

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4

дисциплина: Моделирование сетей передачи данных

Студент: Саинт-Амур Измаэль

Группа: НПИбд-01-20

МОСКВА

2023 г.

Постановка задачи

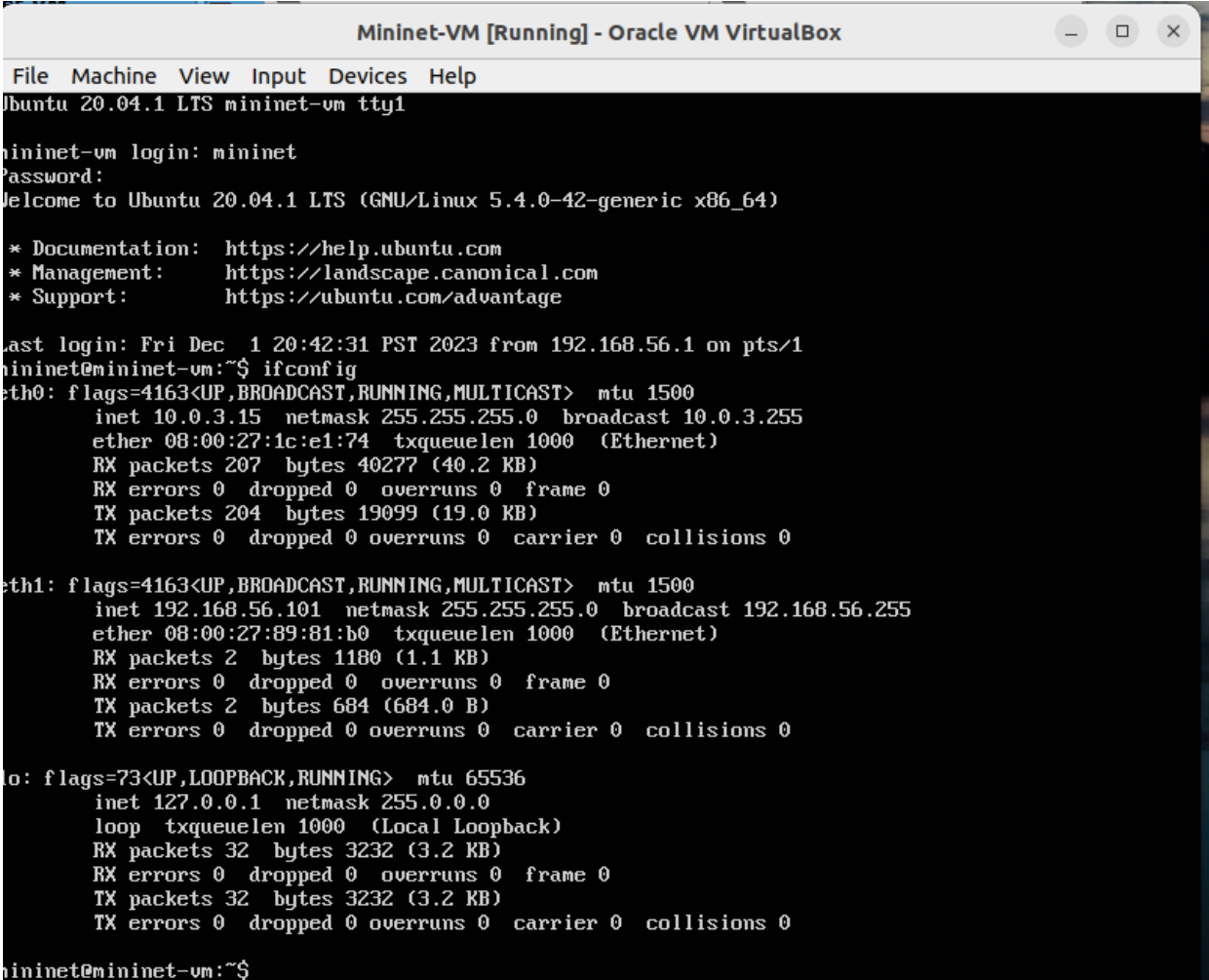
Основной целью работы является знакомство с NETEM — инструментом для тестирования производительности приложений в виртуальной сети, а также получение навыков проведения интерактивного и воспроизводимого экспериментов по измерению задержки и её дрожания (jitter) в моделируемой сети в среде Mininet.

Выполнение работы

4.4. Последовательность выполнения работы

4.4.1. Запуск лабораторной топологии

1. Запустите виртуальную среду с mininet.



```
Mininet-VM [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
ubuntu 20.04.1 LTS mininet-vm tty1

mininet-vm login: mininet
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

Last login: Fri Dec 1 20:42:31 PST 2023 from 192.168.56.1 on pts/1
mininet@mininet-vm:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.0.3.15  netmask 255.255.255.0  broadcast 10.0.3.255
    ether 08:00:27:1c:e1:74  txqueuelen 1000  (Ethernet)
    RX packets 207  bytes 40277 (40.2 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 204  bytes 19099 (19.0 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.56.101  netmask 255.255.255.0  broadcast 192.168.56.255
    ether 08:00:27:89:81:b0  txqueuelen 1000  (Ethernet)
    RX packets 2  bytes 1180 (1.1 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 2  bytes 684 (684.0 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 32  bytes 3232 (3.2 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 32  bytes 3232 (3.2 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

mininet@mininet-vm:~$ _
```

2. Из основной ОС подключитесь к виртуальной машине:

1 ssh -Y [mininet@192.168.56.101](ssh://mininet@192.168.56.101)

```

marc@Ismael:~$ ssh -Y mininet@192.168.56.101
mininet@192.168.56.101'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

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat Dec  9 07:09:51 2023
mininet@mininet-vm:~$

```

3. 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 $DISPLAY
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 570a4a78766b4fe091509bad42a75d1e
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth add mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 570a4a78766b4fe091509bad42a75d1e
root@mininet-vm:~# ^C
root@mininet-vm:~# logout
mininet@mininet-vm:~$

```

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

```
sudo mn --topo=single,2 -x
```

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

5. На хостах h1 и h2 введите 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.

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

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

```
mininet@mininet-vm:~$ 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:10.0
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

```
root@mininet-vm:~# xauth add mininet-vm
MIT-MAGIC_COOKIE_1 295a0ad8e35d1
```

```
"host: h2" (on mininet-vm)
```

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

```
"host: h1" (on mininet-vm)
```

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

```

"host: h1" (on mininet-vm)
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 4e:85:07:b1:db:05 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 1419 bytes 3719160 (3.7 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1419 bytes 3719160 (3.7 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

```

"host: h2" (on mininet-vm)
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 16:05:8e:e7:62:45 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 1535 bytes 3729060 (3.7 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1535 bytes 3729060 (3.7 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

```

root@mininet-vm:/home/mininet# ping 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=9.29 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.490 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.115 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.099 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.114 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.127 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.113 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.119 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.106 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.113 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.118 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.098 ms
^C
--- 10.0.0.2 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11230ms
rtt min/avg/max/mdev = 0.098/0.908/9.293/2.530 ms
root@mininet-vm:/home/mininet#

```

4.4.2. Интерактивные эксперименты

4.4.2.1. 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.

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

```
1 sudo tc qdisc add dev h1-eth0 root netem delay 100ms
```

Здесь:

- 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.

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).

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

```
1 sudo tc qdisc add dev h2-eth0 root netem delay 100ms
```

4. 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# sudo tc qdisc add dev h1-eth0 root netem delay 100ms
root@mininet-vm:/home/mininet# ping 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=104 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=102 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
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=103 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=100 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=100 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=101 ms
^C
--- 10.0.0.2 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 14026ms
rtt min/avg/max/mdev = 100.374/101.189/104.129/0.974 ms
root@mininet-vm:/home/mininet#
```

```
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

```

rtt min/avg/max/mdev = 100.374/101.189/104.129/0.974 ms
root@mininet-vm:/home/mininet# ping 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=205 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=202 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=206 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=203 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=202 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=202 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=203 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=203 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=202 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=201 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=203 ms
^C
--- 10.0.0.2 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11020ms

```

4.4.2.2. Изменение задержки в эмулируемой глобальной сети

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

```
1 sudo tc qdisc change dev h1-eth0 root netem delay 50ms
```

и для получателя h2:

```
1 sudo tc qdisc change dev h2-eth0 root netem delay 50ms
```

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

```



```

root@mininet-vm:/home/mininet# ping 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=104 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=103 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=101 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=102 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=102 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=102 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=103 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=102 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=101 ms
^C
--- 10.0.0.2 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 14040ms
rtt min/avg/max/mdev = 100.639/101.836/103.543/0.779 ms
root@mininet-vm:/home/mininet#

```

4.4.2.3. 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:
`1 sudo tc qdisc del dev h1-eth0 root netem`
 Для получателя h2:
`1 sudo tc qdisc del dev h2-eth0 root netem`
2. Проверьте, что соединение между хостом h1 и хостом h2 не имеет явно установленной 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# sudo tc qdisc del dev h2-eth0 root netem
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 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=5.93 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=1.22 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.457 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.134 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.193 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.152 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.145 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.229 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.103 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.132 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.243 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.822 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.116 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.127 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.115 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=0.103 ms
^C
--- 10.0.0.2 ping statistics ---
16 packets transmitted, 16 received, 0% packet loss, time 15302ms
rtt min/avg/max/mdev = 0.103/0.639/5.931/1.398 ms
root@mininet-vm:/home/mininet#

```

4.4.2.4. Добавление значения дрожания задержки в интерфейс

подключения к эмулируемой глобальной сети

В сетях нет постоянной задержки. Она может варьироваться в зависимости от других потоков трафика, конкурирующих за тот же путь. Джиттер (jitter)

—

это изменение времени задержки. Параметры задержки описываются в терминах теории вероятностей средним значением μ , стандартным отклонением σ и корреляцией. По умолчанию NETEM использует равномерное распределение,

так что задержка находится в пределах $\mu \pm \sigma$. Параметр корреляции управляет

отношением между последовательными псевдослучайными значениями.

1. При необходимости восстановите конфигурацию интерфейсов по умолчанию

на узлах h1 и h2:

```
1 sudo tc qdisc del dev h1-eth0 root netem
```

```
1 sudo tc qdisc del dev h2-eth0 root netem
```

2. Добавьте на узле h1 задержку в 100 мс со случайным отклонением 10 мс:

```
1 sudo tc qdisc add dev h1-eth0 root netem delay 100ms 10ms
```

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
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=98.1 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=103 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=103 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=94.8 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=95.7 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=98.8 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=96.2 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=106 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=105 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=97.6 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=111 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=106 ms
^C
--- 10.0.0.2 ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 12036ms
rtt min/avg/max/mdev = 94.789/101.582/110.623/4.784 ms
root@mininet-vm:/home/mininet#
```

4.4.2.5. 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%:

```
1 sudo tc qdisc add dev h1-eth0 root netem delay 100ms 10ms
```

↪ 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).

4. Восстановите конфигурацию интерфейса по умолчанию на узле h1.

```
root@mininet-vm:/home/mininet# sudo tc qdisc change dev h1-eth0 root netem delay 100ms 10ms 25%
root@mininet-vm:/home/mininet# ping 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=93.9 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=98.3 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=93.4 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=103 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=110 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=99.7 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=98.6 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=107 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=91.1 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=93.4 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=97.8 ms
^C
--- 10.0.0.2 ping statistics ---
14 packets transmitted, 13 received, 7.14286% packet loss, time 13027ms
rtt min/avg/max/mdev = 91.081/99.314/110.390/5.467 ms
root@mininet-vm:/home/mininet#
```

4.4.2.6. Распределение задержки в интерфейсе подключения

к эмулируемой глобальной сети

NETEM позволяет пользователю указать распределение, которое описывает, как задержки изменяются в сети. В реальных сетях передачи данных задержки

неравномерны, поэтому при моделировании может быть удобно использовать некоторое случайное распределение, например, нормальное, парето или паретонормальное.

1. При необходимости восстановите конфигурацию интерфейсов по умолчанию

на узлах h1 и h2.

2. Задайте нормальное распределение задержки на узле h1 в эмулируемой сети:

```
1 sudo tc qdisc add dev h1-eth0 root netem delay 100ms 20ms
```

```
↪ distribution normal
```

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.

Используйте для этого команду ping на терминале хоста h1 с параметром

-с 10.

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 20ms distribution normal
root@mininet-vm:/home/mininet# ping 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=83.0 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=80.9 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=133 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=70.5 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=72.6 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=141 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=109 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=95.9 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=100 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=69.0 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=105 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=74.0 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=124 ms
```

5. Завершите работу mininet в интерактивном режиме, введя в интерфейсе mininet:

1 mininet> exit

```
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 8 terms
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 2345.884 seconds
mininet@mininet-vm:~$
```

4.4.3. Воспроизведение экспериментов

4.4.3.1. Предварительная подготовка

1. Обновите репозитории программного обеспечения на виртуальной машине:

1 sudo apt-get update

2. Установите пакет geee — понадобится для просмотра файлов png:

1 sudo apt install geeqie

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

1 mkdir -p ~/work/lab_netem_i/exрname

Здесь exрname может принимать значения simple-delay, change-delay, jitter-delay, correlation-delay и т.п.

4. Для каждого случая создайте скрипт для проведения эксперимента lab_netem_i.py и скрипт для визуализации результатов ping_plot.

```
mininet@mininet-vm:~$ sudo apt-get update
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Hit:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:5 http://security.ubuntu.com/ubuntu focal-security/main i386 Packages [686 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [2610 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [917 kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [402 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [915 kB]
Get:10 http://security.ubuntu.com/ubuntu focal-security/universe i386 Packages [634 kB]
```

```
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-caplet gnome-control-center
  gnome-control-center-data gnome-control-center-faces gnome-keyring
```

4.3.2. Добавление задержки для интерфейса, подключающегося

к эмулируемой глобальной сети

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

1. В виртуальной среде mininet в своём рабочем каталоге с проектами создайте

каталог simple-delay и перейдите в него:

```
mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_i/expname
mininet@mininet-vm:~$ mkdir -p ~/work/lab_netem_i/simple-delay
mininet@mininet-vm:~$ cd ~/work/lab_netem_i/simple-delay
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ mc

[1]+  Stopped                  mc
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nano lab_netem_i.py
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ cd
```

Создаёте скрипт для эксперимента 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' )
```

3. В отчёте поясните содержание скрипта lab_netem_i.py. В каких строках скрипта задается значение задержки для интерфейса хоста? Каким образом формируется файл с результатами эксперимента для последующего построения графиков, какие значения в нём размещены?

4. Создаёте скрипт для визуализации ping_plot результатов эксперимента:

```

GNU nano 4.8                                     ping_p
#!/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

```

. Задайте права доступа к файлу скрипта:

1 `chmod +x ping_plot`

6. Создайте Makefile для управления процессом проведения эксперимента:

```

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

```

```

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

```

7. Выполните эксперимент:

1 `make`

8. Продемонстрируйте построенный в результате выполнения скриптов

график.

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

```
1 make ping.png
```

10. Продемонстрируйте построенный в результате график.

11. Разработайте скрипт для вычисления на основе данных файла ping.dat минимального, среднего, максимального и стандартного отклонения времени приёма-передачи. Добавьте правило запуска скрипта в Makefile. Продемонстрируйте работу скрипта с выводом значений на экран или в отдельный файл.

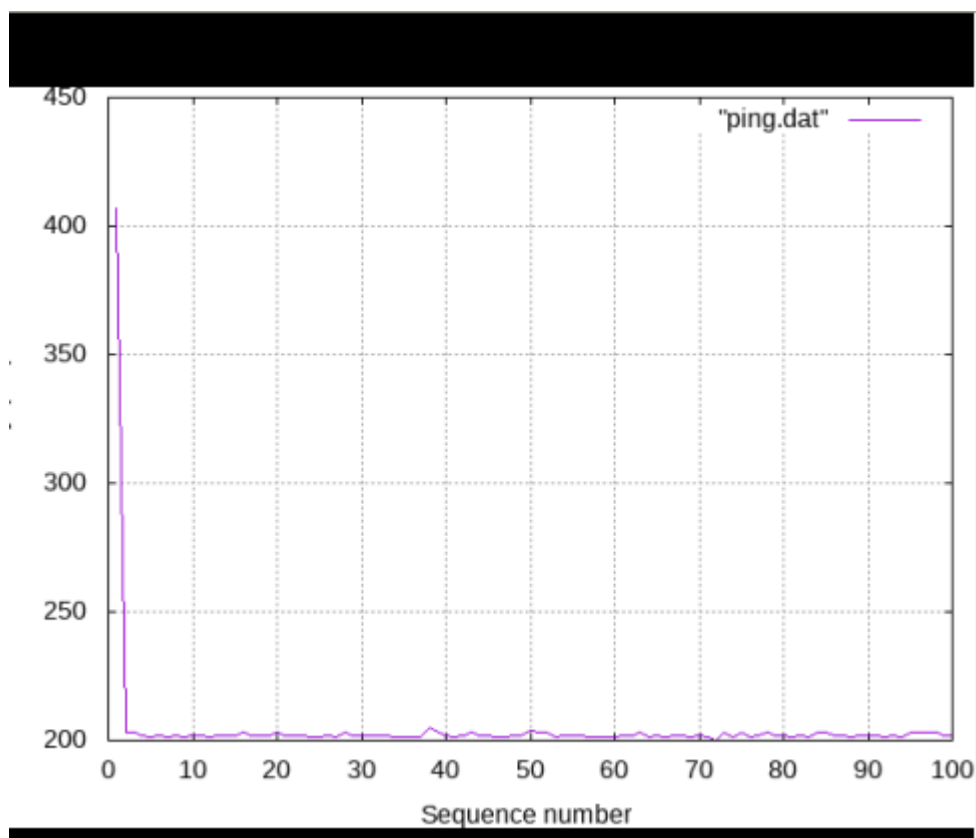
12. Очистите каталог от результатов проведения экспериментов:

```
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')
```

Left	File	Command	Options	Right			
<-	../lab_netem_i/simple-delay	-.[^]>		<-	../lab_netem_i/simple-delay	-.[^]>	
.n	Name	Size	Modify time	.n	Name	Size	Modify time
/..		UP--DIR	Dec 9 08:04	/..		UP--DIR	Dec 9 08:04
	Makefile	159	Dec 9 08:37		Makefile	159	Dec 9 08:37
	lab_netem_i.py	1169	Dec 9 08:43		lab_netem_i.py	1169	Dec 9 08:43
	ping.dat	692	Dec 9 08:45		ping.dat	692	Dec 9 08:45
	*ping_plot	163	Dec 9 08:31		*ping_plot	163	Dec 9 08:31
UP - -DIR				UP - -DIR			
2901M/7492M (38%)				2901M/7492M (38%)			

Hint: Want your plain shell? Press C-o, and get back to MC with C-o again.

Hint: Want your plain shell? Press C-o, and get back to MC with C-o again.



Выводы

В ходе полученные навыки ознакомления 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.

.