

Отчет по лабораторной работе №3 Дисциплина: Моделирование сетей передачи данных

Леснухин Даниил Дмитриевич
Российский университет дружбы народов
Москва

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.

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.

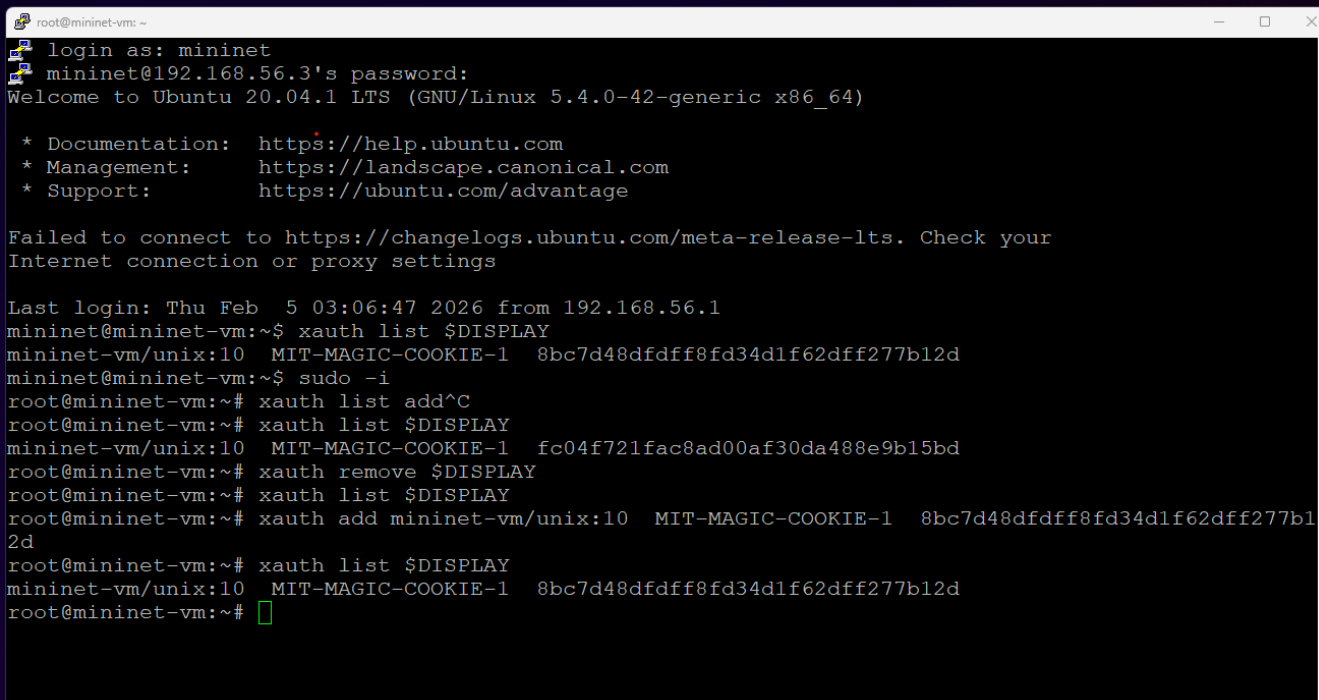
Теоретическое введение

Mininet[@mininet] – 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(VM). 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.

Выполнение лабораторной работы

Запуск лабораторной топологии

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.



```
root@mininet-vm: ~  
login as: mininet  
mininet@192.168.56.3's password:  
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-42-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your  
Internet connection or proxy settings  
  
Last login: Thu Feb  5 03:06:47 2026 from 192.168.56.1  
mininet@mininet-vm:~$ xauth list $DISPLAY  
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 8bc7d48dfdf8fd34d1f62dff277b12d  
mininet@mininet-vm:~$ sudo -i  
root@mininet-vm:~# xauth list add^C  
root@mininet-vm:~# xauth list $DISPLAY  
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 fc04f721fac8ad00af30da488e9b15bd  
root@mininet-vm:~# xauth remove $DISPLAY  
root@mininet-vm:~# xauth list $DISPLAY  
root@mininet-vm:~# xauth add mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 8bc7d48dfdf8fd34d1f62dff277b12d  
root@mininet-vm:~# xauth list $DISPLAY  
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 8bc7d48dfdf8fd34d1f62dff277b12d  
root@mininet-vm:~#
```

Рис. 1: Копируем значения куки

Зададим простейшую топологию, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8

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`

Проверим подключение между хостами h1 и h2 с помощью команды `ping` с параметром `-c 6`

```
root@mininet-vm:/home/mininet# ifconfig
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.2 netmask 255.0.0.0 broadcast 10.255.255.255
    ether aa:51:e1:44:c3:6b txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 1170 bytes 254296 (254.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1170 bytes 254296 (254.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet#

root@mininet-vm:/home/mininet# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.1 netmask 255.0.0.0 broadcast 10.255.255.255
    ether 86:10:be:4d:89:f4 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 1091 bytes 247224 (247.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1091 bytes 247224 (247.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet#
```

Рис. 2: Простейшая топология два хоста

```
"host: h2"@mininet-vm
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 1170 bytes 254296 (254.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1170 bytes 254296 (254.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet# ping -c 6 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=1.84 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.042 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.042 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.044 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.134 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.046 ms

--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5105ms
rtt min/avg/max/mdev = 0.042/0.357/1.835/0.661 ms
root@mininet-vm:/home/mininet#
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.118 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.042 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.033 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5094ms
rtt min/avg/max/mdev = 0.033/0.422/2.097/0.750 ms
root@mininet-vm:/home/mininet# ping -c 6 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.084 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.038 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.067 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.046 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.043 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5117ms
rtt min/avg/max/mdev = 0.038/0.053/0.084/0.016 ms
root@mininet-vm:/home/mininet#
```

Интерактивные эксперименты

Добавление/изменение задержки в эмулируемой глобальной сети

Сетевые эмуляторы задают задержки на интерфейсе. Например, задержка, вносимая в интерфейс коммутатора А, который подключён к интерфейсу коммутатора В, может представлять собой задержку распространения WAN, соединяющей оба коммутатора.

На хосте h1 добавим задержку в 100 мс к выходному интерфейсу.

```
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

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms
root@mininet-vm:/home/mininet#
```

Рис. 3: Установка задержки 100ms

```
root@mininet-vm:/home/mininet# ping -c 6 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=100 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=100 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=100 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=100 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 100.124/100.422/100.743/0.232 ms
root@mininet-vm:/home/mininet#
```

Рис. 4: Проверка задержки 100ms

Для эмуляции глобальной сети с двунаправленной задержкой необходимо к соответствующему интерфейсу на хосте h2 также добавим 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. Минимальное - (202.068), среднее - (202.884), максимальное - (204.777), стандартное отклонение времени (0.943) приема-передачи.

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. Минимальное - (101.050), среднее - (101.253), максимальное - (108.223), стандартное отклонение времени (2.475) приема-передачи.

![Проверка задержки 100ms 2 x 50ms]](7.png)

```

root@mininet-vm:/home/mininet# sudo tc qdisc add dev h2-eth0 root netem delay 100ms
root@mininet-vm:/home/mininet# 
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=100 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=100 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 100.124/100.422/100.743/0.232 ms
root@mininet-vm:/home/mininet# ping -c 6 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=202 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=201 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=200 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=201 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=200 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=201 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5011ms
rtt min/avg/max/mdev = 200.489/200.830/201.697/0.408 ms
root@mininet-vm:/home/mininet# 

```

Рис. 5: Проверка задержки 200ms

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: `sudo tc qdisc del dev h1-eth0 root netem`. Для получателя h2: `sudo tc qdisc del dev h2-eth0 root netem`. 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. Минимальное - (0.145), среднее - (1.256), максимальное - (4.875), стандартное отклонение времени (1.655) приема-передачи.

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. Минимальное - (91.961), среднее - (99.758), максимальное - (106.319), стандартное отклонение времени (5.130) приема-передачи

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

```

root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
root@mininet-vm:/home/mininet# 
tt min/avg/max/mdev = 100.184/100.728/101.197/0.361 ms
root@mininet-vm:/home/mininet# 
root@mininet-vm:/home/mininet# 
root@mininet-vm:/home/mininet# 
root@mininet-vm:/home/mininet# 
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
root@mininet-vm:/home/mininet# ping -c 6 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
4 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.83 ms
4 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.481 ms
4 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.161 ms
4 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.048 ms
4 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.053 ms
4 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.040 ms

-- 10.0.0.2 ping statistics --
6 packets transmitted, 6 received, 0% packet loss, time 5077ms
tt min/avg/max/mdev = 0.040/0.435/1.828/0.641 ms
root@mininet-vm:/home/mininet# 

```

Рис. 6: Восстановление значений по умолчанию

```

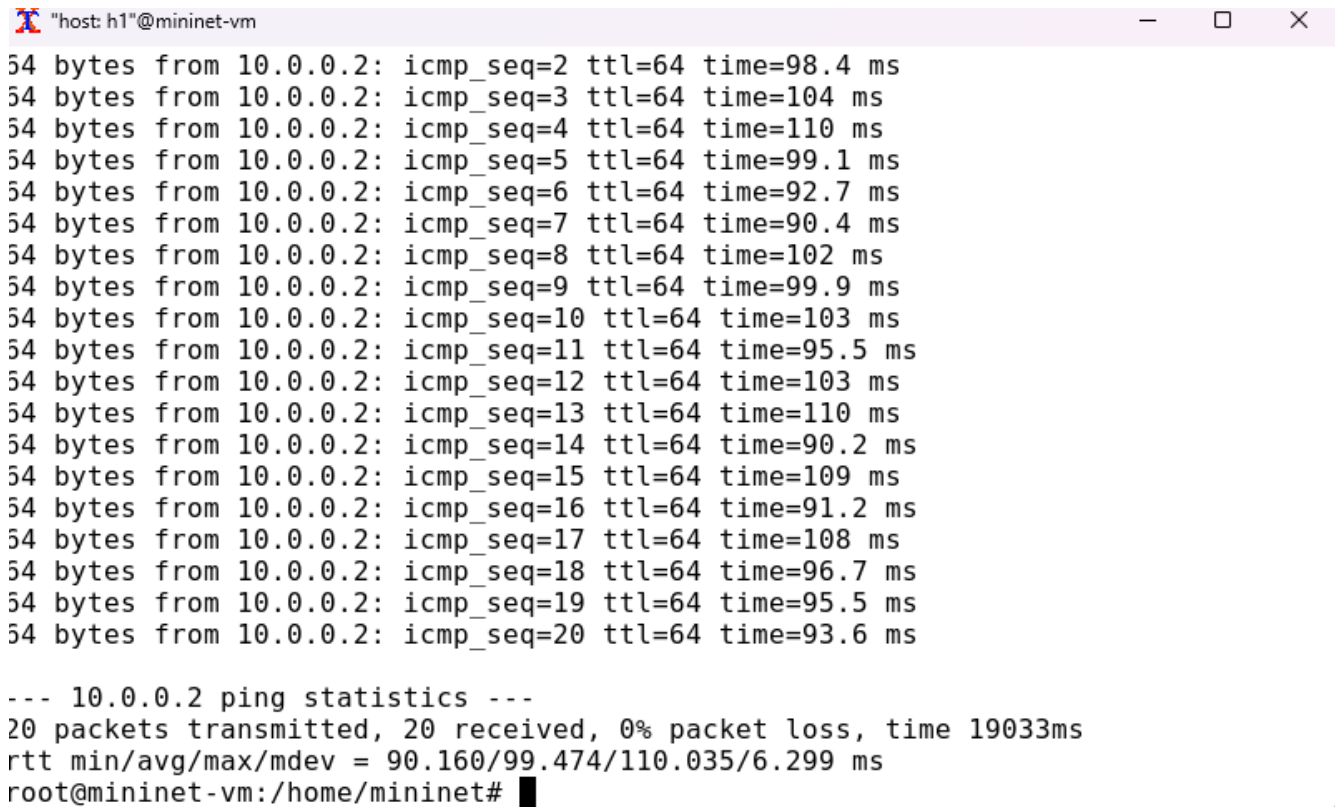
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 1
00ms 10ms
root@mininet-vm:/home/mininet# ping -c 6 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=103 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=96.3 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=104 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=91.6 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=93.1 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=92.9 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5010ms
rtt min/avg/max/mdev = 91.591/96.824/103.618/4.927 ms
root@mininet-vm:/home/mininet# 

```

Рис. 7: Добавление значения дрожания задержки в интерфейс подключения

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. Минимальное - (91.887), среднее - (102.565), максимальное - (111.050), стандартное отклонение времени (5.730) приема-передачи

A terminal window titled "host: h1@mininet-vm" displays the output of a ping command. It shows 20 individual ping results, each with 54 bytes, a TTL of 64, and a time in milliseconds. The times range from approximately 90.2 ms to 110 ms. Below the individual results, a summary line reads "10.0.0.2 ping statistics ---". This is followed by statistics for 20 packets: 20 transmitted, 20 received, 0% packet loss, and a total time of 19033ms. The round-trip time (RTT) statistics are listed as min/avg/max/mdev = 90.160/99.474/110.035/6.299 ms. The prompt at the bottom is root@mininet-vm:/home/mininet#.

```
host: h1@mininet-vm
54 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=98.4 ms
54 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=104 ms
54 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=110 ms
54 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=99.1 ms
54 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=92.7 ms
54 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=90.4 ms
54 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=102 ms
54 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=99.9 ms
54 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=103 ms
54 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=95.5 ms
54 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=103 ms
54 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=110 ms
54 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=90.2 ms
54 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=109 ms
54 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=91.2 ms
54 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=108 ms
54 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=96.7 ms
54 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=95.5 ms
54 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=93.6 ms

--- 10.0.0.2 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19033ms
rtt min/avg/max/mdev = 90.160/99.474/110.035/6.299 ms
root@mininet-vm:/home/mininet#
```

Рис. 8: 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 \pm 20 мс. Используем для этого команду `ping` на терминале хоста h1 с параметром `-c 10`. Восстановим конфигурацию интерфейса по умолчанию на узле h1. Минимальное - (74.585), среднее - (105.367), максимальное - (126.328), стандартное отклонение времени (16.340) приема-передачи. Завершим работу mininet в интерактивном режиме

Воспроизведение экспериментов

Предварительная подготовка

Обновим репозитории программного обеспечения на виртуальной машине: `sudo apt-get update`. Установим пакет `geeie` — понадобится для просмотра файлов `png`: `sudo apt install geeie`.

Для каждого воспроизводимого эксперимента `exprname` создадим свой каталог, в котором будут размещаться файлы эксперимента: `mkdir -p ~/work/lab_netem_i/exprname`. Здесь `exprname` может принимать значения `simple-delay`, `change-delay`, `jitter-delay`, `correlation-delay` и т.п. Для каждого


```

root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 1
00ms 20ms distribution normal
root@mininet-vm:/home/mininet# ping -c 10 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=125 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=113 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=82.5 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=115 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=103 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=98.8 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=96.8 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=95.1 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=78.0 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=116 ms

--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9015ms
rtt min/avg/max/mdev = 77.957/102.259/124.701/14.327 ms
root@mininet-vm:/home/mininet# █

```

Рис. 9: Распределение задержки в интерфейсе подключения

случая создадим скрипт для проведения эксперимента `lab_netem_i.py` и скрипт для визуализации результатов `ping_plot`.

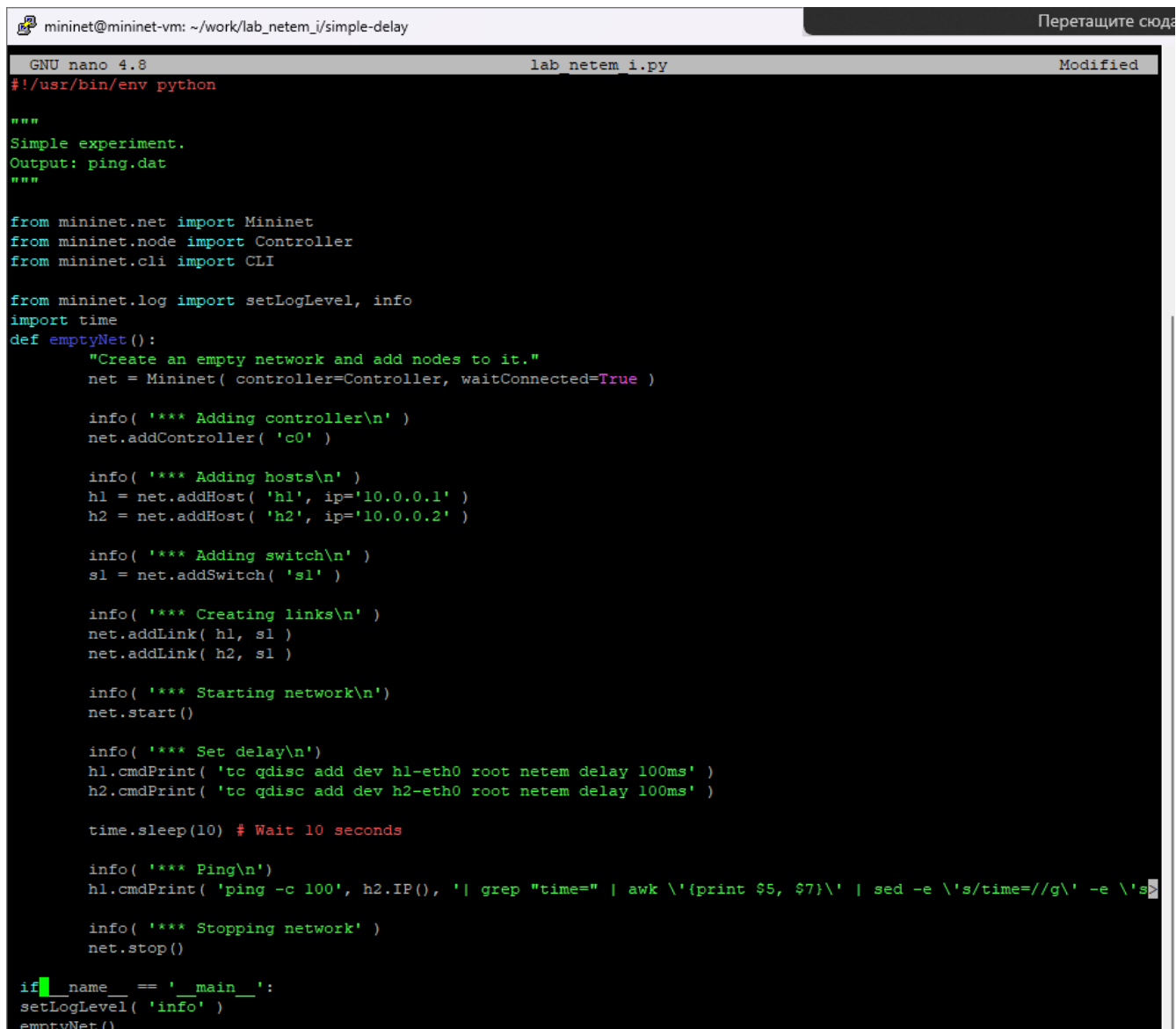
Добавление задержки для интерфейса, подключающегося к эмулируемой глобальной сети

С помощью API Mininet воспроизведем эксперимент по добавлению задержки для интерфейса хоста, подключающегося к эмулируемой глобальной сети.

В виртуальной среде mininet в своём рабочем каталоге с проектами создадим каталог `simple-delay` и перейдем в него.

```
mkdir -p ~/work/lab_netem_i/simple-delay cd ~/work/lab_netem_i/simple-delay
```

Создадим скрипт для эксперимента `lab_netem_i.py`.



```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
GNU nano 4.8 lab_netem_i.py Modified
#!/usr/bin/env python

"""
Simple experiment.
Output: ping.dat
"""

from mininet.net import Mininet
from mininet.node import Controller
from mininet.cli import CLI

from mininet.log import setLogLevel, info
import time

def emptyNet():
    "Create an empty network and add nodes to it."
    net = Mininet( controller=Controller, waitConnected=True )

    info( '*** Adding controller\n' )
    net.addController( 'c0' )

    info( '*** Adding hosts\n' )
    h1 = net.addHost( 'h1', ip='10.0.0.1' )
    h2 = net.addHost( 'h2', ip='10.0.0.2' )

    info( '*** Adding switch\n' )
    s1 = net.addSwitch( 's1' )

    info( '*** Creating links\n' )
    net.addLink( h1, s1 )
    net.addLink( h2, s1 )

    info( '*** Starting network\n' )
    net.start()

    info( '*** Set delay\n' )
    h1.cmdPrint( 'tc qdisc add dev h1-eth0 root netem delay 100ms' )
    h2.cmdPrint( 'tc qdisc add dev h2-eth0 root netem delay 100ms' )

    time.sleep(10) # Wait 10 seconds

    info( '*** Ping\n' )
    h1.cmdPrint( 'ping -c 100', h2.IP(), '| grep "time=" | awk \'{print $5, $7}\'} | sed -e \'/s/time=//g\' -e \'/s/ /&/' )

    info( '*** Stopping network' )
    net.stop()

if __name__ == '__main__':
    setLogLevel( 'info' )
    emptyNet()
```

Рис. 10: Изменение скрипта

```
mininet@mininet-vm: ~/work/lab_netem_i/
GNU nano 4.8
all: ping.dat ping.png

ping.dat:
    sudo python lab_netem_i.
    sudo chown mininet:mininet

ping.png: ping.dat
    ./ping_plot

clean:
    rm -f *.dat *.png
```

Создадим скрипт для визуализации `ping_plot` результатов эксперимента

Зададим права доступа к файлу скрипта: `chmod +x ping_plot`

Выполним эксперимент. Продемонстрируем построенный в результате выполнения скриптов график

Из файла `ping.dat` удалим первую строку и заново постройте график. Продемонстрируем построен-

```

mininet@mininet-vm:~/work/lab_netem_i/simple-delaymake
sudo python lab_netem_i.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
*** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
*** Set delay
*** h1 : ('tc qdisc add dev h1-eth0 root netem delay 100ms',)
*** h2 : ('tc qdisc add dev h2-eth0 root netem delay 100ms',)
*** Ping
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "time=" | awk \'{print $5, $7}\''
sed -e \'/s/time=//g\' -e \'/s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ █

```

Рис. 11: Создание каталогов, права к файлу скрипта

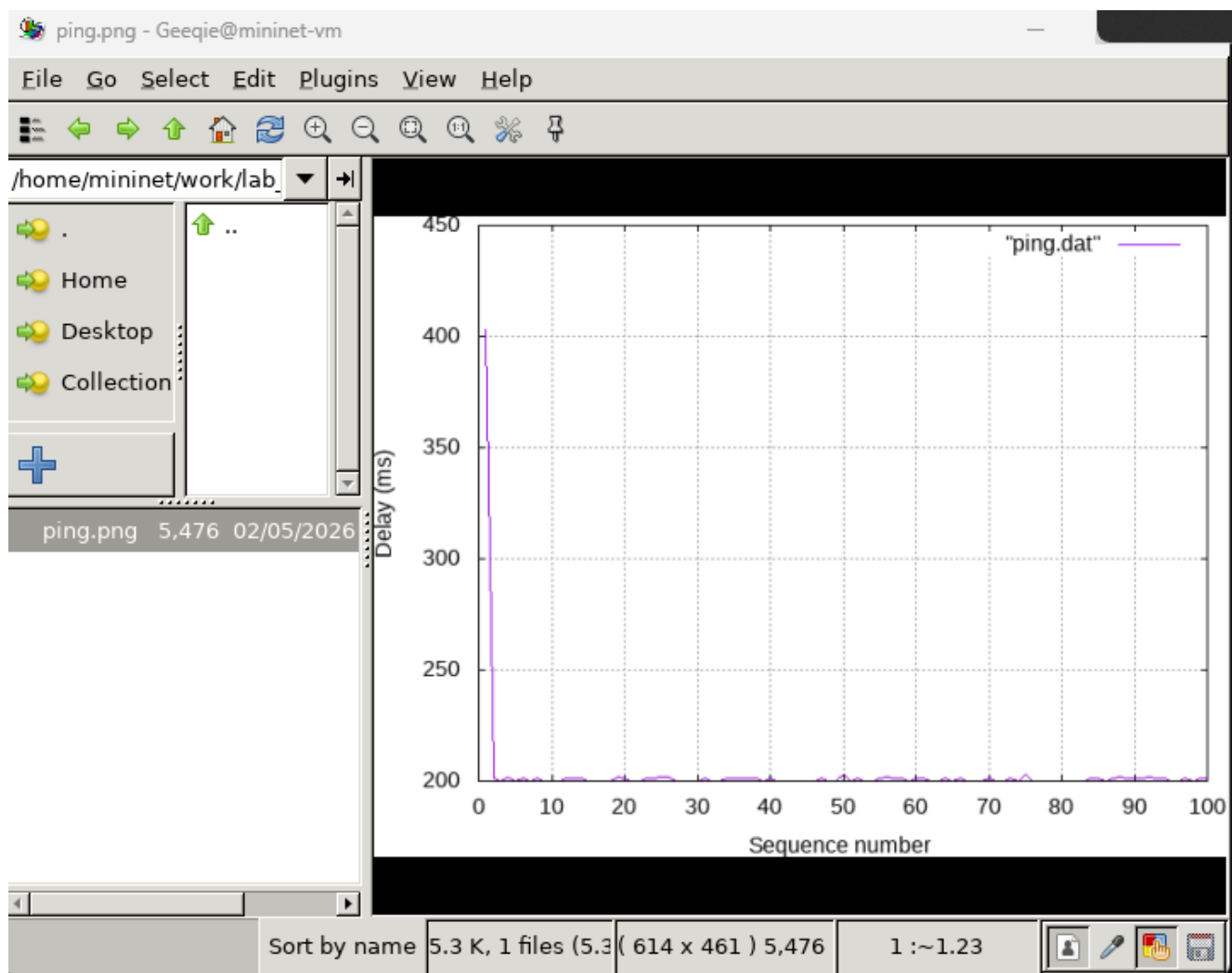
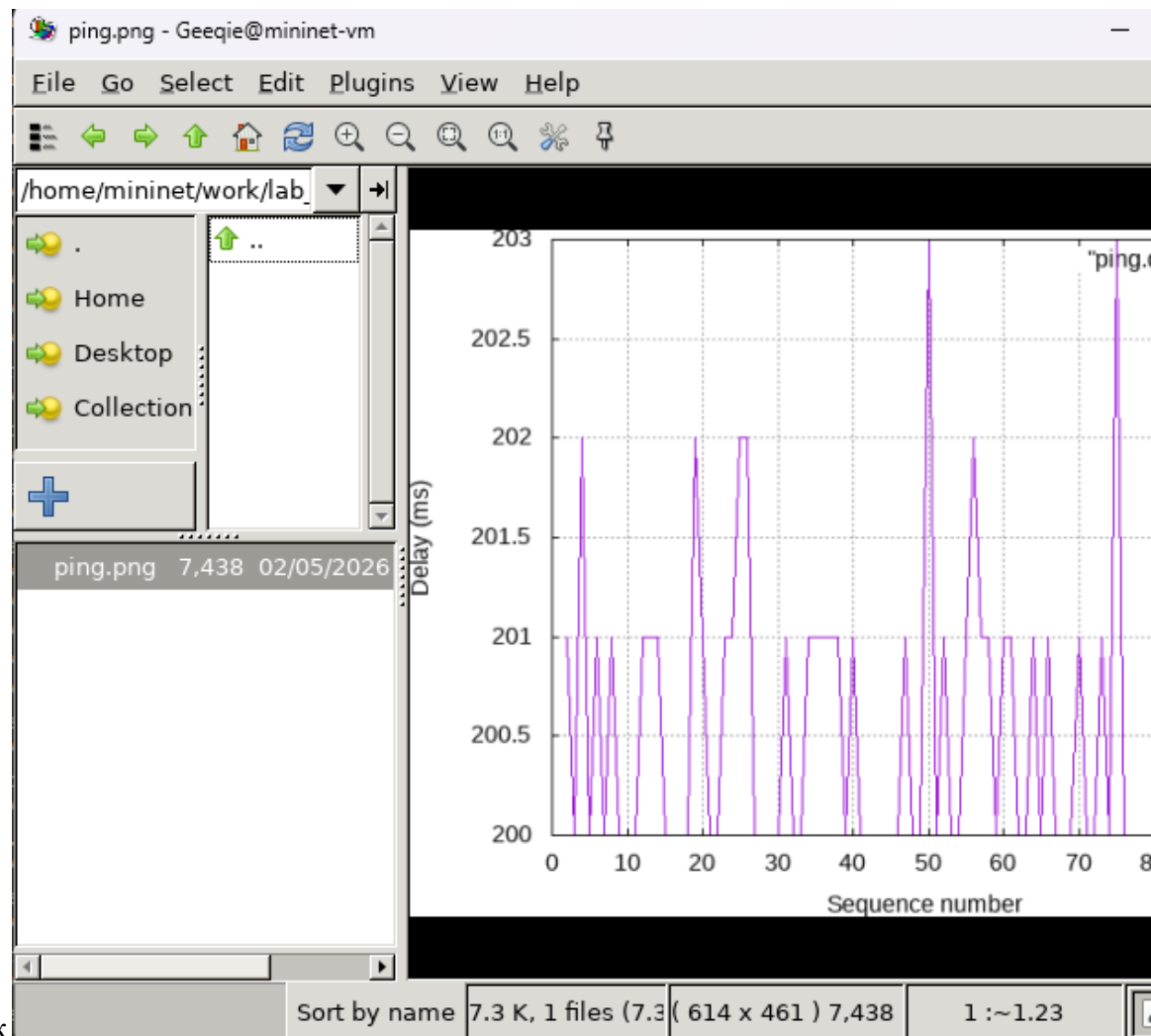


Рис. 12: Результат выполнения скрипта



ный в результате график

Разработаем скрипт для вычисления на основе данных файла ping.dat минимального, среднего,

```
mininet@mininet-vm: ~/work/lab_netem_i/simple-delay
GNU nano 4.8
with open('ping.dat', 'r') as f:
    s = []
    for line in f.readlines():
        if '\n' in line:
            line.replace('\n', ' ')
            s.append([int(j) for j in line.split()])
    s = [j[1] for j in s]
    std = (sum([(i - (sum(s)/len(s))**2) for i in s])**len(s))**0.5
    print(f"min: {min(s)} \max: {max(s)} \std: {std}")
```

максимального и стандартного отклонения времени приёма-передачи

Выводы

В результате выполнения данной лабораторной работы я познакомился с NETEM – инструментом для тестирования производительности приложений в виртуальной сети, а также получил навыки проведения интерактивного и воспроизводимого экспериментов по измерению задержки и её дро-

жания (jitter) в моделируемой сети в среде Mininet.