

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра теории вероятностей и кибербезопасности**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №4**

*дисциплина: Моделирование сетей передачи данных*

Студент: Быстров Глеб

Группа: НПИбд-01-20

**МОСКВА**

2023 г.

## **Цель работы**

В данной лабораторной работе мне будет необходимо познакомиться с 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.

## Описание процесса выполнения работы

### Запуск лабораторной топологии

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 (рис. 1).

```
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 3df09a7b02077183195b201811affba6
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth list $DISPLAY
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 6c186ea2c927a5ad8789d02d0b64393c
root@mininet-vm:~# xauth add mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 3df09a7b02077183195b201811affba6
root@mininet-vm:~# xauth list $DISPLAY
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 3df09a7b02077183195b201811affba6
root@mininet-vm:~#
```

Рис. 1. Копирование значения куки

2. Задайте простейшую топологию, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8 (рис. 2).

```
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:
```

Рис. 2. Простейшая топология

3. 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 (рис. 3-4).

```

"host: h1"@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 f6:50:a8:b7:9b:96 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 973 bytes 263416 (263.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 973 bytes 263416 (263.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Рис. 3. Хост h1

```

"host: h2"@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 fe:f7:c4:e9:87:b5 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 917 bytes 275204 (275.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 917 bytes 275204 (275.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Рис. 4. Хост h2

4. Проверьте подключение между хостами h1 и h2 с помощью команды ping с параметром -c 6. (рис. 5).

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

Min = 0.043 ms

Avg = 0.494 ms

Max = 2.729 ms

Mdev = 0.999 ms

```

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=3.59 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.198 ms

```

Рис. 5. Проверка подключения

## *Интерактивные эксперименты*

5. Добавление/изменение задержки в эмулируемой глобальной сети (рис. 6-7).

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.

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).

Min = 100.156 ms

Avg = 110.647 ms

Max = 101.179 ms

Mdev = 0.385 ms

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).

Min = 200.498 ms

Avg = 201.089 ms

Max = 201.673 ms

Mdev = 0.372 ms

```
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 -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=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=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 501ms
rtt min/avg/max/mdev = 100.156/100.647/101.179/0.385 ms
```

Рис. 6. Хост

```
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=201 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=200 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=201 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 500ms
rtt min/avg/max/mdev = 200.498/201.089/201.673/0.372 ms
```

Рис. 7. Хост

#### 6. Изменение задержки в эмулируемой глобальной сети (рис. 8).

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

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 h1-eth0 root netem delay 50ms
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=102 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=101 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=102 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 5012ms
rtt min/avg/max/mdev = 101.383/101.815/103.409/0.717 ms
```

Рис. 8. Хост

7. Восстановление исходных значений (удаление правил) задержки в эмулируемой глобальной сети (рис. 9).

Восстановите конфигурацию по умолчанию, удалив все правила, применённые к сетевому планировщику соответствующего интерфейса.

Для

отправителя `h1`:

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

Для получателя `h2`:

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

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 h1-eth0 root netem
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=8.39 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=1.72 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.163 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.060 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.048 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.054 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5084ms

```

Рис. 9. Хост

8. Добавление значения дрожания задержки в интерфейс подключения к эмулируемой глобальной сети (рис. 10).

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

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

Проверьте, что соединение от хоста h1 к хосту h2 имеет задержку 100 мс со случайным отклонением  $\pm 10$  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).

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=97.7 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=109 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=99.4 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=102 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=110 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5014ms
rtt min/avg/max/mdev = 97.667/103.510/110.072/4.620 ms

```

Рис. 10. Хост

9. Добавление значения корреляции для джиттера и задержки в интерфейс подключения к эмулируемой глобальной сети (рис. 11).

При необходимости восстановите конфигурацию интерфейсов по



умолчанию на узлах h1 и h2.

Добавьте на интерфейсе хоста h1 задержку в 100 мс с вариацией  $\pm 10$  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  $\pm 10$  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).

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 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=95.7 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=110 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=96.6 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=93.6 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=93.8 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=104 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=96.1 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=93.7 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=91.8 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=91.3 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=105 ms
---
```

Рис. 11. Хост

10. Распределение задержки в интерфейсе подключения к эмулируемой глобальной сети (рис. 12).

```
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 -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=136 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=100 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=71.3 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=104 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=120 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=83.3 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=143 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=111 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=111 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=117 ms
--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9910ms
```

Рис. 12. Хост

## Воспроизведение экспериментов

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

В виртуальной среде mininet в своём рабочем каталоге с проектами создайте каталог simple-delay и перейдите в него:

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

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

Создаёте скрипт для эксперимента lab\_netem\_i.py: (рис. 13).

```
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' )
```

Рис. 13. Скрипт

12. Выполните эксперимент:

Make

Продемонстрируйте построенный в результате выполнения скриптов график. (рис. 14).

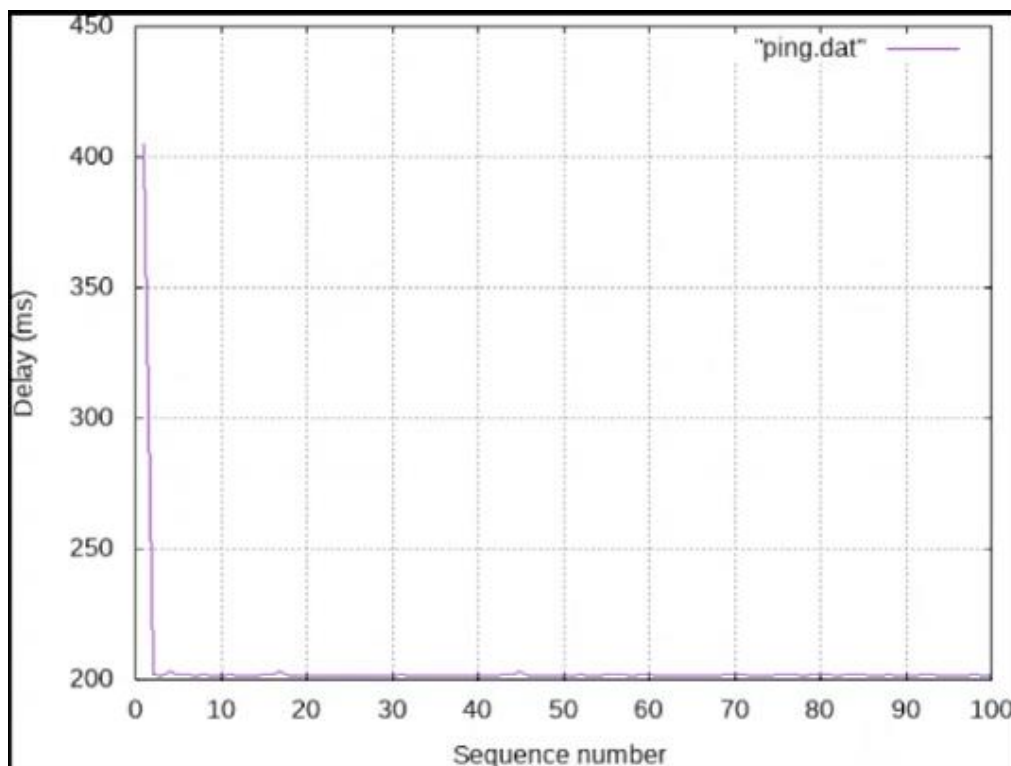


Рис. 14. График

13. Разработайте скрипт для вычисления на основе данных файла ping.dat минимального, среднего, максимального и стандартного отклонения временн приёма-передачи. Добавьте правило запуска скрипта в Makefile (рис. 15).

```
import statistics

def stat(d):
    tm = [float(line.split()[1]) for line in d]
    mint=min(tm)
    avgt=statistics.mean(tm)
    maxt=max(tm)
    stdt=statistics.stdev(tm)
    return mint, avgt, maxt, stdt

def main():
    with open('ping.dat', 'r') as file:
        d=file.readlines()

    mint, avgt, maxt, stdt = stat(d)
    print(f"Min: {mint} ms")
    print(f"Average: {avgt} ms")
    print(f"Max: {maxt} ms")
    print(f"Std: {stdt} ms")

if __name__ == "__main__":
    main()
```

Рис. 15. Скрипт

14. Продемонстрируйте работу скрипта с выводом значений на экран или в отдельный файл (рис. 16):

```
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ sudo python stats.py
Min: 200.0 ms
Average: 203.43 ms
Max: 408.0 ms
Std: 20.678518107231742 ms
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make clean
```

Рис. 16. Вывод

## Вывод

В данной лабораторной работе мне успешно удалось познакомиться 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.