Отчет по лабораторной работе 4

Петрушов Дмитри1, 1032212287

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 Выполнение лабораторной работы

В виртуальной машине mininet исправим права запуска X-соединения и Зададим простейшую топологию, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8 (рис. [[1](#fig:001)]):

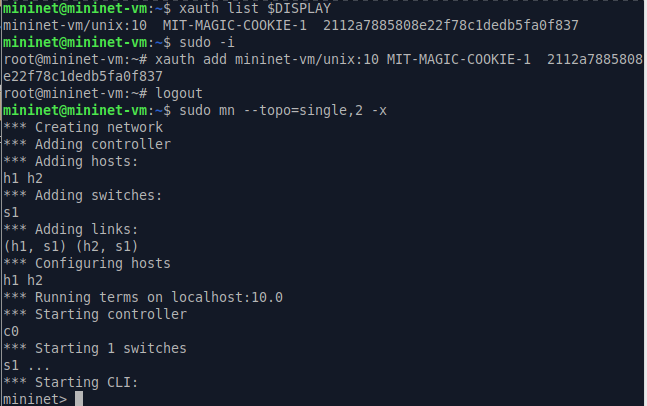


Figure 1: Создание простейшей топологии

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 и Проверим подключение между хостами h1 и h2 с помощью команды ping с параметром -c 6 (рис. [[2](#fig:002)]):

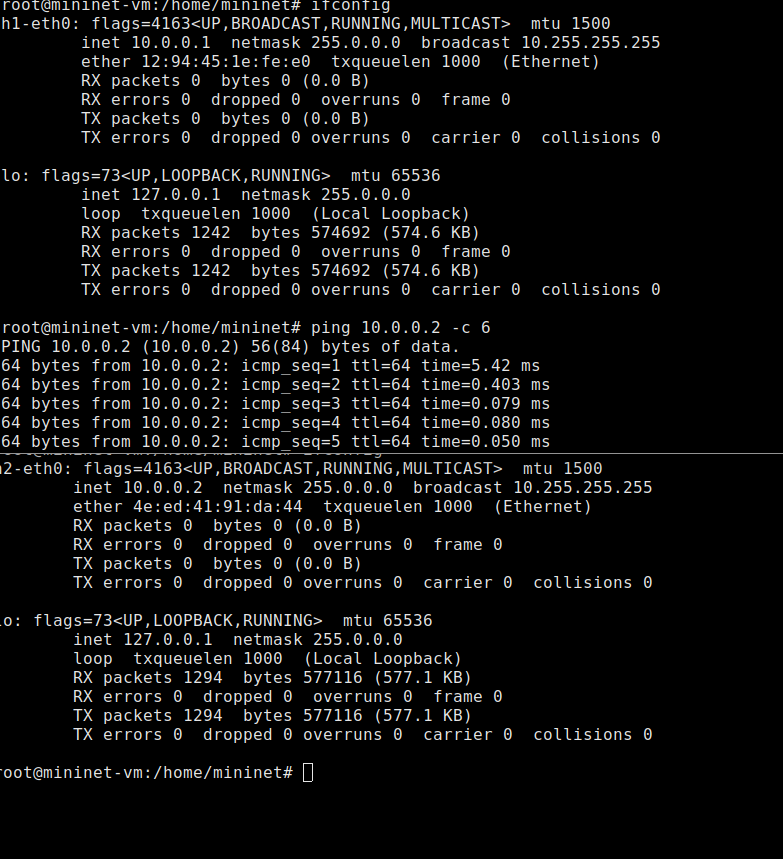


Figure 2: Отображение информации их сетевых интерфейсов и IP-адресов

Проверим подключение между хостами h1 и h2 с помощью команды ping с параметром -c 6 (рис. [[3](#fig:003)]):

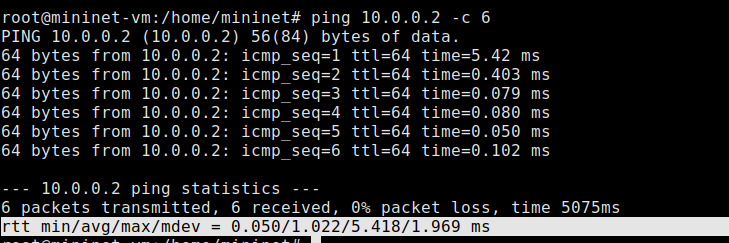


Figure 3: Проверка подключения между хостами h1 и h2

На хосте h1 добавим задержку в 100 мс к выходному интерфейсу (рис. [[4](#fig:004)]):

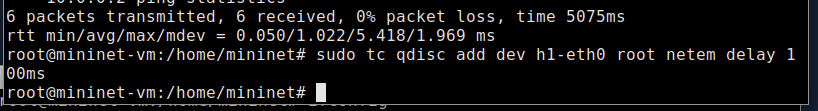


Figure 4: Добавление задержки в 100 мс к выходному интерфейсу на хосте h1

Проверим, что соединение от хоста h1 к хосту h2 имеет задержку 100 мс, используя команду ping с параметром -c 6 с хоста h1 (рис. [[5](#fig:005)]):

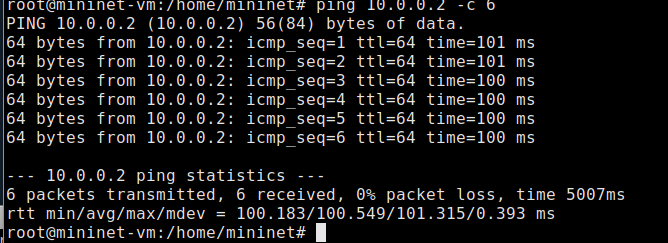


Figure 5: Проверка

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. [[6](#fig:006)]):

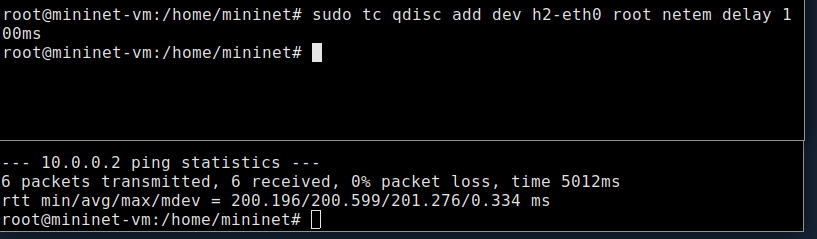


Figure 6: Добавление задержки в 100 мс к выходному интерфейсу на хосте h2 и Проверка

Изменим задержку со 100 мс до 50 мс для отправителя h1 и для получателя h2 (рис. [[7](#fig:007)]):

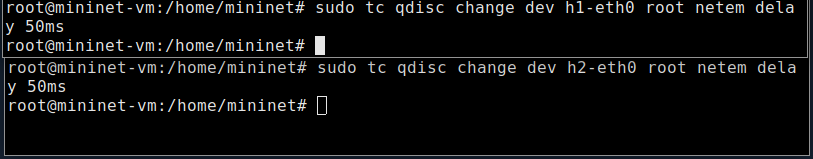


Figure 7: Изменение задержки со 100 мс до 50 мс

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. [[8](#fig:008)]):

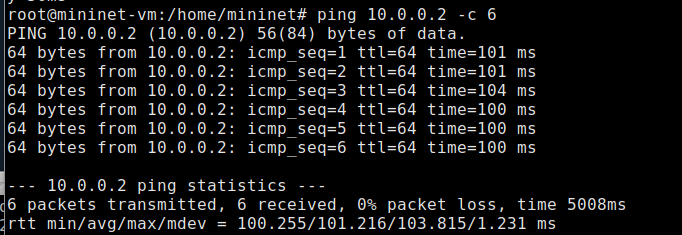


Figure 8: Проверка

Восстановим конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса (рис. [[9](#fig:009)]):

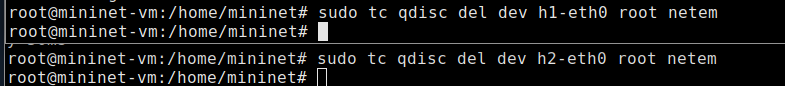


Figure 9: Восстановление конфигураций по умолчанию

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(рис. [[10](#fig:010)]):

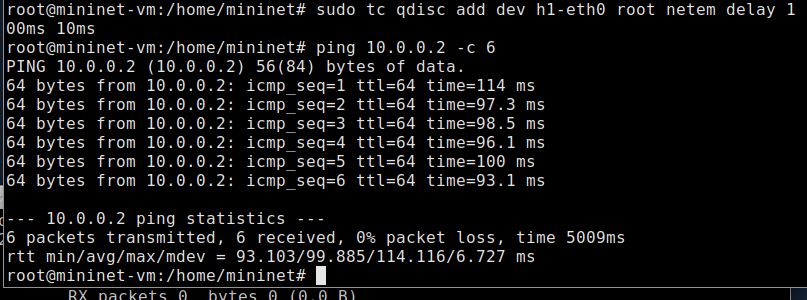


Figure 10: Добавление на узле h1 задержки в 100 мс со случайным отклонением 10 мс и проверка

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. [[11](#fig:011)]):

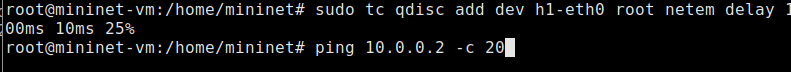


Figure 11: добавление кореляции

Проверим (рис. [[12](#fig:012)]):

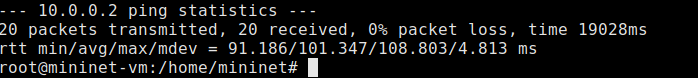


Figure 12: Проверка

Зададим нормальное распределение задержки на узле h1 в эмулируемой сети (рис. [[13](#fig:013)]):

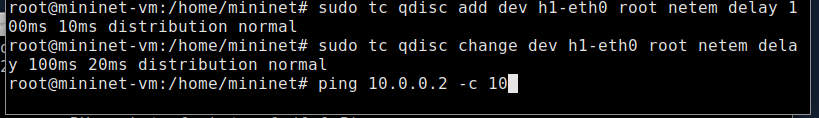


Figure 13: Настройка нормального распределения задержки на узле h1 в эмулируемой сети

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. [[14](#fig:014)]):

Figure 14: Проверка

Figure 14: Проверка

Восстановим конфигурацию интерфейса по умолчанию на узле h1  
Завершим работу mininet в интерактивном режиме  
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. [[15](#fig:015)]):

Figure 15: Создание нового каталога

Figure 15: Создание нового каталога

В виртуальной среде mininet в своём рабочем каталоге с проектами создадим каталог simple-delay и перейдём в него. Создадим скрипт для эксперимента lab\_netem\_i.py. Создадим файл ping\_plot. Затем создадим скрипт для визуализации ping\_plot результатов эксперимента. Зададим права доступа к файлу скрипта. Создадим файла Makefile. нутри файла Makefile поместим скрипт для управления процессом проведения эксперимента (рис. [[16](#fig:016)]):

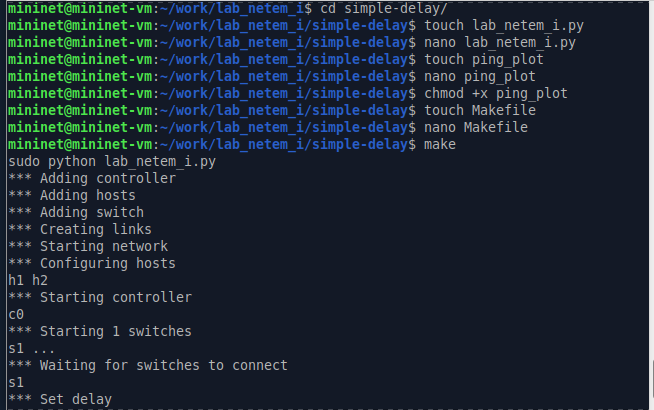


Figure 16: а

Выполним эксперимент и Просмотрим построенный в результате выполнения скриптов график (рис. [[17](#fig:017)]):

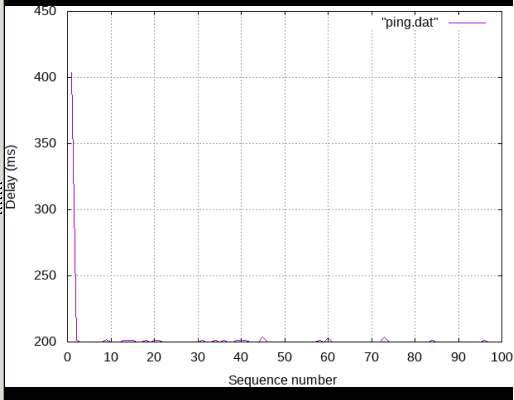


Figure 17: Просмотр графика

Из файла ping.dat удалим первую строку и заново построим график (рис. [[18](#fig:018)]):

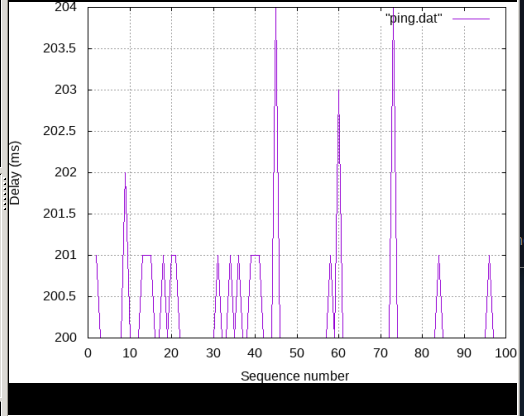


Figure 18: Повторное построение графика

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  
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

# 3 Вывод

В ходе выполнения лабораторной работы познакомились с NETEM — инструментом для тестирования производительности приложений в виртуальной сети, а также получили навыки проведения интерактивного и воспроизводимого экспериментов по измерению задержки и её дрожания (jitter) в моделируемой сети в среде Mininet.

# Список литературы