

Лабораторная работа №4

Эмуляция и измерение задержек в глобальных сетях

Барабанова Кристина

Содержание

Цель работы	4
Задание	5
Выполнение лабораторной работы	6
1. Запуск лабораторной топологии	7
2. Интерактивные эксперименты	10
Добавление/изменение задержки в эмулируемой глобальной сети	10
Изменение задержки в эмулируемой глобальной сети	12
Восстановление исходных значений (удаление правил) задержки в эмулируемой глобальной сети	13
Добавление значения дрожания задержки в интерфейс подключения к эмулируемой глобальной сети	14
Добавление значения корреляции для джиттера и задержки в интерфейс подключения к эмулируемой глобальной сети	15
Распределение задержки в интерфейсе подключения к эмулируемой глобальной сети	16
3. Воспроизведение экспериментов	17
Предварительная подготовка	17
Добавление задержки для интерфейса, подключающегося к эмулируемой глобальной сети	18
<pre>mininet@mininet-vm:~/work/lab_netem_i/simple-delay\$ make clean</pre>	
<pre>rm -f *.dat *.png</pre>	22
5. Выводы	22

Список иллюстраций

1	рис. 1	7
2	рис. 2	7
3	рис. 3	8
4	рис. 4	8
5	рис. 5	8
1	рис. 6	10
2	рис. 7	11
3	рис. 8	11
4	рис. 9	11
5	рис. 10	12
6	рис. 11	12
7	рис. 12	12
8	рис. 13	13
9	рис. 14	13
10	рис. 15	13
11	рис. 16	14
12	рис. 17	15
13	рис. 18	16
1	рис. 19	17
2	рис. 20	18
3	рис. 21	18
4	рис. 22	19
5	рис. 23	19
6	рис. 24	19
7	рис. 25	20
8	рис. 26	20
9	рис. 27	21
10	рис. 28	21
11	рис. 29	22

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.

Выполнение лабораторной работы

1. Запуск лабораторной топологии

Запустите виртуальную среду с mininet. Из основной ОС подключитесь к виртуальной машине:

```
mininet@mininet-vm:~$ ssh -Y mininet@192.168.56.104
mininet@192.168.56.104'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: Fri Jan 30 03:20:11 2026 from 192.168.56.1
```

Рис. 1: рис. 1

В виртуальной машине mininet при необходимости исправьте права запуска X-соединения. Скопируйте значение куки (MIT magic cookie)1 своего пользователя mininet в файл для пользователя root:

```
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:11 MIT-MAGIC-COOKIE-1 afeacff697c10b7cea700f81669d16aa
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth add ^C
root@mininet-vm:~# xauth add mininet-vm/unix:11 MIT-MAGIC-COOKIE-1 afeacff697c
10b7cea700f81669d16aa
root@mininet-vm:~# logout
```

Рис. 2: рис. 2

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

После введения этой команды запустятся терминалы двух хостов, коммутатора и контроллера.

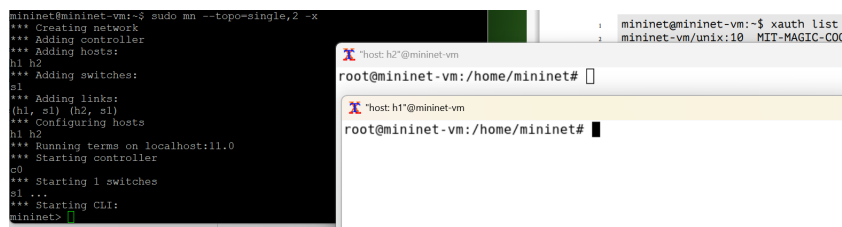


Рис. 3: рис. 3

На хостах h1 и h2 введите команду `ifconfig`, чтобы отобразить информацию, относящуюся к их сетевым интерфейсам и назначенным им IP-адресам

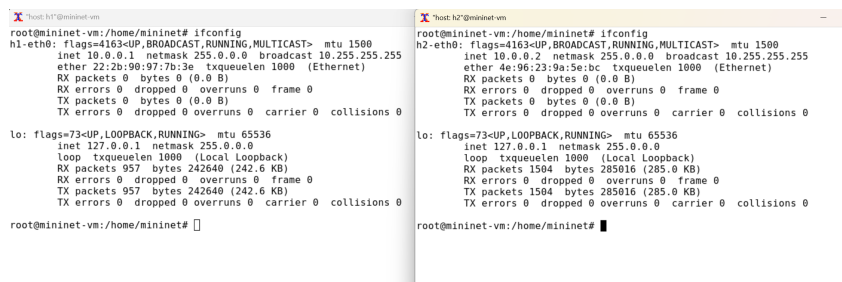


Рис. 4: рис. 4

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

```
root@mininet-virtual-machine:~# ping 10.0.0.2 -c 6
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=1.82 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.317 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.060 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.063 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.061 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 5097ms
rtt min/avg/max/mdev = 0.060/0.398/1.821/0.642 ms
```

Рис. 5: рис. 5

RTT: 0.060/0.398/1.821/0.642 ms

2. 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 пространства WAN, соединяющей оба коммутатора.

1. На хосте h1 добавьте задержку в 100 мс к выходному интерфейсу:

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

Рис. 1: рис. 6

2. Проверьте, что соединение от хоста h1 к хосту h2 имеет задержку 100 мс, используя команду `ping` с параметром `-c 6` с хоста h1. Укажите в отчёте минимальное, среднее, максимальное и стандартное отклонение времени приёма-передачи (RTT).

```

root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 6
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=101 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=100 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 5008ms
rtt min/avg/max/mdev = 100.209/100.596/101.399/0.448 ms

```

Рис. 2: рис. 7

RTT: 100.209/100.596/101.399/0.448

- Для эмуляции глобальной сети с двунаправленной задержкой необходимо к соответствующему интерфейсу на хосте h2 также добавить задержку в 100 миллисекунд:

```

root@mininet-vm:/home/mininet# sudo tc qdisc add dev h2-eth0 root netem delay 100ms

```

Рис. 3: рис. 8

- Проверьте, 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. Укажите в отчёте минимальное, среднее, максимальное и стандартное отклонение времени приёма-передачи (RTT).

```

root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 6
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=201 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=200 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=201 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 5003ms
rtt min/avg/max/mdev = 200.362/200.962/201.873/0.481 ms

```

Рис. 4: рис. 9

RTT: 200.362/200.962/201.873/0.481

Изменение задержки в эмулируемой глобальной сети

1. Измените задержку со 100 мс до 50 мс для отправителя h1:

```
root@mininet-vm:/home/mininet# sudo tc qdisc change dev h1-eth0 root netem delay 50ms
```

Рис. 5: рис. 10

и для получателя h2:

```
root@mininet-vm:/home/mininet# sudo tc qdisc change dev h2-eth0 root netem delay 50ms
```

Рис. 6: рис. 11

2. Проверьте, что соединение от хоста h1 к хосту h2 имеет задержку 100 мс, используя команду ping с параметром -c 6 с терминала хоста h1. Укажите в отчёте минимальное, среднее, максимальное и стандартное отклонение времени приёма-передачи (RTT).

```
root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 6
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=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=101 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 5005ms
rtt min/avg/max/mdev = 100.267/100.968/101.548/0.436 ms
```

Рис. 7: рис. 12

RTT: 100.267/100.968/101.548/0.436

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

1. Восстановите конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса. Для отправителя h1:

```
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
```

Рис. 8: рис. 13

Для получателя h2:

```
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h2-eth0 root netem
```

Рис. 9: рис. 14

2. Проверьте, что соединение между хостом h1 и хостом h2 не имеет явно утановленной задержки, используя команду ping с параметром -c 6 с терминала хоста h1. Укажите в отчёте минимальное, среднее, максимальное и стандартное отклонение времени приёма-передачи (RTT).

```
root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 6
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=1.17 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.480 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.186 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=0.123 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.059 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5089ms
rtt min/avg/max/mdev = 0.059/0.349/1.172/0.394 ms
```

Рис. 10: рис. 15

RTT: 0.059/0.349/1.172/0.394 ms

Добавление значения дрожания задержки в интерфейс подключения к эмулируемой глобальной сети

2. Добавьте на узле h1 задержку в 100 мс со случайным отклонением 10 мс:
3. Проверьте, 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`. Укажите в отчёте минимальное, среднее, максимальное и стандартное отклонение времени приёма-передачи (RTT).
4. Восстановите конфигурацию интерфейса по умолчанию на узле h1.

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms 10ms
root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 6
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=109 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=93.0 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=110 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=109 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=107 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=104 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 93.020/105.409/109.846/5.818 ms
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
```

Рис. 11: рис. 16

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

1. При необходимости восстановите конфигурацию интерфейсов по умолчанию на узлах h1 и h2.
2. Добавьте на интерфейсе хоста h1 задержку в 100 мс с вариацией ± 10 мс и значением корреляции в 25%:
3. Убедитесь, 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%. Используйте для этого в терминале хоста h1 команду ping с параметром -c 20. Укажите в отчёте минимальное, среднее, максимальное и стандартное отклонение времени приёма-передачи (RTT).

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 1
00ms 10ms 25%
root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 20
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=92.2 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=91.1 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=106 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=99.3 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=94.1 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=105 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=96.3 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=92.9 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=95.9 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=109 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=95.3 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=92.7 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=93.5 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=107 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=109 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=93.0 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=93.6 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=94.2 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=104 ms

--- 10.0.0.2 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19033ms
rtt min/avg/max/mdev = 91.080/98.370/108.719/6.057 ms
```

Рис. 12: рис. 17

4. Восстановите конфигурацию интерфейса по умолчанию на узле h1.

Распределение задержки в интерфейсе подключения к эмулируемой глобальной сети

1. При необходимости восстановите конфигурацию интерфейсов по умолчанию на узлах h1 и h2.
2. Задайте нормальное распределение задержки на узле h1 в эмулируемой сети:
3. Убедитесь, что все пакеты, покидающие хост h1 на интерфейсе h1-eth0, будут иметь время задержки, которое распределено в диапазоне 100 мс \pm 20 мс.

```
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 10.0.0.2 -c 10
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=93.4 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=99.5 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=126 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=81.3 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=97.8 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=91.3 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=84.2 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=83.1 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=114 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=149 ms

--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9016ms
rtt min/avg/max/mdev = 81.331/101.977/149.122/20.629 ms
root@mininet-vm:/home/mininet# █
```

Рис. 13: рис. 18

4. Восстановите конфигурацию интерфейса по умолчанию на узле h1.
5. Завершите работу mininet в интерактивном режиме.

3. Воспроизведение экспериментов

Предварительная подготовка

1. Обновите репозитории программного обеспечения на виртуальной машине:

```
mininet@mininet-vm:~$ sudo apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [128 kB]
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]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [1,114 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [3,957 kB]
Fetched 5,455 kB in 9s (636 kB/s)
Reading package lists... Done
```

Рис. 1: рис. 19

2. Установите пакет `gheeie` — понадобится для просмотра файлов `png`:

```
mininet@mininet-vm:~$ sudo apt install geeqie
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
acl apg apport apport-symptoms aptdaemon aptdaemon-data avahi-daemon
avahi-utils bluez bolt cheese-common colord colord-data
cracklib-runtime cups-bsd cups-client cups-common cups-pk-helper
dbus dbus-x11 dconf-cli desktop-file-utils dns-root-data
dnsmasq-base docbook-xml evolution-data-server
evolution-data-server-common exiftran exiv2 fprintd gcr gdm3
geeqie-common geoclue-2.0 gir1.2-accountsservice-1.0 gir1.2-atk-1.0
gir1.2-atspi-2.0 gir1.2-freedesktop gir1.2-gck-1 gir1.2-gcr-3
gir1.2-gdesktopenums-3.0 gir1.2-gdkpixbuf-2.0 gir1.2-gdm-1.0
gir1.2-geoclue-2.0 gir1.2-gnomebluetooth-1.0 gir1.2-gnomedesktop-3.0
gir1.2-graphene-1.0 gir1.2-gtk-3.0 gir1.2-gweather-3.0
gir1.2-ibus-1.0 gir1.2-json-1.0 gir1.2-mutter-6 gir1.2-nm-1.0
gir1.2-nma-1.0 gir1.2-notify-0.7 gir1.2-packagekitglib-1.0
gir1.2-pango-1.0 gir1.2-polkit-1.0 gir1.2-rsvg-2.0 gir1.2-secret-1
gir1.2-soup-2.4 gir1.2-upowerglib-1.0 gir1.2-vte-2.91 gjs
gkbd-capplet gnome-control-center gnome-control-center-data
gnome-control-center-faces gnome-keyring gnome-keyring-pkcs11
gnome-menus gnome-online-accounts gnome-session-bin
gnome-session-common gnome-settings-daemon
gnome-settings-daemon-common gnome-shell gnome-shell-common
gnome-startup-applications gnome-user-docs gstreamer1.0-clutter-3.0
gstreamer1.0-gl gstreamer1.0-plugins-good gstreamer1.0-pulseaudio
gstreamer1.0-x 1965-va-driver ibus ibus-data ibus-gtk ibus-gtk3
iio-sensor-proxy im-config intel-media-va-driver ippusbxd
language-selector-common language-selector-gnome libaa1 libaacs0
libaom0 libappindicator3-1 libappstream4 libasound2-plugins libass9
libavahi-core7 libavahi-glib1 libavc1394-0 libavcodec58 libavfilter7
libavformat58 libavutil56 libbdlplus0 libbluetooth3 libbluray2
libboost-thread1.71.0 libbs2b0 libcac0 libcamel-1.2-62
libcanberra-gtk3-0 libcanberra-gtk3-module libcanberra-pulse
libcheese-gtk25 libcheese8 libchromaprint1 libclutter-1.0-0
libclutter-1.0-common libclutter-gst-3.0-0 libclutter-gtk-1.0-0
libcodecs-2.0.9 libcogl-common libcogl-pango28 libcogl-path20
libcogl20 libcolord-gtk1 libcolorhug2 libcrack2 libcups2 libdaemon0
libdbus-1-3 libdbusmenu-glib4 libdbusmenu-gtk3-4 libdrm-amdgpu1
libdv4 libebackend-1.2-10 libebook-1.2-20 libebook-contacts-1.2-3
...
```

Рис. 2: рис. 20

3. Для каждого воспроизводимого эксперимента ехрname создайте свой каталог, в котором будут размещаться файлы эксперимента:

```
mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_i/expname
```

Рис. 3: рис. 21

Добавление задержки для интерфейса, подключающегося к эмулируемой глобальной сети

1. В виртуальной среде mininet в своём рабочем каталоге с проектами создайте каталог simple-delay и перейдите в него.
2. Создаёте скрипт для эксперимента lab_netem_i.py:

```

GNU nano 4.8                                lab_netem_i.py
#!/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()

  Get Help      Write Out    Where Is      Cut Text      Justify       Cur Pos      Undo
  Exit          Read File    Replace       Paste Text    To Spell     Go To Line   Redo

```

Рис. 4: рис. 22

4. Создаёте скрипт для визуализации ping_plot результатов эксперимента.

```

GNU nano 4.8                                ping_plot
#!/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. 5: рис. 23

5. Задайте права доступа к файлу скрипта:

```

mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ chmod +x ping_plot

```

Рис. 6: рис. 24

6. Создайте Makefile для управления процессом проведения эксперимента:

```
GNU nano 4.8                               Makefile
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. 7: рис. 25

7. Выполните эксперимент:

```
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
```

Рис. 8: рис. 26

9. Из файла ping.dat удалите первую строку и заново постройте график:

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make ping.png  
./ping_plot
```

Рис. 9: рис. 27

10. Продемонстрируйте построенный в результате график

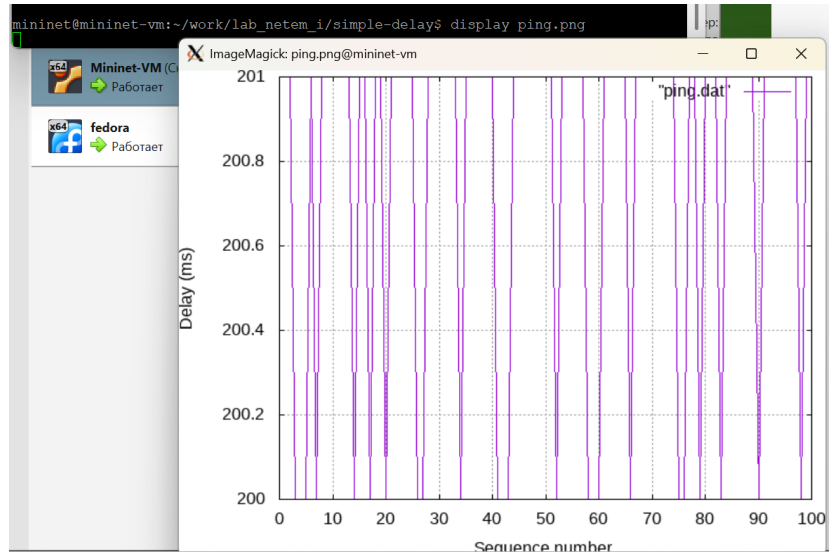


Рис. 10: рис. 28

11. Разработайте скрипт для вычисления на основе данных файла ping.dat минимального, среднего, максимального и стандартного отклонения времени приёма-передачи.

```

GNU nano 4.8                                rtt.py
with open('ping.dat', 'r') as f:
    s = []

    for line in f:
        line = line.strip()
        if not line:
            continue

        parts = line.split()
        s.append(float(parts[1]))

avg = sum(s) / len(s)
std = (sum((i - avg) ** 2 for i in s) / (len(s) - 1)) ** 0.5

print(
    f"min: {min(s)}\n"
    f"max: {max(s)}\n"
    f"avg: {avg}\n"
    f"std: {std}"
)

```

Рис. 11: рис. 29

12. Очистите каталог от результатов проведения экспериментов:

```

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

```

5. Выводы

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.