Лабораторная работа №4

Эмуляция и измерение задержек в глобальных сетях

Ланцова Яна Игоревна

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.

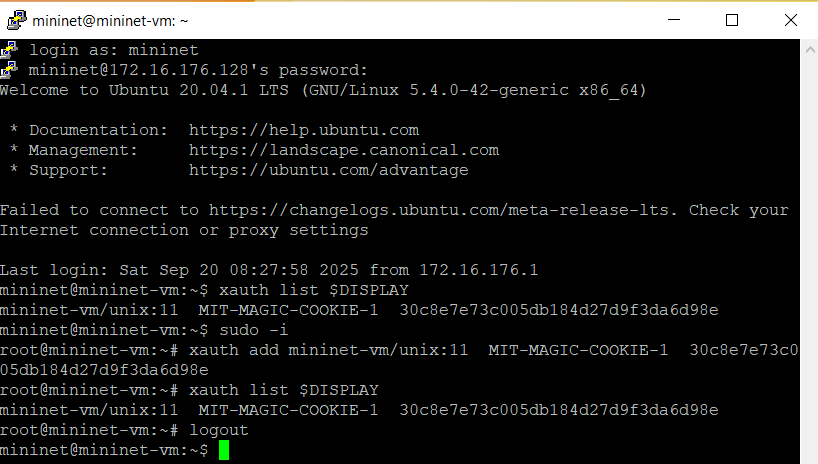
# 2 Задачи

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

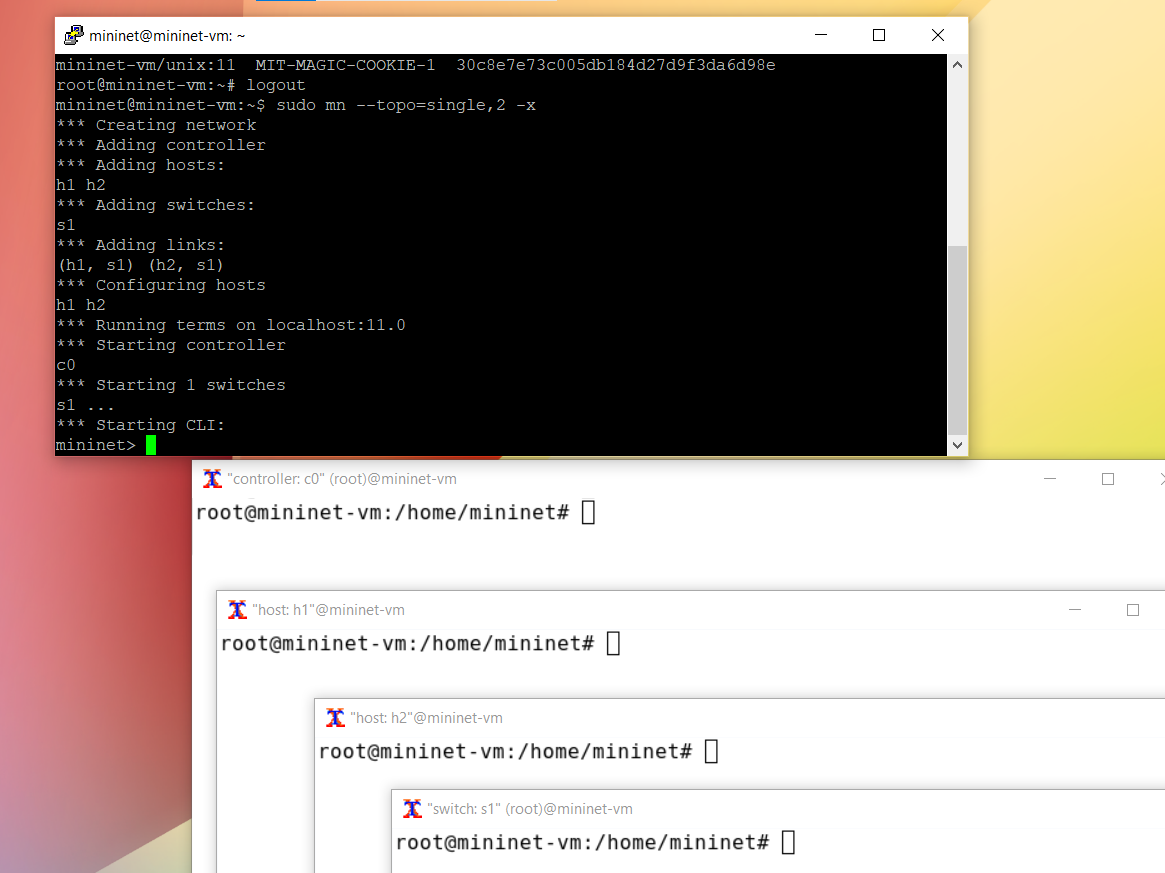
## 3.1 Запуск лабораторной топологии

Из основной ОС подключимся к виртуальной машине и в исправим права запуска X-соединения. Скопируем значение куки (MIT magic cookie) своего пользователя mininet в файл для пользователя root (рис. [**fig:001?**]).



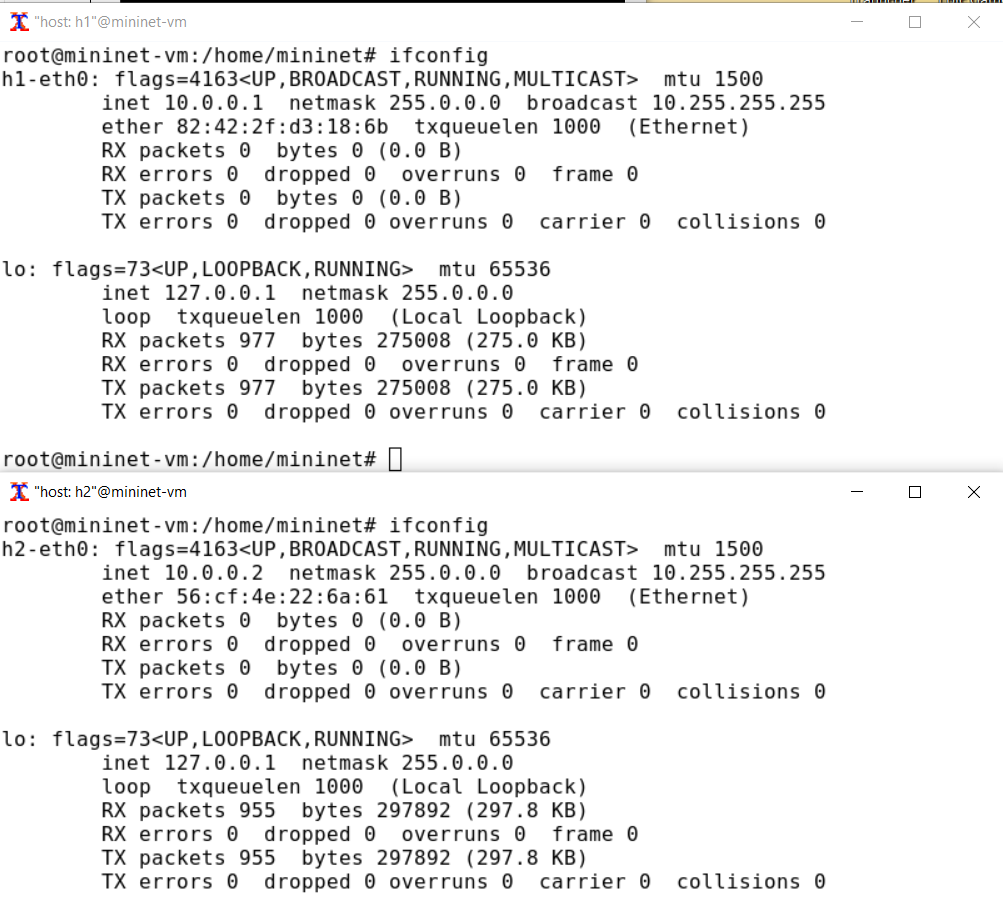
Исправление MIT magic cookie

Зададим простейшую топологию, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8 (рис. [**fig:002?**]). После введения этой команды запустятся терминалы двух хостов, коммутатора и контроллера. Терминалы коммутатора и контроллера можно закрыть.



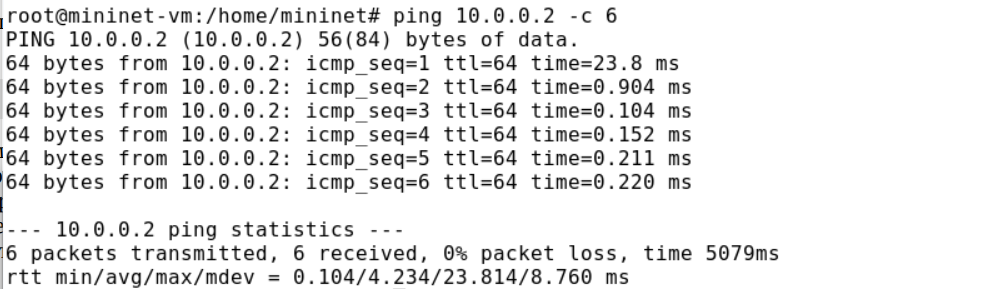
Простейшая топология

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. [**fig:003?**]).



ifconfig на хостах h1 и h2

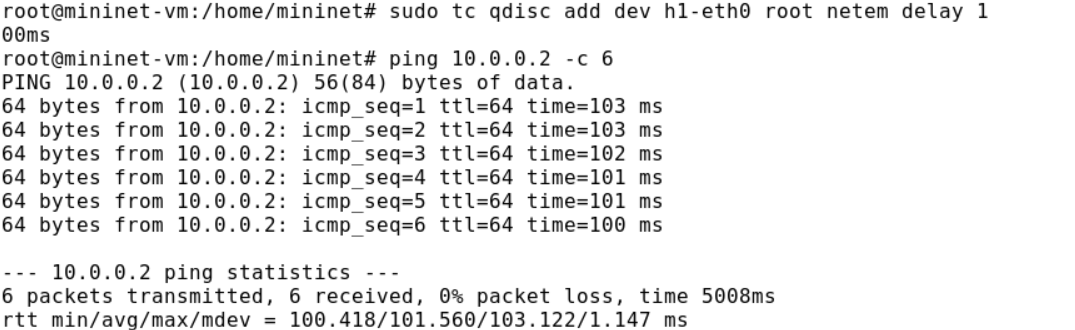
Проверим подключение между хостами h1 и h2 с помощью команды ping с параметром -c 6 (рис. [**fig:004?**]).



Проверка подключения между хостами

## 3.2 Добавление/изменение задержки в эмулируемой глобальной сети

На хосте h1 добавим задержку в 100 мс к выходному интерфейсу. Проверим, что соединение от хоста h1 к хосту h2 имеет задержку 100 мс, используя команду ping с параметром -c 6 с хоста h1 (рис. [**fig:005?**]).

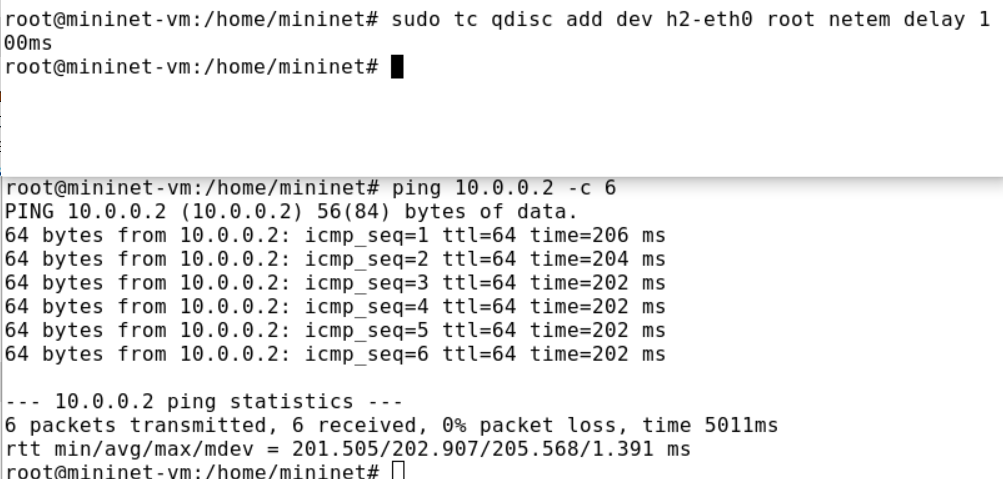


Добавление задержки в 100мс

В введенной команде:

* 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.

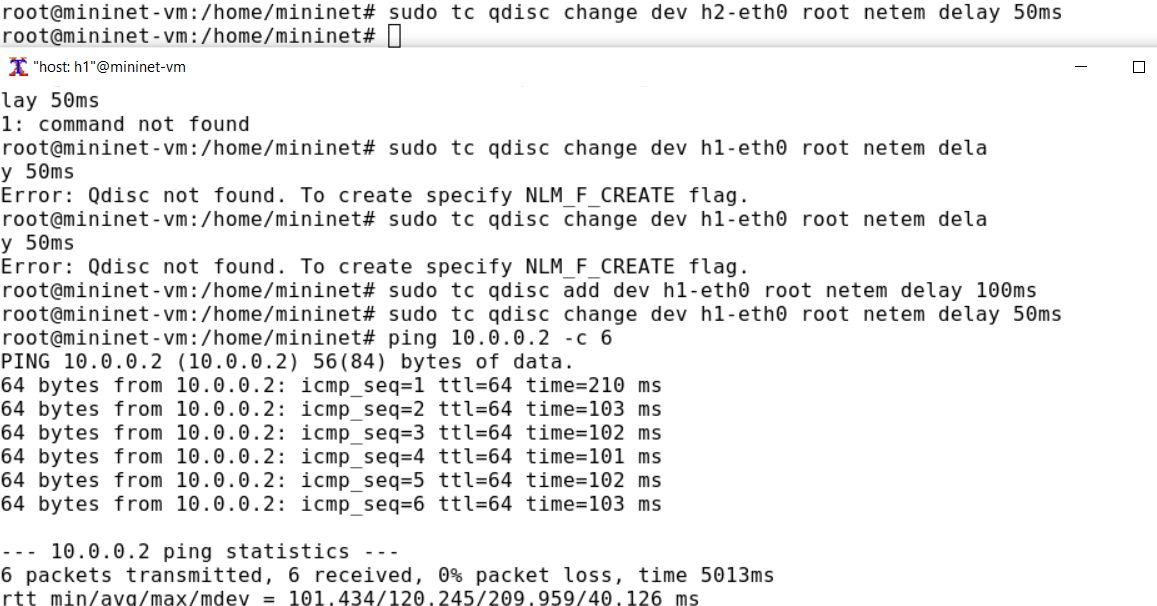
Для эмуляции глобальной сети с двунаправленной задержкой необходимо к соответствующему интерфейсу на хосте h2 также добавить задержку в 100 миллисекунд. Проверим, что соединение между хостом h1 и хостом h2 имеет RTT в 200 мс (100 мс от хоста h1 к хосту h2 и 100 мс от хоста h2 к хосту h1), повторив команду ping с параметром -c 6 на терминале хоста h1 (рис. [**fig:006?**]).



Двунаправленная задержка соединения

## 3.3 Изменение задержки в эмулируемой глобальной сети

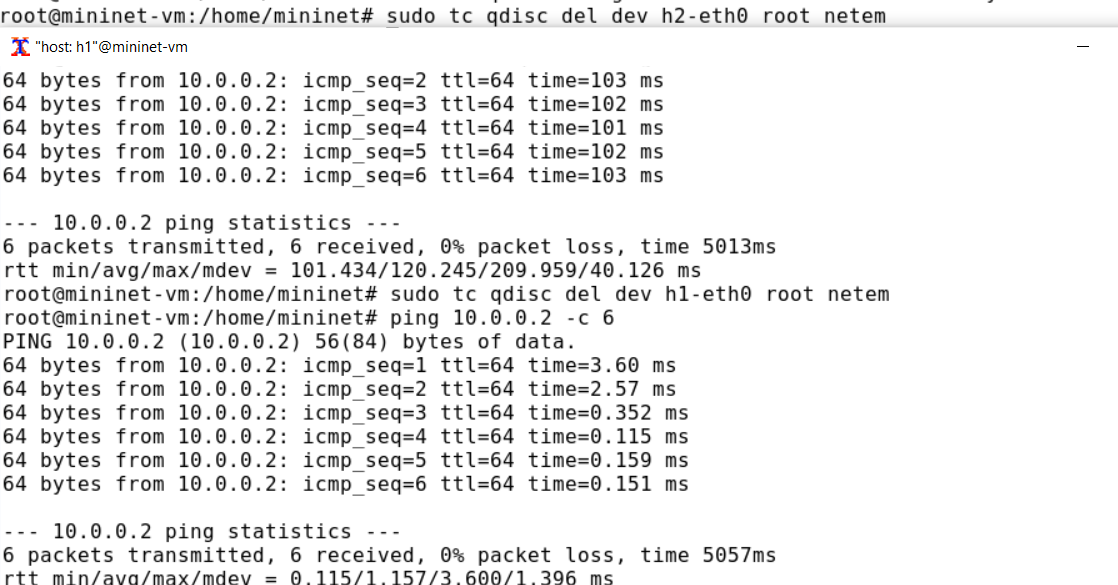
Изменим задержку со 100 мс до 50 мс для отправителя h1 и для получателя h2. Проверим, что соединение от хоста h1 к хосту h2 имеет задержку 100 мс, используя команду ping с параметром -c 6 с терминала хоста h1. (рис. [**fig:007?**]).



Изменение задержки на 50мс

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

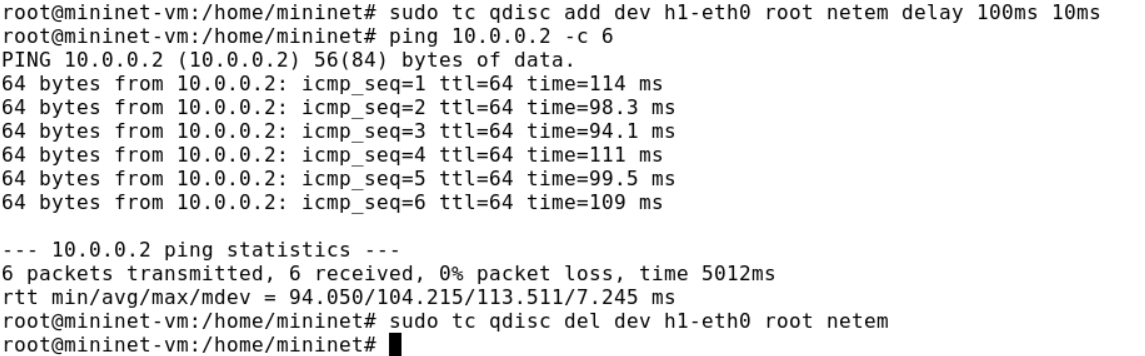
Восстановим конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса для отправителя h1 и для получателя h2. Проверим, что соединение между хостом h1 и хостом h2 не имеет явно установленной задержки, используя команду ping с параметром -c 6 с терминала хоста h1 (рис. [**fig:008?**]).



Восстановление исходных значений задержки

## 3.5 Добавление значения дрожания задержки в интерфейс подключения к эмулируемой глобальной сети

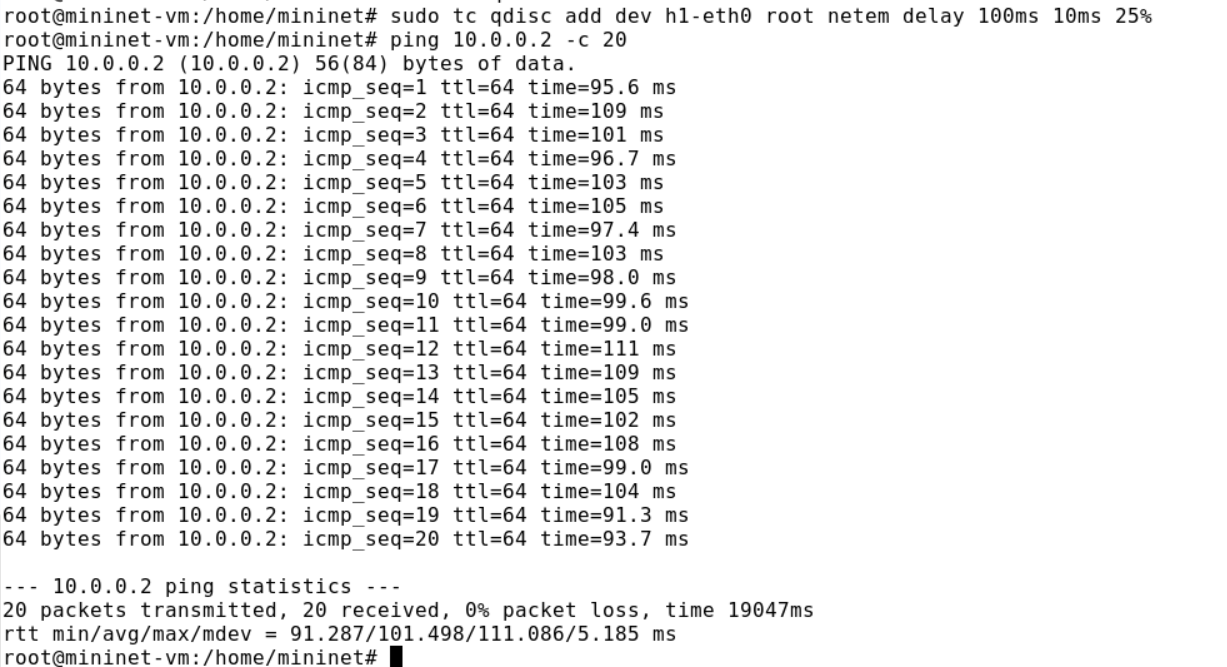
Добавим на узле h1 задержку в 100 мс со случайным отклонением 10 мс. Проверим, что соединение от хоста h1 к хосту h2 имеет задержку 100 мс со случайным отклонением ±10 мс, используя в терминале хоста h1 команду ping с параметром -c 6. Восстановим конфигурацию интерфейса по умолчанию на узле h1 (рис. [**fig:009?**]). Увидим, что в первом случае у нас создалась сеть с настроенными параметрами, а во втором случае дефолтная сеть без этих параметров.



Добавление значения дрожания задержки в интерфейс подключения

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

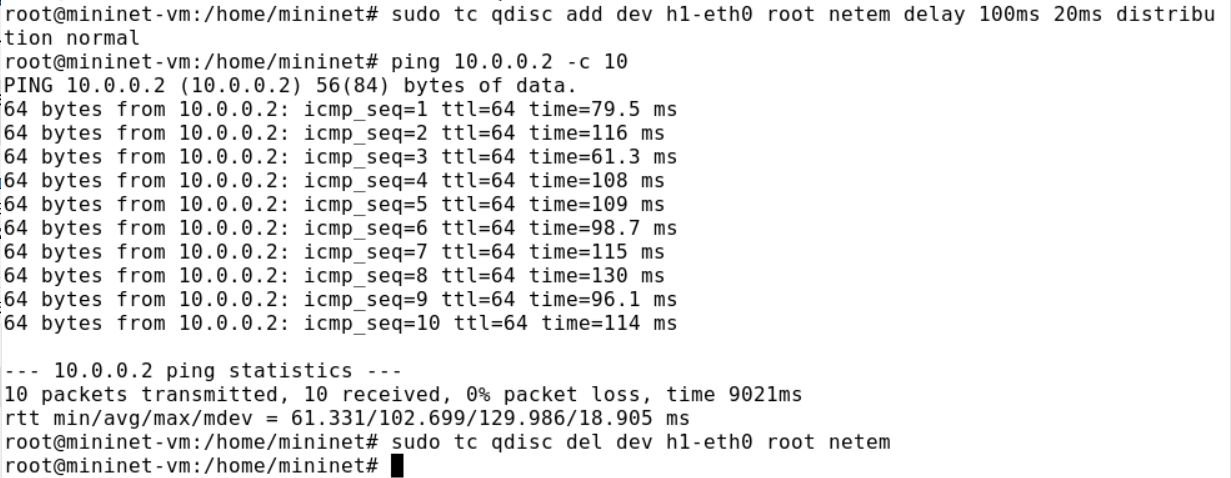
Добавим на интерфейсе хоста h1 задержку в 100 мс с вариацией ±10 мс и значением корреляции в 25%. Убедимся, что все пакеты, покидающие устройство h1 на интерфейсе h1-eth0, будут иметь время задержки 100 мс со случайным отклонением ±10 мс, при этом время передачи следующего пакета зависит от предыдущего 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. [**fig:010?**]).



Добавление значения корреляции для джиттера и задержки в интерфейс подключения

## 3.7 Распределение задержки в интерфейсе подключения к эмулируемой глобальной сети

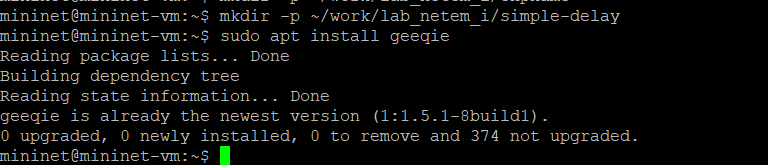
Зададим нормальное распределение задержки на узле h1 в эмулируемой сети. Убедимся, что все пакеты, покидающие хост h1 на интерфейсе h1-eth0, будут иметь время задержки, которое распределено в диапазоне 100 мс ±20 мс. Используем для этого команду ping на терминале хоста h1 с параметром -c 10. Восстановим конфигурацию интерфейса по умолчанию на узле h1. Завершим работу mininet в интерактивном режиме(рис. [**fig:011?**]):



Распределение задержки в интерфейсе подключения

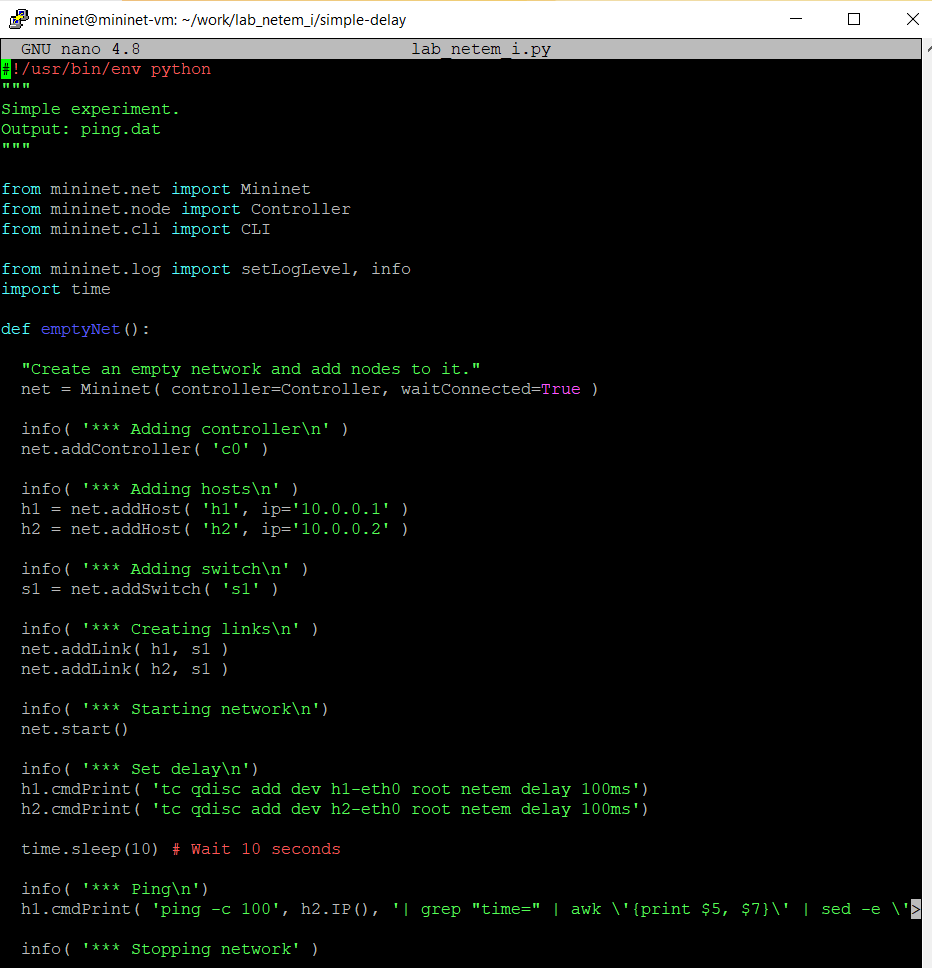
## 3.8 Воспроизведение экспериментов. Добавление задержки для интерфейса, подключающегося к эмулируемой глобальной сети

С помощью API Mininet воспроизведем эксперимент по добавлению задержки для интерфейса хоста, подключающегося к эмулируемой глобальной сети. В виртуальной среде mininet установим пакет geeqie (понадобится для просмотра файлов png), в своём рабочем каталоге с проектами создадим каталог simple-delay и перейдем в него (рис. [**fig:012?**]).



Установка пакета и создание каталога

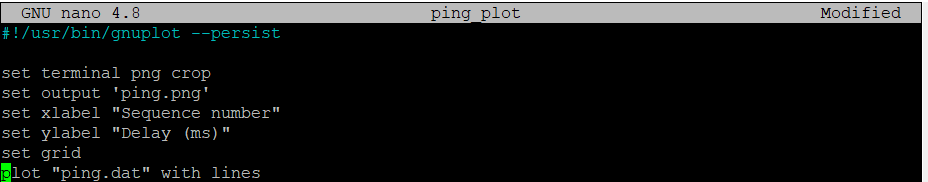
Создадим скрипт для эксперимента lab\_netem\_i.py (рис. [**fig:013?**]).



Скрипт на Python для эксперемента

В этом скрипте создается простейщая топология сети, затем с помощью комнанд, использованных нами ранее задается задержка в 100 мс для обоих хостов, после чего пингуется второй хост (100 сообщений отправляется), при этом из сообщений при пинге вытаскиваются номер сообщения и значение времени, которые записываются в файл с данными.

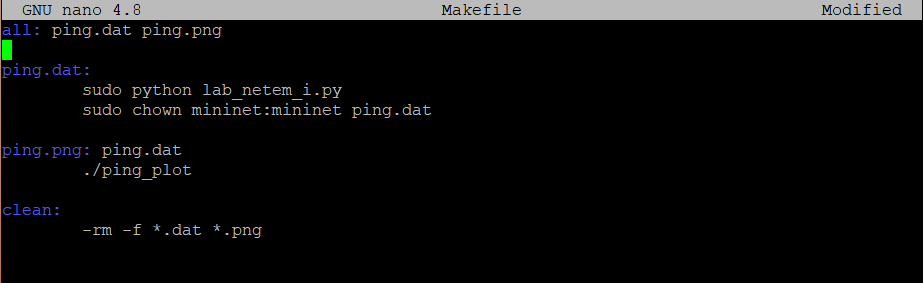
Создаём скрипт для визуализации ping\_plot результатов эксперимента (рис. [**fig:014?**]):



Скрипт для визуализации ping\_plot

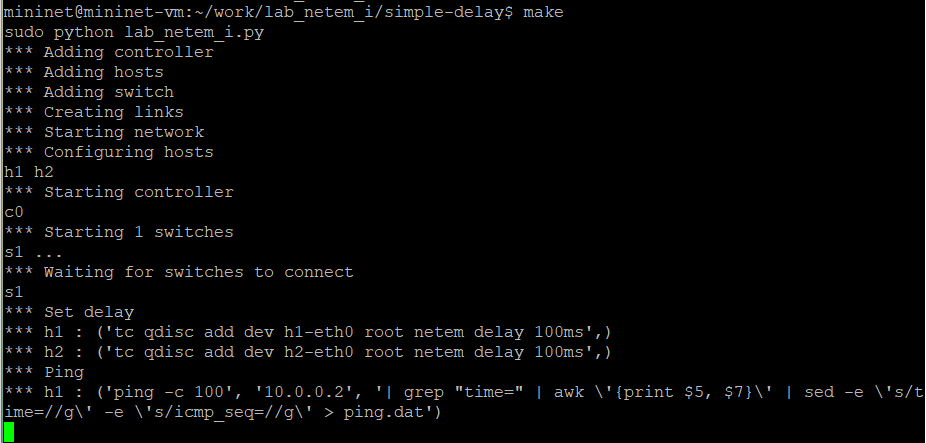
Зададим права доступа к файлу скрипта: chmod +x ping\_plot.

Создадим Makefile для управления процессом проведения эксперимента (рис. [**fig:015?**]).



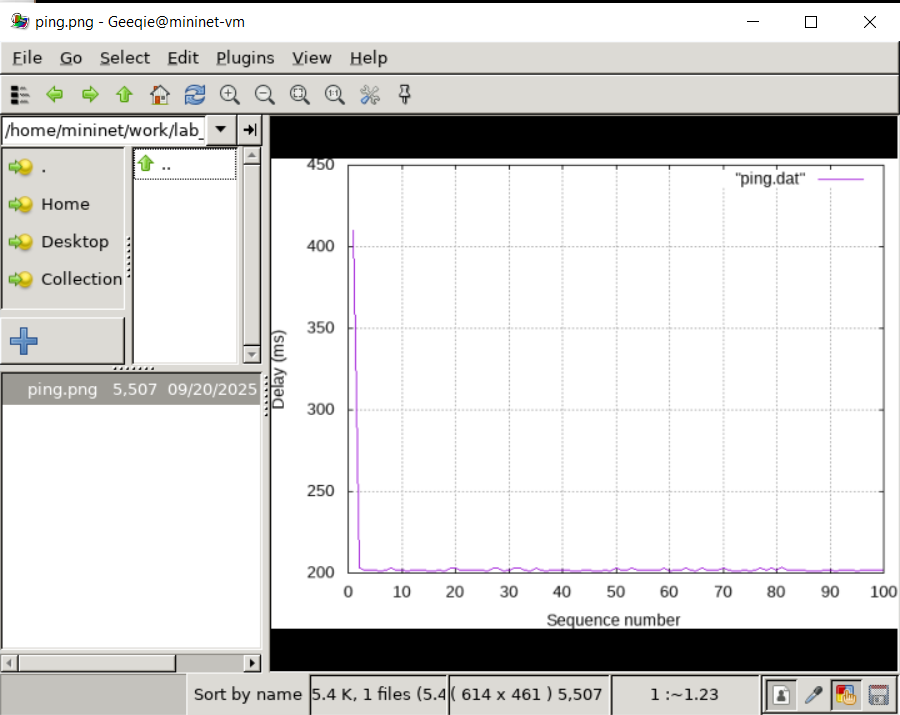
Makefile для управления процессом проведения эксперимента

Выполним эксперимент, написав команду make (рис. [**fig:016?**]).



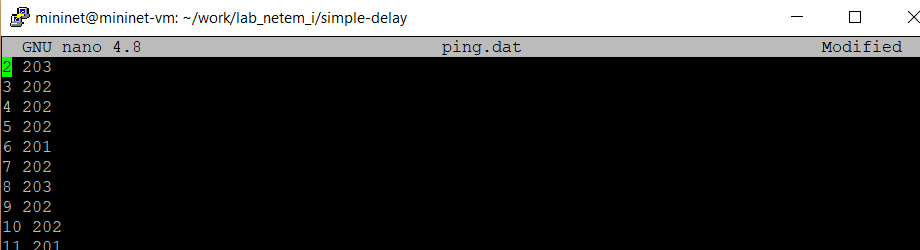
Запуск эксперимента

Продемонстрируем построенный в результате выполнения скриптов график (рис. [**fig:017?**]).



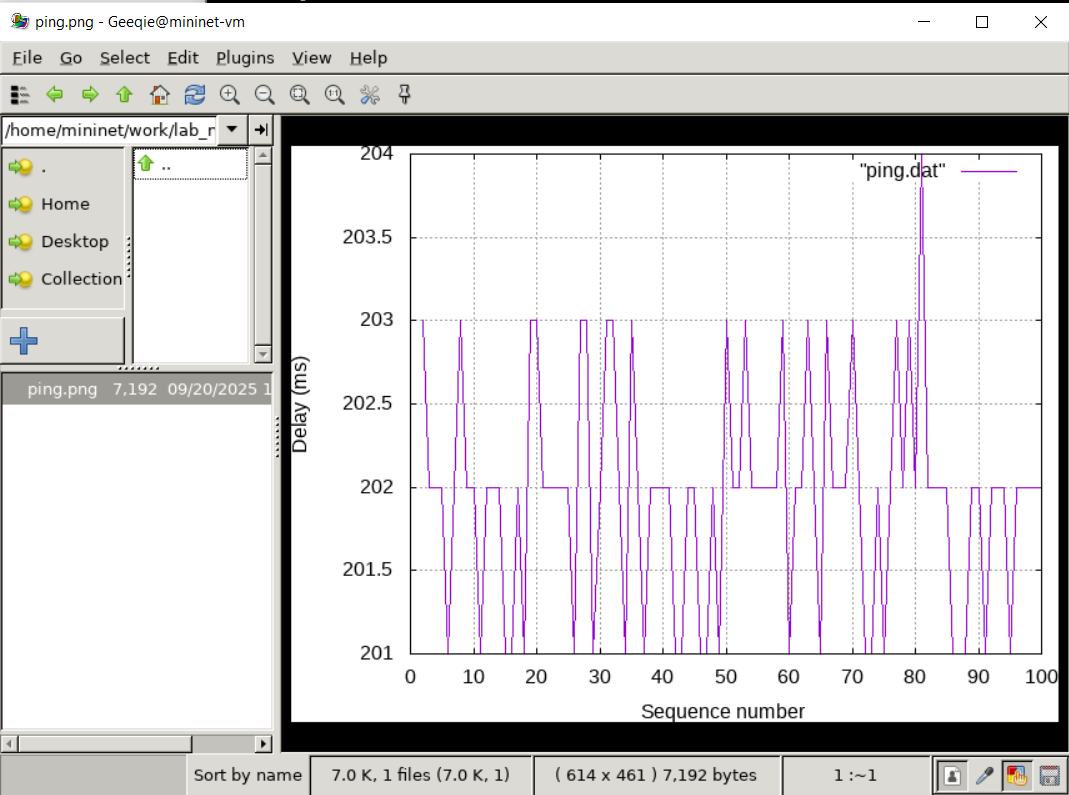
Визуализация эксперимента

Из файла ping.dat удалим первую строку (рис. [**fig:018?**]).



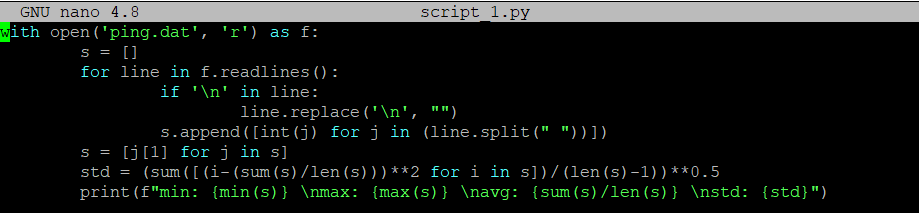
Удаление строки из файла .dat

Из файла ping.dat удалим первую строку и заново построим график (рис. [**fig:019?**]).



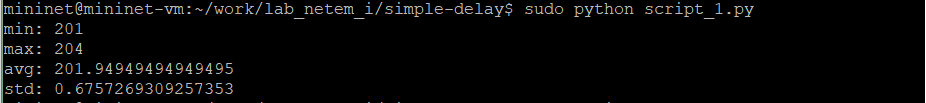
Визуализация эксперимента

Разработаем скрипт для вычисления на основе данных файла ping.dat минимального, среднего, максимального и стандартного отклонения времени приёма-передачи (рис. [**fig:020?**]).



Скрипт script\_1.py

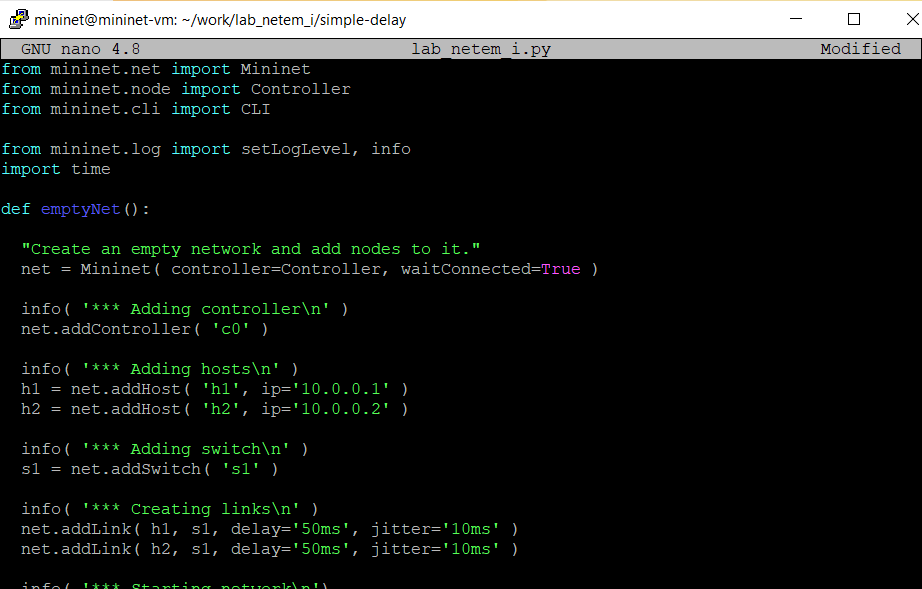
Продемонстрируем работу скрипта с выводом значений на экран или в отдельный файл (рис. [**fig:021?**]).



Результат работы скрипта script\_1.py

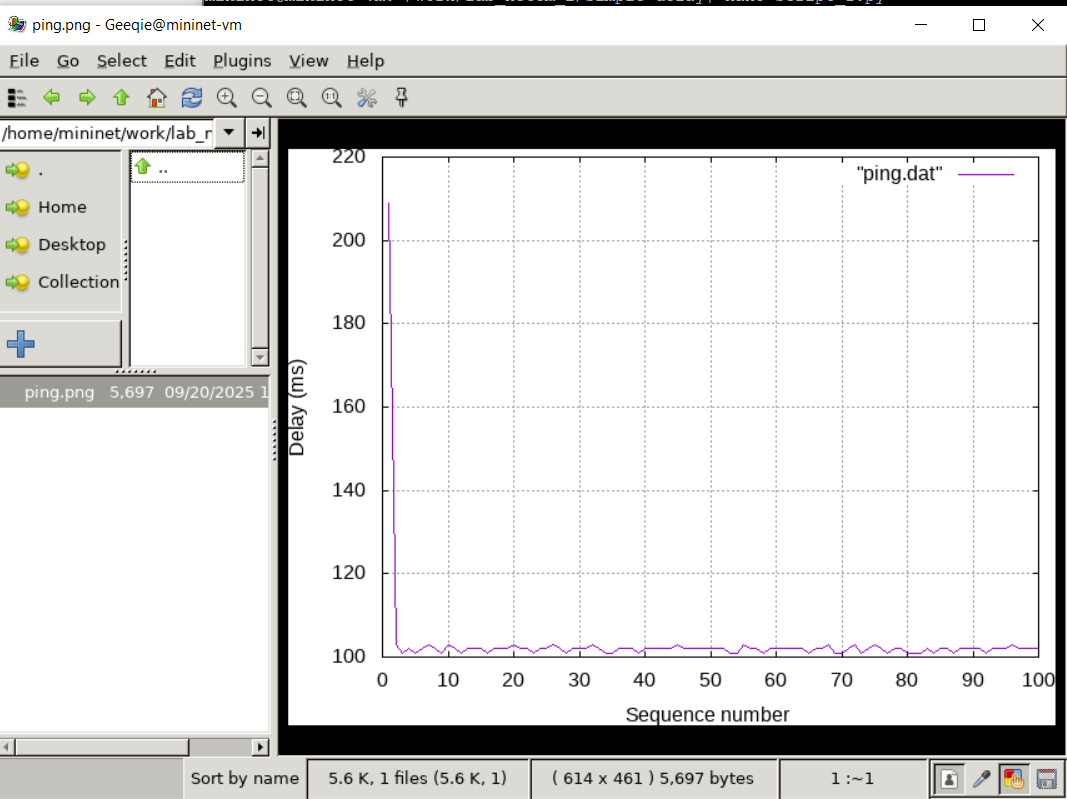
## 3.9 Задание для самостоятельной работы

Самостоятельно реализуем воспроизводимые эксперименты по изменению задержки, джиттера, значения корреляции для джиттера и задержки, распределения времени задержки в эмулируемой глобальной сети (рис. [**fig:022?**]).



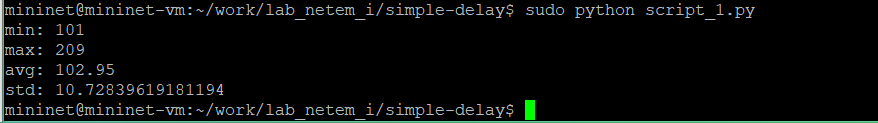
Изменение файла lab\_netem\_i.py

Продемонстрируем построенный в результате выполнения скриптов график (рис. [**fig:023?**]).



Визуализация эксперимента

Вычислим минимальное, среднее, максимальное и стандартное отклонение времени приёма-передачи для каждого случая (рис. [**fig:024?**]).



Результат работы скрипта script\_1.py

# 4 Выводы

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