

Лабораторная работа № 4

Моделирование сетей передачи данных

Доберштейн Алина Сергеевна

Содержание

1 Цель работы	5
2 Задание	6
3 Теоретическое введение	7
4 Выполнение лабораторной работы	8
4.1 Интерактивные эксперименты	10
4.1.1 Добавление задержки в эмулируемой глобальной сети	10
4.1.2 Изменение задержки в эмулируемой глобальной сети	10
4.1.3 Восстановление исходных значений (удаление правил) задержки в эмулируемой глобальной сети	11
4.1.4 Добавление значения дрожания задержки в интерфейс подключения к эмулируемой глобальной сети	12
4.1.5 Добавление значения корреляции для джиттера и задержки в интерфейс подключения к эмулируемой глобальной сети	12
4.1.6 Распределение задержки в интерфейсе подключения к эмулируемой глобальной сети	13
4.2 Воспроизведение экспериментов	14
4.2.1 Предварительная подготовка	14
4.2.2 Добавление задержки для интерфейса, подключающегося к эмулируемой глобальной сети	14
4.2.3 Воспроизводимый эксперимент по изменению задержки	18
4.2.4 Воспроизводимый эксперимент по изменению джиттера	20
4.2.5 Воспроизводимый эксперимент по изменению значения корреляции джиттера и задержки	22
4.2.6 Воспроизводимый эксперимент по изменению распределения времени задержки в эмулируемой глобальной сети	24
5 Выводы	27
Список литературы	28

Список иллюстраций

4.1	Проверка прав запуска X-соединения	8
4.2	Запуск топологии	9
4.3	Проверка связи между хостами	9
4.4	Проверка связи между хостами	10
4.5	Проверка связи между хостами	10
4.6	Проверка связи между хостами	11
4.7	Проверка связи между хостами	11
4.8	Проверка связи между хостами	12
4.9	Проверка связи между хостами	12
4.10	Проверка связи между хостами	13
4.11	Проверка связи между хостами	14
4.12	Обновление репозиториев	14
4.13	Создание каталога для эксперимента	15
4.14	Скрипт для эксперимента	15
4.15	Скрипт для визуализации эксперимента	15
4.16	Права доступа	16
4.17	Makefile	16
4.18	Выполнение эксперимента	16
4.19	График	17
4.20	Скрипт для вычисления rtt	17
4.21	Вычисленное rtt	17
4.22	Создание каталога для эксперимента	18
4.23	Файлы для эксперимента	18
4.24	Скрипт для эксперимента	18
4.25	Выполнение эксперимента	19
4.26	График	19
4.27	Создание каталога для эксперимента	20
4.28	Скрипт для эксперимента	20
4.29	Выполнение эксперимента	21
4.30	График	21
4.31	Вычисленное rtt	22
4.32	Создание каталога для эксперимента	22
4.33	Скрипт для эксперимента	23
4.34	Выполнение эксперимента	23
4.35	График	24
4.36	Создание каталога для эксперимента	24
4.37	Скрипт для эксперимента	25

4.38 Выполнение эксперимента	25
4.39 График	26

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 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 подобавлению/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 Теоретическое введение

NETEM—сетевой эмулятор Linux, используемый для тестирования производительности реальных клиент-серверных приложений в виртуальной сети. Виртуальная сеть в данном случае представляет собой лабораторную среду для воспроизведения поведения глобальной сети (Wide Area Network, WAN). NETEM позволяет пользователю задать ряд параметров сети, например, задержку, дрожание задержки (jitter), уровень потери пакетов, дублирование и изменение порядка пакетов.

4 Выполнение лабораторной работы

Запустила виртуальную среду с mininet, пожключилась из основной ОС, проверила права запуска X-соединения (рис. 4.1).

```
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ xauth list $DISPLAY
mininet-vm/unix:12 MIT-MAGIC-COOKIE-1 9713b86bb098fbfdb897dd6199ab032e
mininet@mininet-vm:~/work/lab_iperf3/iperf3$ sudo -i
root@mininet-vm:~# xauth list $DISPLAY
root@mininet-vm:~# xauth add mininet-vm/unix:12 MIT-MAGIC-COOKIE-1 9713b86bb098fbfdb897dd6199ab032e
xfrq@xauth: command not found
root@mininet-vm:~# xauth add mininet-vm/unix:12 MIT-MAGIC-COOKIE-1 9713b86bb098fbfdb897dd6199ab032e
xfrq@xauth: command not found
root@mininet-vm:~# xauth add mininet-vm/unix:12 MIT-MAGIC-COOKIE-1 9713b86bb098fbfdb897dd6199ab032e
xauth: command not found
root@mininet-vm:~# xauth add mininet-vm/unix:12 MIT-MAGIC-COOKIE-1 9713b86bb098fbfdb897dd6199ab032e
Command 'xfrq@xauth' not found, did you mean:
  command 'xauth' from deb xauth (1:1.1-0ubuntu1)
Try: apt install <deb name>
root@mininet-vm:~# xauth add mininet-vm/unix:12 MIT-MAGIC-COOKIE-1 9713b86bb098fbfdb897dd6199ab032e
root@mininet-vm:~# xauth list $DISPLAY
mininet-vm/unix:12 MIT-MAGIC-COOKIE-1 9713b86bb098fbfdb897dd6199ab032e
root@mininet-vm:~# logout
```

Рис. 4.1: Проверка прав запуска X-соединения

Задала простейшую топологию, состояющую из двух хостов и коммутатора. (рис. 4.2).

```

mininet@mininet-vm:/work/lab_iperf3/iperf3$ sudo mn --topo=single,2 -x
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Running terms on localhost:12.0
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 8 terms
*** Stopping 2 links
...
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 1677.753 seconds

```

Рис. 4.2: Запуск топологии

На хостах вывела команду ifconfig, проверила подключение между хостами.(рис. 4.3).

```

root@mininet-vm:/home/mininet/work/lab_iperf3/iperf3# 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 le:c1:9f:fd:a8:ef  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 889  bytes 252964 (252.9 KB)
          RX errors 0  dropped 0  overruns 0  frame 0
          TX packets 889  bytes 252964 (252.9 KB)
          TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

root@mininet-vm:/home/mininet/work/lab_iperf3/iperf3# 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=4.43 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.407 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.063 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.061 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.080 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.070 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5092ms
rtt min/avg/max/mdev = 0.061/0.851/4.426/1.603 ms
root@mininet-vm:/home/mininet/work/lab_iperf3/iperf3# []

```

Рис. 4.3: Проверка связи между хостами

минимальное отклонение времени приёма-передачи (RTT) - 0.061 ms, среднее - 0.851 ms,

максимальное - 4.426 ms и стандартное - 1.603 ms

4.1 Интерактивные эксперименты

4.1.1 Добавление задержки в эмулируемой глобальной сети

На хосте h1 добавила задержку в 100 мс к выходному интерфейсу, проверила соединение от h1 к h2(рис. 4.4).

```
root@mininet-vm:/home/mininet/work/lab_iperf3/iperf3# sudo tc qdisc add dev h1-  
eth0 root netem delay 100ms  
root@mininet-vm:/home/mininet/work/lab_iperf3/iperf3# 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=101 ms  
64 bytes from 10.0.0.2: icmp seq=3 ttl=64 time=101 ms  
64 bytes from 10.0.0.2: icmp seq=4 ttl=64 time=100 ms  
64 bytes from 10.0.0.2: icmp seq=5 ttl=64 time=101 ms  
64 bytes from 10.0.0.2: icmp seq=6 ttl=64 time=101 ms  
  
--- 10.0.0.2 ping statistics ---  
6 packets transmitted, 6 received, 0% packet loss, time 5003ms  
rtt min/avg/max/mdev = 100.256/101.190/103.270/0.971 ms  
root@mininet-vm:/home/mininet/work/lab_iperf3/iperf3#
```

Рис. 4.4: Проверка связи между хостами

минимальное отклонение времени приёма-передачи (RTT) - 100.256 ms, среднее - 101.190 ms, максимальное - 103.270 ms и стандартное - 0.971 ms

Для эмуляции глобальной сети с двунаправленной задержкой добавила задержку в 100 мс на h2. Проверила связь между хостами (рис. 4.5).

```
root@mininet-vm:/home/mininet/work/lab_iperf3/iperf3# 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=203 ms  
64 bytes from 10.0.0.2: icmp seq=3 ttl=64 time=203 ms  
64 bytes from 10.0.0.2: icmp seq=4 ttl=64 time=202 ms  
64 bytes from 10.0.0.2: icmp seq=5 ttl=64 time=202 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 5007ms  
rtt min/avg/max/mdev = 201.260/202.136/203.114/0.673 ms  
root@mininet-vm:/home/mininet/work/lab_iperf3/iperf3#
```

Рис. 4.5: Проверка связи между хостами

минимальное отклонение времени приёма-передачи (RTT) - 201.260 ms, среднее - 202.136 ms, максимальное - 203.114 ms и стандартное - 0.673 ms

4.1.2 Изменение задержки в эмулируемой глобальной сети

Изменила задержку со 100мс до 50мс для отправителя h1 и получателя h2. Проверила связь между хостами (рис. 4.6).

```

root@mininet-vm:/home/mininet/work/lab_iperf3/iperf3# sudo tc qdisc change dev
h1-eth0 root netem delay 50ms
root@mininet-vm:/home/mininet/work/lab_iperf3/iperf3# 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=102 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=102 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=101 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=103 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=102 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5006ms
rtt min/avg/max/mdev = 100.731/101.756/102.840/0.662 ms
root@mininet-vm:/home/mininet/work/lab_iperf3/iperf3# █

```

Рис. 4.6: Проверка связи между хостами

минимальное отклонение времени приёма-передачи (RTT) - 100.731 ms, среднее - 102.840 ms, максимальное - 102.840 ms и стандартное - 0.662 ms

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

Восстановила конфигурацию по умолчанию, удалив все правила. Проверила связь между хостами (рис. 4.7).

```

root@mininet-vm:/home/mininet/work/lab_iperf3/iperf3# sudo tc qdisc del dev h1-
eth0 root netem
root@mininet-vm:/home/mininet/work/lab_iperf3/iperf3# 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=0.350 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.076 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.074 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.074 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.074 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.065 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5079ms
rtt min/avg/max/mdev = 0.065/0.118/0.350/0.103 ms
root@mininet-vm:/home/mininet/work/lab_iperf3/iperf3# █

```

Рис. 4.7: Проверка связи между хостами

минимальное отклонение времени приёма-передачи (RTT) - 0.065 ms, среднее - 0.118 ms, максимальное - 0.350 ms и стандартное - 0.103 ms

4.1.4 Добавление значения дрожания задержки в интерфейс подключения к эмулируемой глобальной сети

В сетях нет постоянной задержки. На узле h1 добавила задержку 100 мс со случайным отклонением 10 мс. Проверила связь между хостами (рис. 4.8).

```
root@mininet-vm:/home/mininet/work/lab_iperf3/iperf3# sudo tc qdisc add dev h1-eth0 root netem delay 100ms 10ms
root@mininet-vm:/home/mininet/work/lab_iperf3/iperf3# 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=96.8 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=99.2 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=108 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=94.2 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=92.9 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 92.902/98.491/107.684/4.848 ms
root@mininet-vm:/home/mininet/work/lab_iperf3/iperf3# █
root@mininet-vm:~# xauth list | DISPLAY
```

Рис. 4.8: Проверка связи между хостами

минимальное отклонение времени приёма-передачи (RTT) - 92.902 ms, среднее - 98.491 ms, максимальное - 107.684 ms и стандартное - 4.848 ms

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

Добавила на интерфейсе h1 задержку в 100мс с вариацией +-10 мс и значением корреляции 25%. Проверила связь между хостами (рис. 4.9-4.10).

```
root@mininet-vm:/home/mininet/work/lab_iperf3/iperf3# sudo tc qdisc add dev h1-eth0 root netem delay 100ms 25% 25%
root@mininet-vm:/home/mininet/work/lab_iperf3/iperf3# 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=100 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=98.0 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=94.2 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=108 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=92.0 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=111 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 91.957/100.524/110.646/6.823 ms
```

Рис. 4.9: Проверка связи между хостами

```

root@mininet-vm:/home/mininet/work/lab_iperf3/iperf3# ping -c 20 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=111 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=92.9 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=96.9 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=105 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=107 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=107 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=108 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=98.9 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=94.4 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=103 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=109 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=104 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=99.2 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=106 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=102 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=96.7 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=93.0 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=92.9 ms

--- 10.0.0.2 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19038ms
rtt min/avg/max/mdev = 92.925/101.460/110.800/5.515 ms
root@mininet-vm:/home/mininet/work/lab_iperf3/iperf3# sudo tc qdisc del dev h1-
eth0 root netem

```

Рис. 4.10: Проверка связи между хостами

минимальное отклонение времени приёма-передачи (RTT) - 91.957 ms, среднее - 100.524 ms, максимальное - 110.646 ms и стандартное - 6.823 ms

4.1.6 Распределение задержки в интерфейсе подключения к эмулируемой глобальной сети

Задала нормальное распределение задержки на узле h1 в эмулируемой сети. Убедилась, что все пакеты, покидающие хост h1 на интерфейсе h1-eth0, имеют время задержки, которое распределено в диапазоне 100 мс ±20 мс. Восстановим конфигурацию интерфейса по умолчанию на узле h1. Завершим работу mininet в интерактивном режиме (рис. 4.11).

```

root@mininet-vm:/home/mininet/work/lab_iperf3/iperf3# sudo tc qdisc del dev h1-
eth0 root netem
root@mininet-vm:/home/mininet/work/lab_iperf3/iperf3# sudo tc qdisc add dev h1-
eth0 root netem delay 100ms 20ms distribution normal
root@mininet-vm:/home/mininet/work/lab_iperf3/iperf3# 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=74.0 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=106 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=113 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=86.5 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=80.9 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=141 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=95.4 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=79.7 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=112 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=92.1 ms
...
--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9011ms
rtt min/avg/max/mdev = 73.989/98.125/141.228/19.348 ms
root@mininet-vm:/home/mininet/work/lab_iperf3/iperf3# █

```

Рис. 4.11: Проверка связи между хостами

4.2 Воспроизведение экспериментов

4.2.1 Предварительная подготовка

Обновила репозитории ПО на виртуальной машине, установила пакет geeqie. (рис. 4.12).

```

mininet@mininet-vm:~$ sudo apt-get update
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [128 kB]
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [128 kB]
Reading package lists... Done
E: Release file for http://security.ubuntu.com/ubuntu/dists/focal-security/InRelease is not valid yet (invalid for another 12h 55min 0s). Updates for this repository will not be applied.
E: Release file for http://us.archive.ubuntu.com/ubuntu/dists/focal-updates/InRelease is not valid yet (invalid for another 15h 24min 54s). Updates for this repository will not be applied.
E: Release file for http://us.archive.ubuntu.com/ubuntu/dists/focal-backports/InRelease is not valid yet (invalid for another 13h 4min 31s). Updates for this repository will not be applied.
mininet@mininet-vm:~$ sudo apt install geeqie

```

Рис. 4.12: Обновление репозиториев

4.2.2 Добавление задержки для интерфейса, подключающегося к эмулируемой глобальной сети

С помощью API Mininet воспроизвела эксперимент по добавлению задержки для интерфейса хоста, подключающегося к эмулируемой глобальной сети. В виртуальной среде mininet в своём рабочем каталоге с проектами создала каталог simple-delay и перешла в него. (рис. 4.13).

```

mininet@mininet-vm:~/work$ mkdir -p ~/work/lab_neterm_i/simple-delay
mininet@mininet-vm:~/work$ cd ~/work/lab_neterm_i/simple-delay
mininet@mininet-vm:~/work/lab_neterm_i/simple-delay$ touch lab_neterm_i.py
mininet@mininet-vm:~/work/lab_neterm_i/simple-delay$ ls
lab_neterm_i.py
mininet@mininet-vm:~/work/lab_neterm_i/simple-delay$ mcedit lab_neterm_i.py

```

Рис. 4.13: Создание каталога для эксперимента

Создала скрипт для эксперимента (рис. 4.14).

```

/home/mininet/work/lab_neterm_i/simple-delay/lab_neterm_i.py [----] 14 L: 1+ 51 / 51 + (1268/12800) <EOF>
#!/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 switches')
    s1 = net.addSwitch( 's1' )

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

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

    info('*** Set delay')
    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)

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

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

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

```

Рис. 4.14: Скрипт для эксперимента

Создала скрипт для визуализации результатов эксперимента (рис. 4.15).

```

/home/mininet/work/lab_neterm_i/simple-delay/ping_plot [----] 26 L: 1+ 7
#!/usr/bin/gnuplot --persist

set terminal png crop
set output 'ping.png'
set xlabel "Sequence number"
set ylabel "Delay (ms)"
set grid
plot "ping.dat" with lines

```

Рис. 4.15: Скрипт для визуализации эксперимента

Задала права доступа к файлу скрипта (рис. 4.16).

```

mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ chmod +x ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ touch Makefile
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ mcedit Makefile
^[[<0;160;1m
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ mcedit Makefile
^[[<0;160;1m

```

Рис. 4.16: Права доступа

Создала Makefile для управления процессом проведения эксперимента (рис. 4.17).

```

/home/mininet/work/lab_netem_i/simple-delay/Makefile [-H--] 26 L:[

all: ping.dat ping.png

ping.dat:
<----->sudo python lab_netem_i.py
<----->sudo chown mininet:mininet ping.dat

ping.png: ping.dat
<----->./ping_plot

clean:
<----->-rm -f *.dat *.png

```

Рис. 4.17: Makefile

Выполнила эксперимент (рис. 4.18).

```

mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make
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, $7)\' | sed -e \'s/time=/g\' -e \'s/icmp_seq=/g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
...
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ ls
lab_netem_i.py Makefile ping.dat ping_plot ping.png ping_pot

```

Рис. 4.18: Выполнение эксперимента

Продемонстрировала график (рис. 4.19).

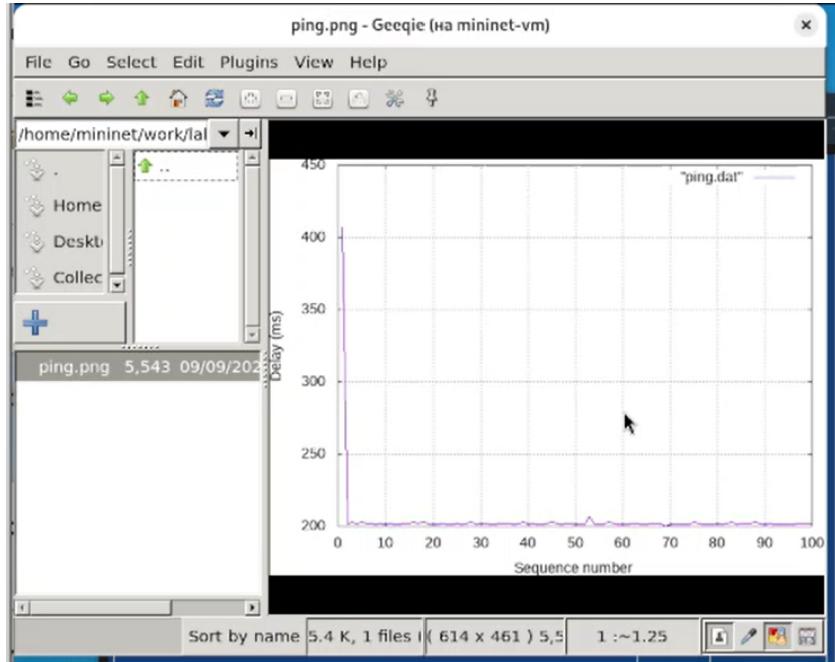


Рис. 4.19: График

Разработала скрипт для вычисления минимального, среднего, максимального и стандартного rtt на основе данных файла ping.dat, добавила правило запуска в Makefile (рис. 4.20).

```
/home/mininet/work/lab_netem_i/simple-delay/rtt.py [-M--] 78 L:[ 1+ 8 9/ 9] *(345 /
with open('ping.dat', 'r') as file:
    s = []
    for line in file.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)-1))**0.5
print(f"min: {min(s)} \max: {max(s)} \n avg: {sum(s)/len(s)} \n std: {std}")
```

Рис. 4.20: Скрипт для вычисления rtt

Продемонстрировала работу скрипта (рис. 4.21).

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ sudo python rtt.py
min: 200 \max: 407
avg: 203.72
std: 20.552018259661388
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ mcedit ping.dat

mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make clean
rm -f *.dat *.png
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ cd
```

Рис. 4.21: Вычисленное rtt

4.2.3 Воспроизводимый эксперимент по изменению задержки

В виртуальной среде mininet в своём рабочем каталоге с проектами создала каталог change-delay и перешла в него. (рис. 4.22).

```
mininet@mininet-vm:~/work/lab_neterm_i$ mkdir -p ~/work/lab_neterm_i/change-delay
mininet@mininet-vm:~/work/lab_neterm_i$ cd change-delay
-bash: cd: change_delay: No such file or directory
mininet@mininet-vm:~/work/lab_neterm_i$ cd change-delay
```

Рис. 4.22: Создание каталога для эксперимента

Скопировала необходимые файлы из предыдущего эксперимента (рис. 4.23).

```
mininet@mininet-vm:~/work/lab_neterm_i/change-delay$ ls
lab_neterm_i.py  Makefile  ping_plot  rtt.py
mininet@mininet-vm:~/work/lab_neterm_i/change-delay$ mcedit lab_neterm_i.py
```

Рис. 4.23: Файлы для эксперимента

Создала скрипт для эксперимента (рис. 4.24).

```
/home/mininet/work/lab_neterm_i/change-delay/lab_neterm_i.py  [-M--] 14 L:[ 1+57 58/ 58] *(1546/1546b) <EOF>
#!/usr/bin/env python
...
simple experiment.
Output to 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)

    info('*** Ping\n')
    h1.cmdPrint('ping -c 50', h2.IP(), '| grep "time=" | awk \'{print $5, $7}\' | sed -e \'$s/time//g\' -e \'$s/icmp_seq//g\' > First_ping.dat')

    info('*** Change delay\n')
    h1.cmdPrint('tc qdisc change dev h1-eth0 root netem delay 50ms')
    h2.cmdPrint('tc qdisc change dev h2-eth0 root netem delay 50ms')

    info('*** Ping\n')
    h1.cmdPrint('ping -c 50', h2.IP(), '| grep "time=" | awk \'{print $5, $7}\' | sed -e \'$s/time//g\' -e \'$s/icmp_seq//g\' > ping.dat')
    ...
    info('*** Stopping network')
    net.stop()

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

Рис. 4.24: Скрипт для эксперимента

Выполнила эксперимент (рис. 4.25).

```

mininet@mininet-vm:~/work/lab_neterm_i/change-delay$ make
sudo python lab_neterm_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 50', '| grep "time=" | awk \'(print $5, $7)\' | sed -e \'s/time//g\' -e \'s/icmp_seq//g\' > first_ping.dat')
*** Change delay
*** h1 : ('tc qdisc change dev h1-eth0 root netem delay 50ms')
*** h2 : ('tc qdisc change dev h2-eth0 root netem delay 50ms')
*** Ping
*** h1 : ('ping -c 50', '| grep "time=" | awk \'(print $5, $7)\' | sed -e \'s/time//g\' -e \'s/icmp_seq//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_neterm_i/change-delay$ mc
Select an editor. To change later, run 'select-editor'.
1. /bin/nano      <---- easiest
2. /usr/bin/vim.basic
3. /usr/bin/mcedit
4. /usr/bin/vim.tiny
5. /usr/bin/emacs
6. /bin/ed

Choose 1-6 [1]: exit
Choose 1-6 [1]:
Choose 1-6 [1]:
mininet@mininet-vm:~/work/lab_neterm_i/change-delay$ sudo python rtt.py
min: 100 |max: 103
avg: 101.36
std: 0.6627093431281831

```

Рис. 4.25: Выполнение эксперимента

Продемонстрировала график (рис. 4.26).

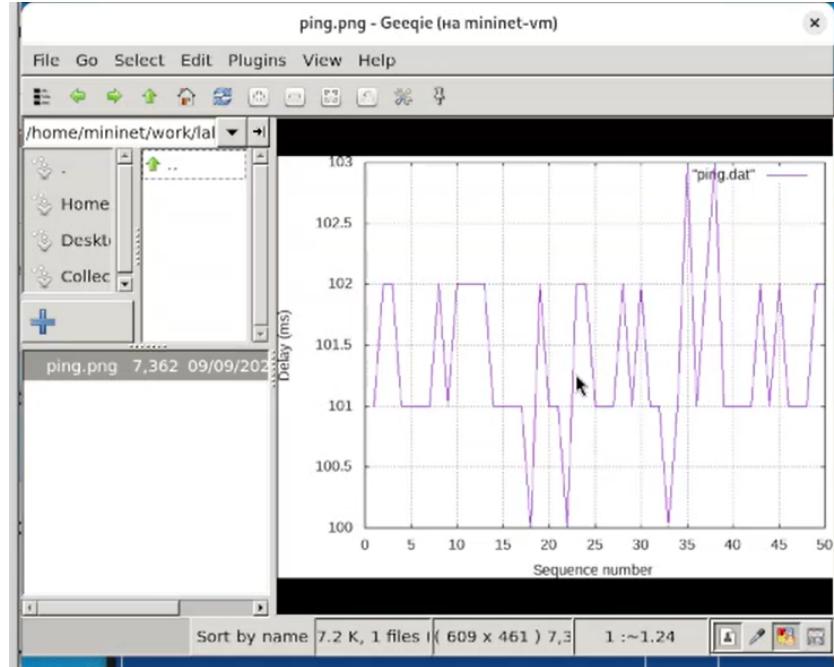


Рис. 4.26: График

4.2.4 Воспроизводимый эксперимент по изменению джиттера

В виртуальной среде mininet в своём рабочем каталоге с проектами создала каталог jitter-delay и перешла в него. Скопировала необходимые файлы из предыдущего эксперимента. (рис. 4.27).

```
mininet@mininet-vm: /work/lab_netem_1/change-delay$ cd ..
mininet@mininet-vm: /work/lab_netem_1$ mkdir -p ~/work/lab_netem_1/jitter-delay
mininet@mininet-vm: /work/lab_netem_1$ cp -r ~/work/lab_netem_1/change-delay/lab_netem_1.py ~/work/lab_netem_1/jitter-delay/lab_netem_1.py
mininet@mininet-vm: /work/lab_netem_1$ cp -r ~/work/lab_netem_1/change-delay/rtt.py ~/work/lab_netem_1/jitter-delay/rtt.py
mininet@mininet-vm: /work/lab_netem_1$ cp -r ~/work/lab_netem_1/change-delay/Makefile ~/work/lab_netem_1/jitter-delay/Makefile
mininet@mininet-vm: /work/lab_netem_1$ cp -r ~/work/lab_netem_1/change-delay/ping_plot ~/work/lab_netem_1/jitter-delay/ping_plot
mininet@mininet-vm: /work/lab_netem_1$ cd jitter-delay
-bash: cd: /jitter-delay: No such file or directory
mininet@mininet-vm: /work/lab_netem_1$ cd jitter-delay
mininet@mininet-vm: /work/lab_netem_1/jitter-delay$ ls
lab_netem_1.py  Makefile  ping_plot  rtt.py
mininet@mininet-vm: /work/lab_netem_1/jitter-delay$ mcedit lab_netem_1.py
```

Рис. 4.27: Создание каталога для эксперимента

Создала скрипт для эксперимента (рис. 4.28).

```
/home/mininet/work/lab_netem_1/jitter-delay/lab_netem_1.py [-H--] 70 L: 1+47 48/ 58] +(1281/1564b) 39 0x827
#!/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 10ms')
    h2.cmdPrint('tc qdisc add dev h2-eth0 root netem delay 100ms 10ms')

    time.sleep(10)

    info('*** Ping\n')
    h1.cmdPrint('ping -c 50', h2.IP(), '| grep "time:" | awk \'{print $5, $7}\' | sed -e \'/s/time//g\' -e \'s/icmp_seq//g\' > first_ping.dat')

    info('*** Change delay\n')
    h1.cmdPrint('tc qdisc change dev h1-eth0 root netem delay 50ms 5ms')
    h2.cmdPrint('tc qdisc change dev h2-eth0 root netem delay 50ms 5ms')

    info('*** Ping\n')
    h1.cmdPrint('ping -c 50', h2.IP(), '| grep "time:" | awk \'{print $5, $7}\' | sed -e \'/s/time//g\' -e \'s/icmp_seq//g\' > ping.dat')
    ...
    info('*** Stopping network')
    net.stop()

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

Рис. 4.28: Скрипт для эксперимента

Выполнила эксперимент (рис. 4.29).

```

root@mininet-vm:~/work/lab_netem_1/jitter-delay$ ping.py
mininet@mininet-vm:~/work/lab_netem_1/jitter-delay$ make
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 10ms')
*** h2 : ('tc qdisc add dev h2-eth0 root netem delay 100ms 10ms')
*** Ping
*** h1 : ('ping -c 50', '10.0.0.2', '| grep "time" | awk \'(print $5, $7)\' | sed -e \'$s/time=/g\' -e \'$s/icmp_seq=/g\' > first_ping.dat')
*** Change delay
*** h1 : ('tc qdisc change dev h1-eth0 root netem delay 50ms 5ms')
*** h2 : ('tc qdisc change dev h2-eth0 root netem delay 50ms 5ms')
*** Ping
*** h1 : ('ping -c 50', '10.0.0.2', '| grep "time" | awk \'(print $5, $7)\' | sed -e \'$s/time=/g\' -e \'$s/icmp_seq=/g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
...
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chmod mininet:mininet ping.dat
./ping-jitter
mininet@mininet-vm:~/work/lab_netem_1/jitter-delay$ ls
first_ping.dat lab.netem.i.py Makefile ping.dat ping_plot ping.png rtt.py
mininet@mininet-vm:~/work/lab_netem_1/jitter-delay$ geeqie ping.png

```

Рис. 4.29: Выполнение эксперимента

Продемонстрировала график (рис. 4.30).

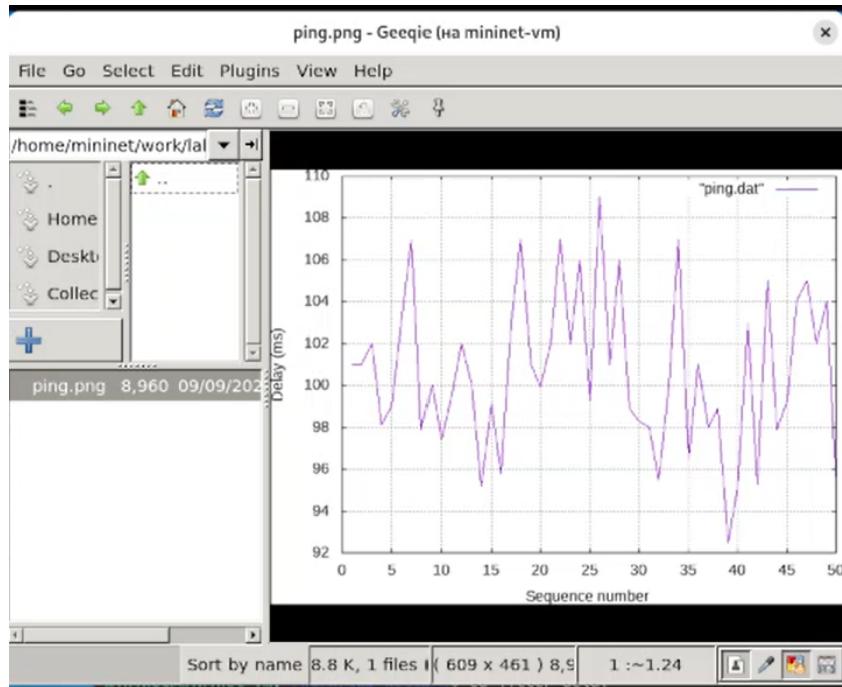


Рис. 4.30: График

Продемонстрировала работу скрипта для вычисления rtt (рис. 4.31).

```

mininet@mininet-vm:~/work/lab_netem_i/jitter-delay$ mcedit rtt.py

mininet@mininet-vm:~/work/lab_netem_i/jitter-delay$ sudo python rtt.py
min: 92.5 \max: 109.0
avg: 100.634
std: 3.7713533542282907

```

Рис. 4.31: Вычисленное rtt

4.2.5 Воспроизводимый эксперимент по изменению значения корреляции джиттера и задержки

В виртуальной среде mininet в своём рабочем каталоге с проектами создала каталог correlation-delay и перешла в него. Скопировала необходимые файлы из предыдущего эксперимента. (рис. 4.32).

```

mininet@mininet-vm:~/work/lab_netem_i/jitter-delay$ cd ..
mininet@mininet-vm:~/work/lab_netem_i$ mkdir -p /correlation-delay
mkdir: cannot create directory '/correlation-delay': Permission denied
mininet@mininet-vm:~/work/lab_netem_i$ mkdir -p correlation-delay
mininet@mininet-vm:~/work/lab_netem_i$ cd correlation-delay
mininet@mininet-vm:~/work/lab_netem_i$ ls
mininet@mininet-vm:~/work/lab_netem_i$ correlation-delay$ cp -v /work/lab_netem_i/change-delay/lab_netem_i.py ~/work/lab_netem_i/correlation-delay/lab_netem_i.py
mininet@mininet-vm:~/work/lab_netem_i$ correlation-delay$ cp -v /work/lab_netem_i/change-delay/rtt.py ~/work/lab_netem_i/correlation-delay/rtt.py
mininet@mininet-vm:~/work/lab_netem_i$ correlation-delay$ cp -v /work/lab_netem_i/change-delay/Makefile ~/work/lab_netem_i/correlation-delay/Makefile
mininet@mininet-vm:~/work/lab_netem_i$ correlation-delay$ cp -v /work/lab_netem_i/jitter-delay/rtt.py ~/work/lab_netem_i/correlation-delay/rtt.py
lab_netem_i.py Makefile rtt.py
mininet@mininet-vm:~/work/lab_netem_i/correlation-delay$ ls
mininet@mininet-vm:~/work/lab_netem_i/correlation-delay$ cp -v /work/lab_netem_i/change-delay/ping_plot ~/work/lab_netem_i/correlation-delay/ping_plot
cp: cannot create regular file '/home/mininet/work/lab_netem_i/correlation-delay/ping_plot': No such file or directory
mininet@mininet-vm:~/work/lab_netem_i/correlation-delay$ cp -v /work/lab_netem_i/change-delay/ping_plot ~/work/lab_netem_i/correlation-delay/ping_plot
mininet@mininet-vm:~/work/lab_netem_i/correlation-delay$ mcedit lab_netem_i.py
mininet@mininet-vm:~/work/lab_netem_i/correlation-delay$ mcedit rtt.py

```

Рис. 4.32: Создание каталога для эксперимента

Создала скрипт для эксперимента (рис. 4.33).

```

/home/mininet/work/lab_netem_i/correlation-delay/lab_netem_i.py  [----] 73 L:[ 1+38 39/ 58] *(922 /1580b)  39 0x027
#!/usr/bin/env python

...
Simple experiment.
Outputs 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 10ms 25%')
    h2.cmdPrint('tc qdisc add dev h2-eth0 root netem delay 100ms 10ms 25%')

    time.sleep(10)

    info('*** Ping\n')
    h1.cmdPrint('ping -c 50', h2.IP(), '| grep "time" | awk \'{print $5, $7}\' | sed -e \'$s/time//g\' -e \'$s/icmp_seq//g\' > first_ping.dat')

    info('*** Change delay\n')
    h1.cmdPrint('tc qdisc change dev h1-eth0 root netem delay 50ms 5ms 20%')
    h2.cmdPrint('tc qdisc change dev h2-eth0 root netem delay 50ms 5ms 20%')

    info('*** Ping\n')
    h1.cmdPrint('ping -c 50', h2.IP(), '| grep "time" | awk \'{print $5, $7}\' | sed -e \'$s/time//g\' -e \'$s/icmp_seq//g\' > ping.dat')

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

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

```

Рис. 4.33: Скрипт для эксперимента

Выполнила эксперимент (рис. 4.34).

```

mininet@mininet-vm:~/work/lab_netem_i/correlation-delay$ make
sudo python lab_netem_i.py
*** Adding controller
*** Adding host h1
*** Adding host h2
*** Adding switch s1
*** 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 10ms 25%',)
*** h2 : ('tc qdisc add dev h2-eth0 root netem delay 100ms 10ms 25%',)
*** Ping
*** h1 : ('ping -c 50', '10.0.0.2', '| grep "time" | awk \'{print $5, $7}\' | sed -e \'$s/time//g\' -e \'$s/icmp_seq//g\' > first_ping.dat')
*** Change delay
*** h1 : ('tc qdisc change dev h1-eth0 root netem delay 50ms 5ms 20%',)
*** h2 : ('tc qdisc change dev h2-eth0 root netem delay 50ms 5ms 20%',)
*** Ping
*** h1 : ('ping -c 50', '10.0.0.2', '| grep "time" | awk \'{print $5, $7}\' | sed -e \'$s/time//g\' -e \'$s/icmp_seq//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 2 links
...
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/correlation-delay$ geegoo ping.png
Gtk-Message: 13:59:33.405: Failed to load module "canberra-gtk-module"
Gtk-Message: 13:59:33.405: Failed to load module "pk-gtk-module"
mininet@mininet-vm:~/work/lab_netem_i/correlation-delay$ sudo python rtt.py
min: 93.6 \max: 111.0
avg: 106.89460000000002
std: 4.369953532653525

```

Рис. 4.34: Выполнение эксперимента

Продемонстрировала график (рис. 4.35).

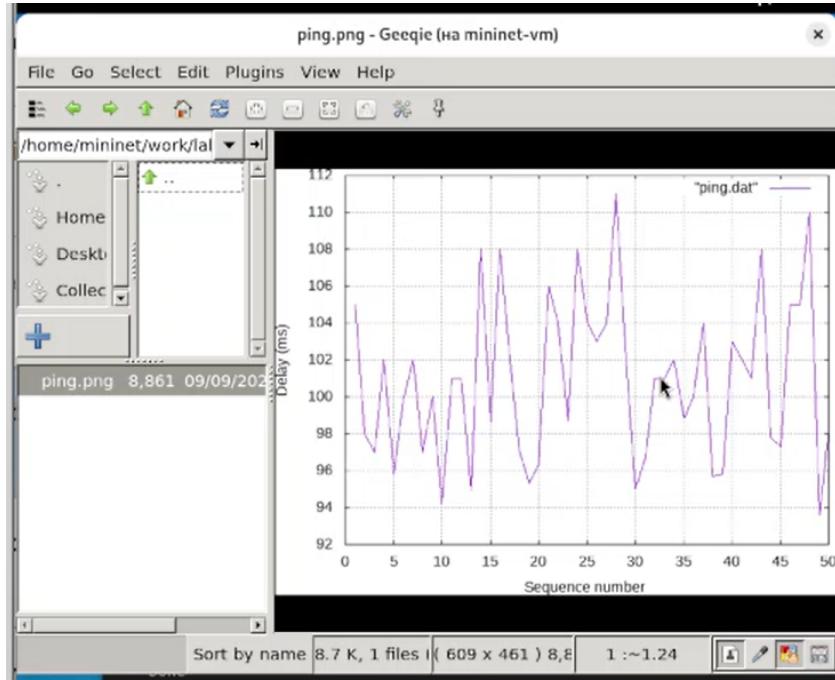


Рис. 4.35: График

4.2.6 Воспроизводимый эксперимент по изменению распределения времени задержки в эмулируемой глобальной сети

В виртуальной среде mininet в своём рабочем каталоге с проектами создала каталог distribution-delay и перешла в него. Скопировала необходимые файлы из предыдущего эксперимента. (рис. 4.36).

```
mininet@mininet-vm:~/work/lab_netem_1/correlation-delay$ cd ..
mininet@mininet-vm:~/work/lab_netem_1$ mkdir -p distribution-delay
mininet@mininet-vm:~/work/lab_netem_1$ cd distribution-delay
mininet@mininet-vm:~/work/lab_netem_1/distribution-delay$ cp ~/work/lab_netem_1/correlation-delay/lab_netem_i.py ~/work/lab_netem_1/distribution-delay/lab_netem_i.py
mininet@mininet-vm:~/work/lab_netem_1/distribution-delay$ cp ~/work/lab_netem_1/correlation-delay/Makefile ~/work/lab_netem_1/distribution-delay/Makefile
mininet@mininet-vm:~/work/lab_netem_1/distribution-delay$ cp ~/work/lab_netem_1/correlation-delay/ping_plot ~/work/lab_netem_1/distribution-delay/ping_plot
mininet@mininet-vm:~/work/lab_netem_1/distribution-delay$ cp ~/work/lab_netem_1/correlation-delay/rtt.py ~/work/lab_netem_1/distribution-delay/rtt.py
mininet@mininet-vm:~/work/lab_netem_1/distribution-delay$ ls
lab_netem_i.py Makefile ping_plot rtt.py
mininet@mininet-vm:~/work/lab_netem_1/distribution-delay$ mcedit lab_netem_i.py
```

Рис. 4.36: Создание каталога для эксперимента

Создала скрипт для эксперимента (рис. 4.37).

```

/home/mininet/work/lab_netem_1/distribution-delay/lab_netem_1.py [----] 90 L:[ 1+47 48/ 58] *(1329/18126) 39 0x027
$ ./user/mininet 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 10ms 25%')
    h2.cmdPrint('tc qdisc add dev h2-eth0 root netem delay 100ms 10ms 25%')

    time.sleep(10)

    info('*** Ping\n')
    h1.cmdPrint('ping -c 50', h2.IP(), '| grep "time=" | awk \'{print $5, $7}\' | sed -e \'$s/time//g\' -e \'$s/icmp_seq//g\' > first_ping.dat')

    info('*** Change delay\n')
    h1.cmdPrint('tc qdisc change dev h1-eth0 root netem delay 50ms 5ms distribution normal')
    h2.cmdPrint('tc qdisc change dev h2-eth0 root netem delay 50ms 5ms distribution normal')

    info('*** Ping\n')
    h1.cmdPrint('ping -c 50', h2.IP(), '| grep "time=" | awk \'{print $5, $7}\' | sed -e \'$s/time//g\' -e \'$s/icmp_seq//g\' > ping.dat')

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

if __name__ == '__main__':
    setLogLevel('info')
    ...

```

Рис. 4.37: Скрипт для эксперимента

Выполнила эксперимент (рис. 4.38).

```

mininet@mininet-vm:/work/lab_netem_1/distribution-delay$ make
sudo python lab_netem_1.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 10ms 25%')
*** h2 : ('tc qdisc add dev h2-eth0 root netem delay 100ms 10ms 25%')
...
*** Ping
...
*** h1 : ('ping -c 50', '10.0.0.2', '| grep "time=" | awk \'{print $5, $7}\' | sed -e \'$s/time//g\' -e \'$s/icmp_seq//g\' > first_ping.dat')
...
*** Change delay
...
*** h1 : ('tc qdisc change dev h1-eth0 root netem delay 50ms 5ms distribution normal')
*** h2 : ('tc qdisc change dev h2-eth0 root netem delay 50ms 5ms distribution normal')
...
*** Ping
...
*** h1 : ('ping -c 50', '10.0.0.2', '| grep "time=" | awk \'{print $5, $7}\' | sed -e \'$s/time//g\' -e \'$s/icmp_seq//g\' > ping.dat')
...
*** Stopping network
...
*** Stopping 1 controllers
c0
...
*** Stopping 2 links
...
*** Stopping 1 switches
s1
...
*** Stopping 2 hosts
h1 h2
...
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:/work/lab_netem_1/distribution-delay$ geogebra ping.png
Gtk-WARNING: 14:09:23.089: Failed to load module "canberra-gtk-module"
Gtk-MESSAGE: 14:09:23.089: Failed to load module "pango-gtk-module"
mininet@mininet-vm:/work/lab_netem_1/distribution-delay$ sudo python rtt.py
min: 81.3 max: 123.0
avg: 101.29599999999999
std: 8.007622898825588

```

Рис. 4.38: Выполнение эксперимента

Продемонстрировала график (рис. 4.39).

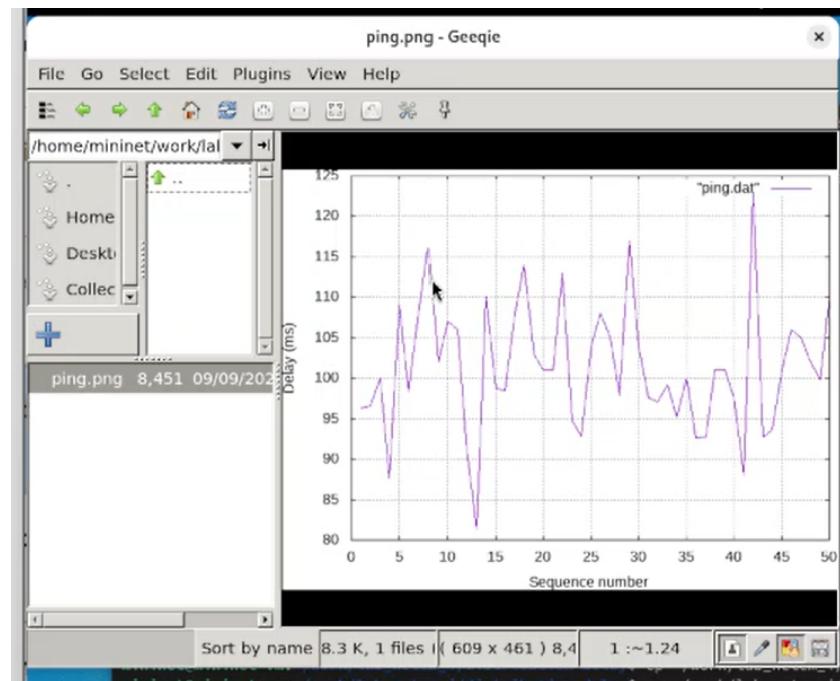


Рис. 4.39: График

5 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.

Список литературы