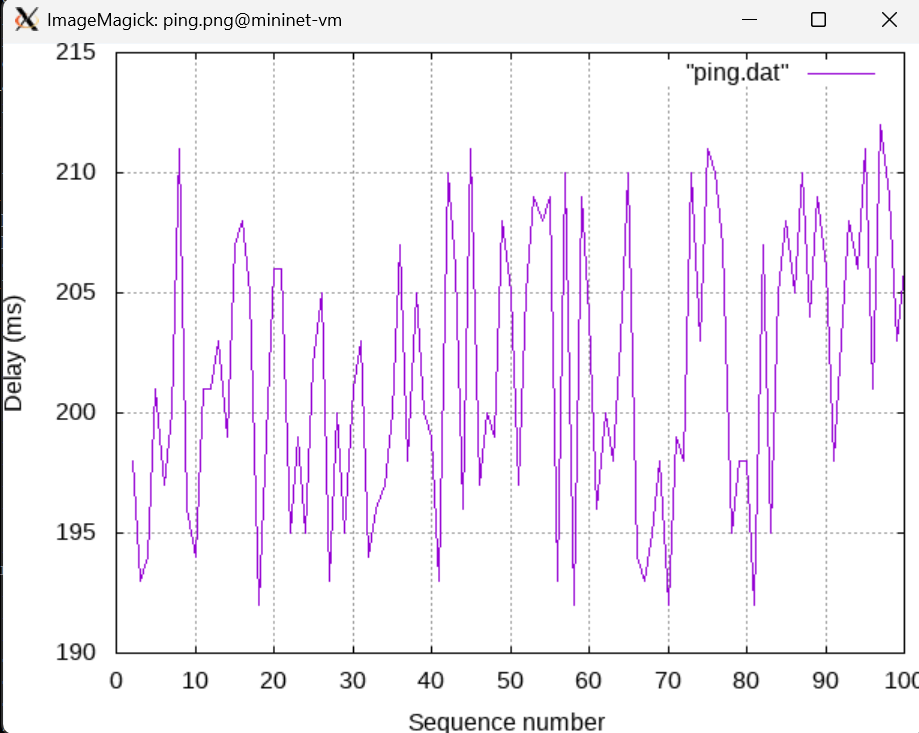


Рис. 44: Просмотр графика

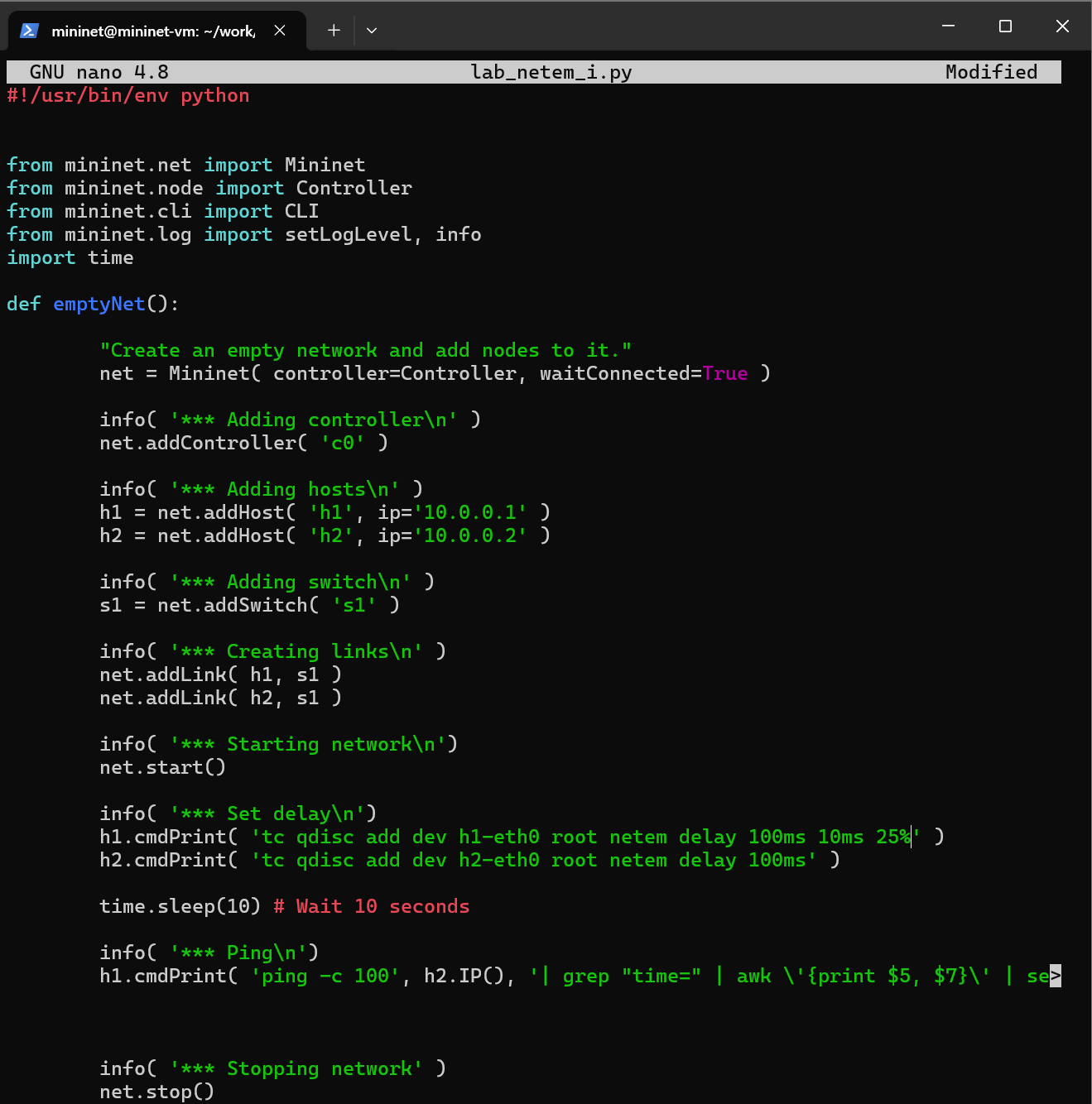


Рис. 45: Воспроизводимый эксперимент по изменению значения корреляции для джиттера и задержки

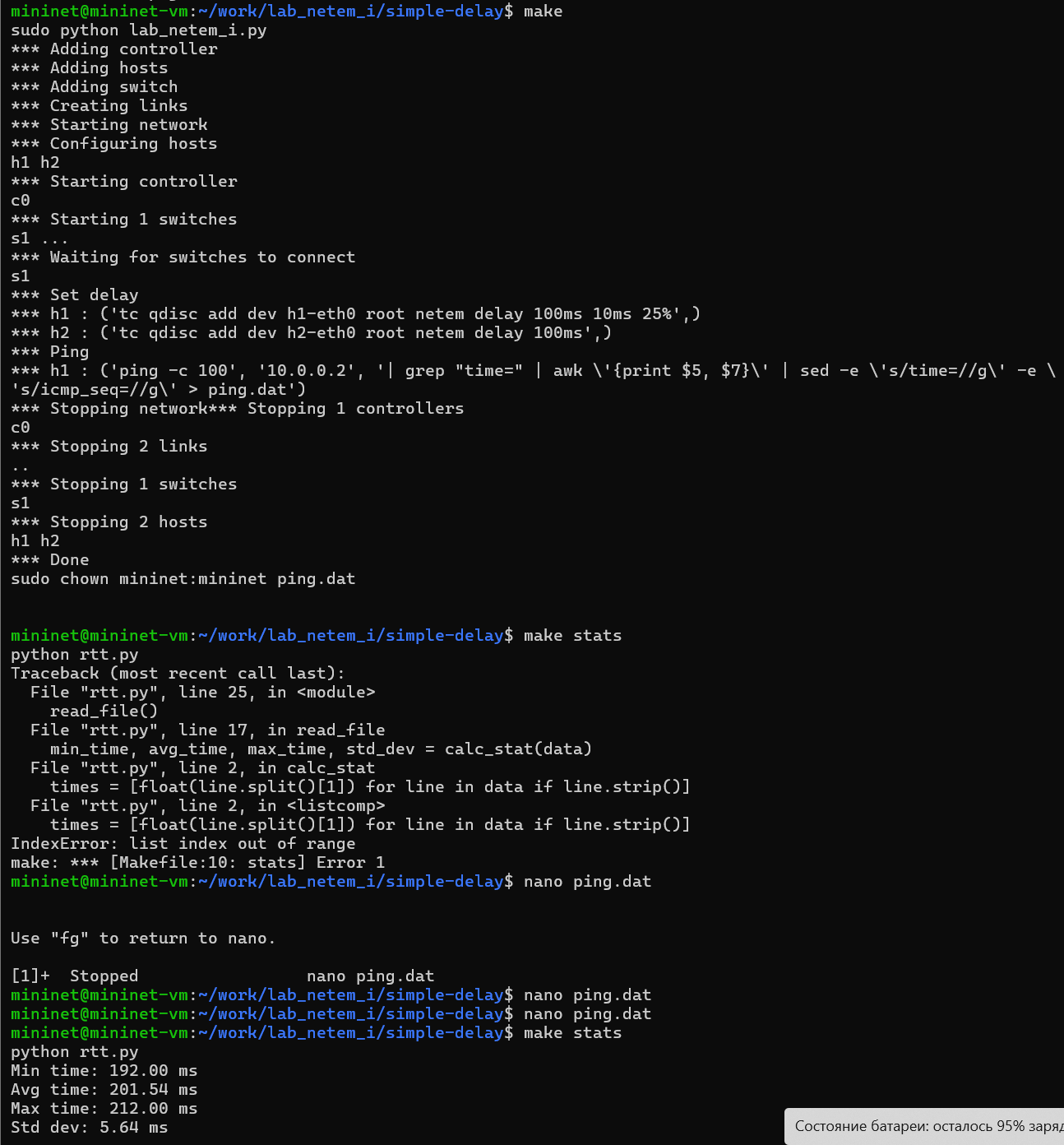


Рис. 46: Воспроизводимый эксперимент по изменению значения корреляции для джиттера и задержки

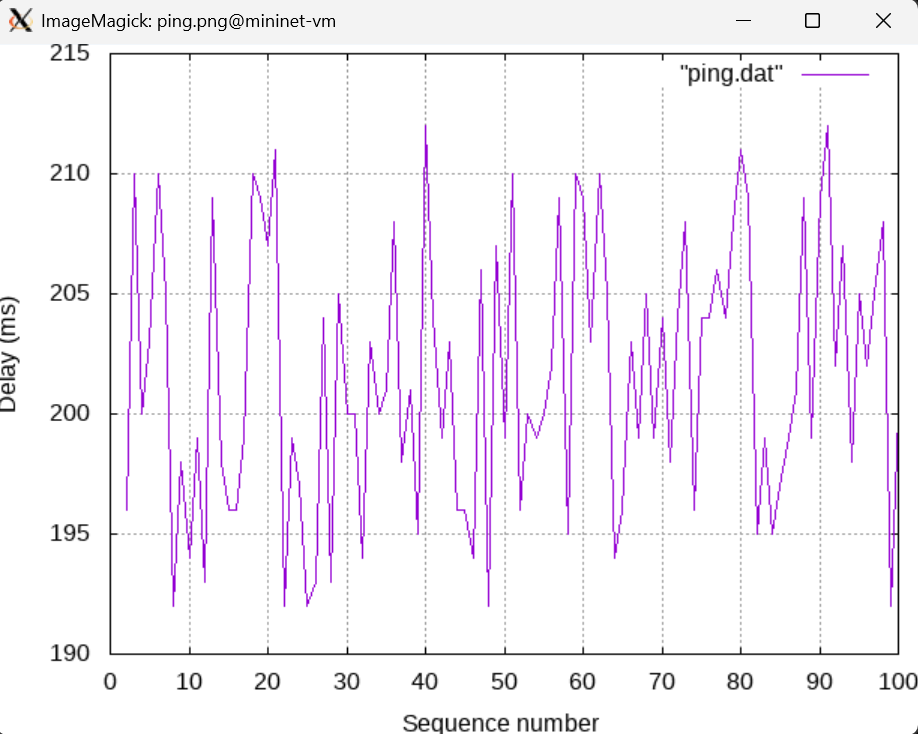


Рис. 47: Просмотр графика

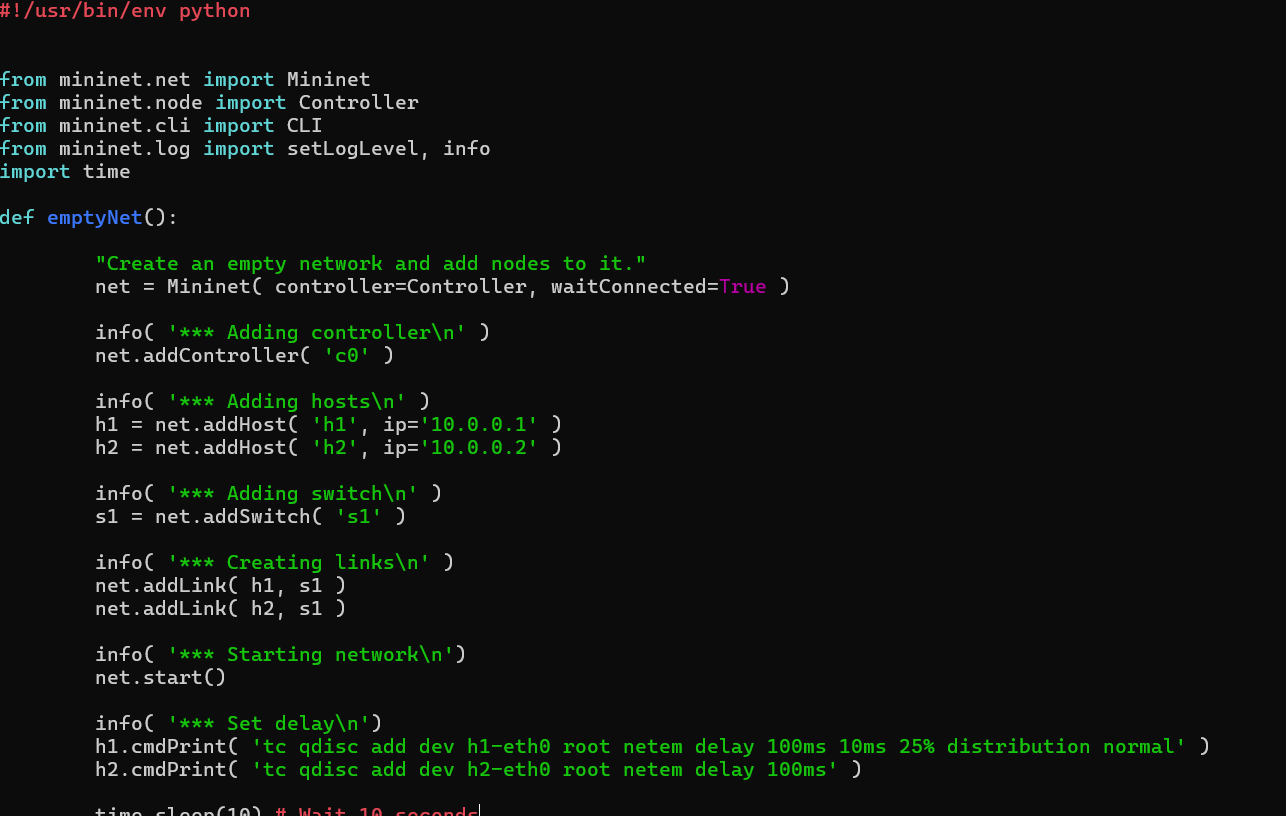


Рис. 48: Воспроизводимый эксперимент по изменению распределения времени задержки в эмулируемой глобальной сети

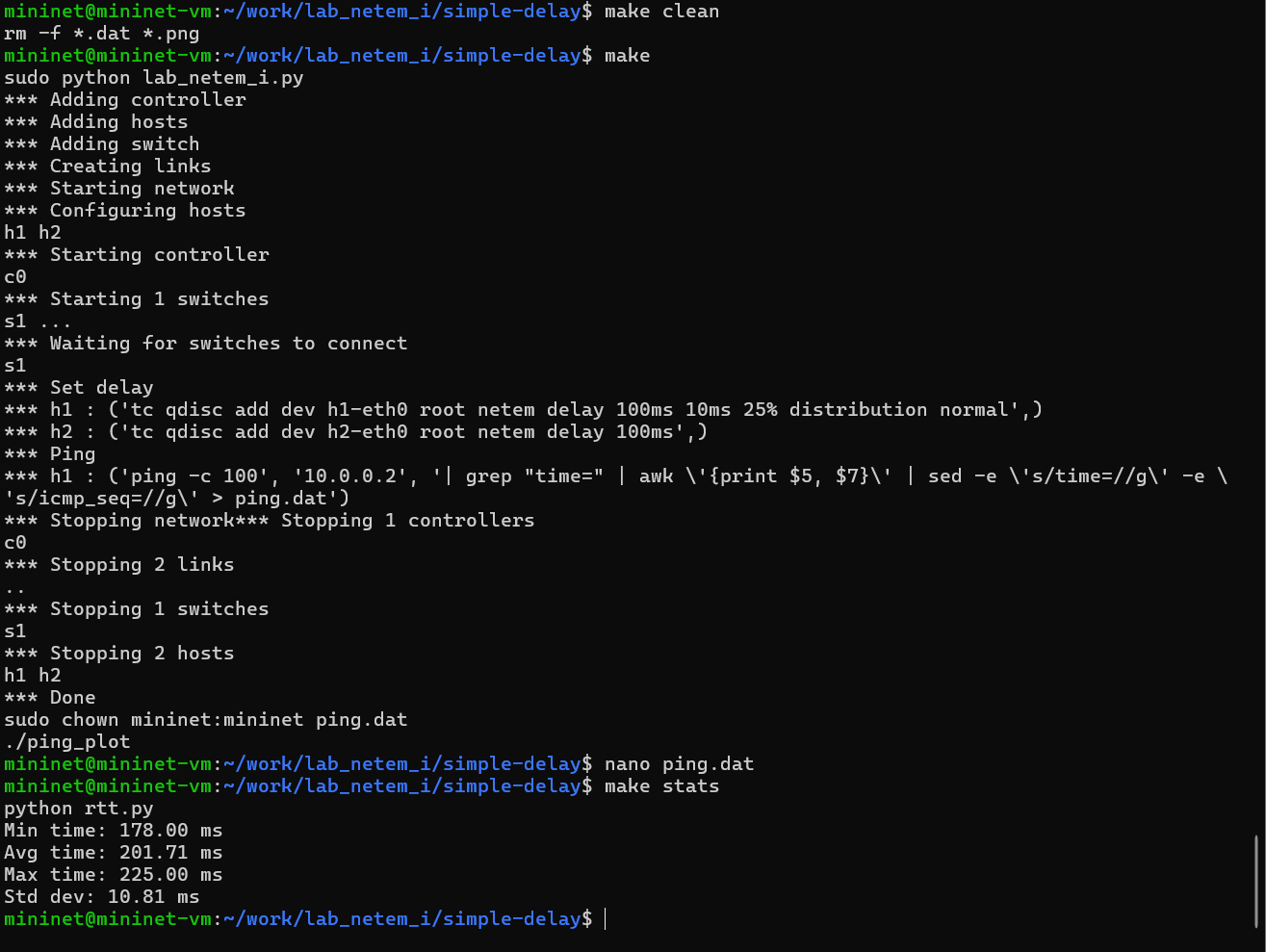


Рис. 49: Воспроизводимый эксперимент по изменению распределения времени задержки в эмулируемой глобальной сети

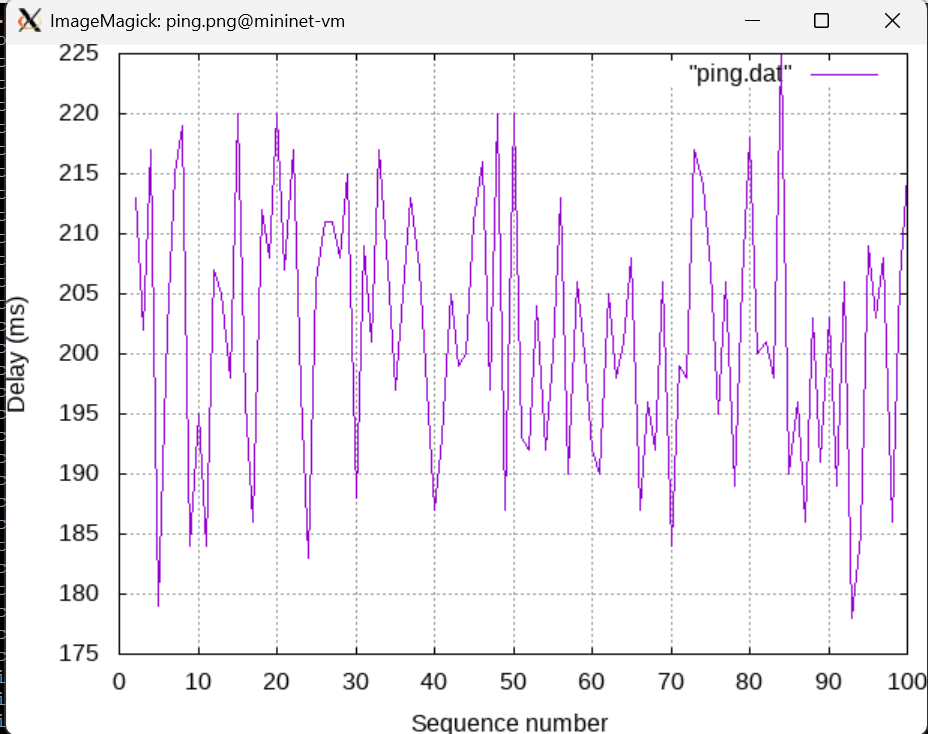


Рис. 50: Просмотр графика

# 4 Выводы

В ходе выполнения лабораторной работы мы познакомились 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.

# Список литературы

1. NETEM [Электронный ресурс]. URL: <https://habr.com/ru/articles/949088/>.