

Лабораторная работа № 4

Эмуляция и измерение задержек в глобальных сетях

Старовойтов Егор Сергеевич

Содержание

Цель работы

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

Задание

1. Задайте простейшую топологию, состоящую из двух хостов и коммутатора с назначенной по умолчанию mininet сетью 10.0.0.0/8.
2. Проведите интерактивные эксперименты по добавлению/изменению задержки, джиттера, значения корреляции для джиттера и задержки, распределения времени задержки в эмулируемой глобальной сети.
3. Реализуйте воспроизводимый эксперимент по заданию значения задержки в эмулируемой глобальной сети. Постройте график.
4. Самостоятельно реализуйте воспроизводимые эксперименты по изменению задержки, джиттера, значения корреляции для джиттера и задержки, распределения времени задержки в эмулируемой глобальной сети. Постройте графики.

Теоретическое введение

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. NETEM реализован в Linux и состоит из двух частей: модуля ядра для организации очередей и утилиты

командной строки для его настройки. Между протоколом IP и сетевым устройством создаётся очередь с дисциплиной обслуживания. Дисциплина обслуживания очереди реализуется как объект с двумя интерфейсами. Один интерфейс ставит пакеты в очередь для отправки, а другой интерфейс отправляет пакеты на сетевое устройство. На основе дисциплины обслуживания очередей принимается решение о том, какие пакеты отправлять, какие пакеты задерживать и какие пакеты отбрасывать. Дисциплины обработки очередей можно разделить на бесклассовые и классовые. Бесклассовые дисциплины, в общем, получают данные, переупорядочивают, вносят задержку или уничтожают их. Такие дисциплины могут использоваться для указания ограничений интерфейса целиком, без какого-либо разделения по классам. Наиболее распространённой бесклассовой дисциплиной является FIFO (первым пришёл, первым обслужен). В NETEM и в Linux в целом эта дисциплина обслуживания очереди используется по умолчанию. Классовые дисциплины широко используются в случаях, когда тот или иной вид трафика необходимо обрабатывать по-разному. Примером классовой дисциплины может служить CBQ — Class Based Queueing (дисциплина обработки очередей на основе классов). Классы трафика организованы в дерево — у каждого класса есть не более одного родителя; класс может иметь множество потомков. Классы, которые не имеют родителей, называются корневыми. Классы, которые не имеют потомков, называются классами-ветками.

Выполнение лабораторной работы

1. Подключение по ssh и выдача прав графическим приложениям

```
via v22.11.0
) ssh -Y mininet@192.168.56.101
mininet@192.168.56.101's password:
Warning: No xauth data; using fake authentication data for X11 forwarding.
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 '22.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Thu Dec  5 10:28:23 2024
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 a752fa358e4cb4864df44322bbd801bc
mininet@mininet-vm:~$ MIT-MAGIC-COOKIE-1 a752fa358e4cb4864df44322bbd801bc
MIT-MAGIC-COOKIE-1: command not found
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# auth add mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 a752fa358e4cb4864df44322bbd801bc

Command 'auth' not found, did you mean:

  command 'iauth' from deb ircd-irc2 (2.11.2p3~dfsg-5build1)
  command 'oauth' from deb ruby-oauth (0.5.4-1)
  command 'xauth' from deb xauth (1:1.1-0ubuntu1)

Try: apt install <deb name>

root@mininet-vm:~# xauth add mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 a752fa358e4cb4864df44322bbd801bc
root@mininet-vm:~# logout
mininet@mininet-vm:~$
```

Step 1

2. Терминалы хостов h1 и h2.

```
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 fa:ed:9d:e7:df:33 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 1074 bytes 270456 (270.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1074 bytes 270456 (270.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet# whoami
root
root@mininet-vm:/home/mininet# tty
/dev/pts/8
root@mininet-vm:/home/mininet#
```

```
root@mininet-vm:/home/mininet# tty
/dev/pts/5
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 9a:67:53:f7:d5:fb 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 1261 bytes 285372 (285.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1261 bytes 285372 (285.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

Step 2

3. ping (h2 - слева, h1 - справа)

```
root@mininet-vm:/home/mininet# ping 10.0.0.1 -c 6
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data:
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=2.22 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.124 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.027 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.020 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.029 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.026 ms

--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5093ms
rtt min/avg/max/mdev = 0.020/0.407/2.218/0.810 ms
root@mininet-vm:/home/mininet#
```

```
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=2.00 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.032 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.030 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.030 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.027 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.022 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5097ms
rtt min/avg/max/mdev = 0.022/0.356/1.998/0.734 ms
root@mininet-vm:/home/mininet#
```

Step 3

4. ping с установленными задержками

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms
[cannot find device "h1-eth0"]
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 fa:ed:9d:e7:df:33 txqueuelen 1000 (Ethernet)
    RX packets 14 bytes 1260 (1.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 14 bytes 1260 (1.2 KB)
    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 2589 bytes 517844 (517.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2589 bytes 517844 (517.8 KB)
    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# ping 10.0.0.1 -c 6
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data:
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=205 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=206 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=206 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=203 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=205 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=203 ms

--- 10.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5007ms
rtt min/avg/max/mdev = 202.635/204.563/206.384/1.426 ms
root@mininet-vm:/home/mininet#
```

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h2-eth0 root netem delay 100ms
[cannot find device "h2-eth0"]
root@mininet-vm:/home/mininet#
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=201 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=202 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=201 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=202 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5017ms
rtt min/avg/max/mdev = 200.729/201.713/202.731/0.627 ms
root@mininet-vm:/home/mininet#
```

Step 4

5. ping с задержками 50мс

<pre>root@mininet-vn:/home/mininet# sudo tc qdisc change dev h2-eth0 root netem delay 50ms root@mininet-vn:/home/mininet# ping 10.0.0.1 -c 6 PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data. 64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=151 ms 64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=153 ms 64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=151 ms 64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=151 ms 64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=154 ms 64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=153 ms --- 10.0.0.1 ping statistics --- 6 packets transmitted, 6 received, 0% packet loss, time 5012ms rtt min/avg/max/mdev = 151.179/152.296/153.847/1.049 ms root@mininet-vn:/home/mininet#</pre>	<pre>root@mininet-vn:/home/mininet# sudo tc qdisc change dev h1-eth0 root netem delay 100ms root@mininet-vn:/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=151 ms 64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=151 ms 64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=150 ms 64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=155 ms 64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=153 ms 64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=155 ms --- 10.0.0.2 ping statistics --- 6 packets transmitted, 6 received, 0% packet loss, time 5016ms rtt min/avg/max/mdev = 150.440/152.593/155.152/1.885 ms root@mininet-vn:/home/mininet#</pre>
---	--

Step 5

6. Удаление задержек

<pre>root@mininet-vn:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms 10ms Cannot find device "h1-eth0" root@mininet-vn:/home/mininet# sudo tc qdisc add dev h2-eth0 root netem delay 100ms 10ms root@mininet-vn:/home/mininet# ping 10.0.0.1 -c 6 PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data. 64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=193 ms 64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=204 ms 64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=211 ms 64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=212 ms 64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=205 ms 64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=204 ms --- 10.0.0.1 ping statistics --- 6 packets transmitted, 6 received, 0% packet loss, time 5013ms rtt min/avg/max/mdev = 192.047/204.541/211.730/6.110 ms root@mininet-vn:/home/mininet#</pre>	<pre>root@mininet-vn:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms 10 root@mininet-vn:/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=193 ms 64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=199 ms 64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=187 ms 64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=198 ms 64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=189 ms 64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=200 ms --- 10.0.0.2 ping statistics --- 6 packets transmitted, 6 received, 0% packet loss, time 5007ms rtt min/avg/max/mdev = 187.200/194.522/200.338/5.065 ms root@mininet-vn:/home/mininet#</pre>
--	--

Step 6

7. Дрожание задержки

<pre>root@mininet-vn:/home/mininet# sudo tc qdisc add dev h2-eth0 root netem delay 100ms 10ms 25% Error: Exclusivity flag on, cannot modify. root@mininet-vn:/home/mininet# sudo tc qdisc change dev h2-eth0 root netem delay 100ms 10ms 25% root@mininet-vn:/home/mininet# ping 10.0.0.1 -c 6 PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data. 64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=203 ms 64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=205 ms 64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=200 ms 64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=202 ms 64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=205 ms 64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=204 ms --- 10.0.0.1 ping statistics --- 6 packets transmitted, 6 received, 0% packet loss, time 5015ms rtt min/avg/max/mdev = 200.158/203.147/205.446/1.706 ms root@mininet-vn:/home/mininet#</pre>	<pre>root@mininet-vn:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms 10ms root@mininet-vn:/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=193 ms 64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=199 ms 64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=187 ms 64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=198 ms 64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=189 ms 64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=200 ms --- 10.0.0.2 ping statistics --- 6 packets transmitted, 6 received, 0% packet loss, time 5007ms rtt min/avg/max/mdev = 187.200/194.522/200.338/5.065 ms root@mininet-vn:/home/mininet# sudo tc qdisc change dev h1-eth0 root netem delay 100ms 10ms 2 Command "changew" is unknown, try "tc qdisc help". root@mininet-vn:/home/mininet# sudo tc qdisc change dev h1-eth0 root netem delay 100ms 10ms 25% root@mininet-vn:/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=193 ms 64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=195 ms 64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=186 ms 64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=204 ms 64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=212 ms 64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=185 ms --- 10.0.0.2 ping statistics --- 6 packets transmitted, 6 received, 0% packet loss, time 5013ms rtt min/avg/max/mdev = 185.484/195.932/211.556/9.295 ms root@mininet-vn:/home/mininet#</pre>
---	---

Step 7

8. Дрожание задержки с вариациями

<pre>root@mininet-vn:/home/mininet# sudo tc qdisc change dev h2-eth0 root netem delay 100ms 20ms distribution normal root@mininet-vn:/home/mininet# ping 10.0.0.1 -c 6 PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data. 64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=147 ms 64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=106 ms 64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=177 ms 64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=213 ms 64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=235 ms 64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=168 ms --- 10.0.0.1 ping statistics --- 6 packets transmitted, 6 received, 0% packet loss, time 5017ms rtt min/avg/max/mdev = 106.206/174.150/234.862/42.024 ms root@mininet-vn:/home/mininet#</pre>	<pre>root@mininet-vn:/home/mininet# sudo tc qdisc change dev h1-eth0 root netem delay 100ms 30ms distribution normal root@mininet-vn:/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=187 ms 64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=213 ms 64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=186 ms 64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=255 ms 64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=189 ms 64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=188 ms --- 10.0.0.2 ping statistics --- 6 packets transmitted, 6 received, 0% packet loss, time 5010ms rtt min/avg/max/mdev = 186.011/203.074/254.947/25.059 ms root@mininet-vn:/home/mininet#</pre>
--	--

Step 8

9. Задержки с нормальным распределением

```
from mininet.log import setLogLevel, info
import time

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

    info( '*** Adding controller\n' )
    net.addController( 'c0' )
    info( '*** Adding hosts\n' )
    h1 = net.addHost( 'h1', ip='10.0.0.1' )
    h2 = net.addHost( 'h2', ip='10.0.0.2' )
    info( '*** Adding switch\n' )
    s1 = net.addSwitch( 's1' )

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

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

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

    time.sleep(10) # Wait 10 seconds
    info( '*** Ping\n' )
    h1.cmdPrint( 'ping -c 100', h2.IP(), '| grep "time=" | awk \'{print $5, $7}\'} | sed -e \'/s/time=//g\''
-e \'/s/icmp_seq=//g\' > ping.dat' )

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

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

lab_netem_i.py

44,0-1

Bot

"lab_netem_i.py" 47L, 1226C written

Step 9

10. lab_netem script

```
ll: 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
```

Step 10

11. Makefile

```
ll: 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
```

Step 11

12. Автоматизированное проведение эксперимента

```
*** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
t0
*** 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, $7}\'' | sed -e \'s/time=//g\' -e
s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
t0
*** Stopping 2 links
.
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
ping.png: ping.dat
make: ping.png: Command not found
make: *** [Makefile:6: ping.dat] Error 127
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ cat Makefile
all: ping.dat ping.png

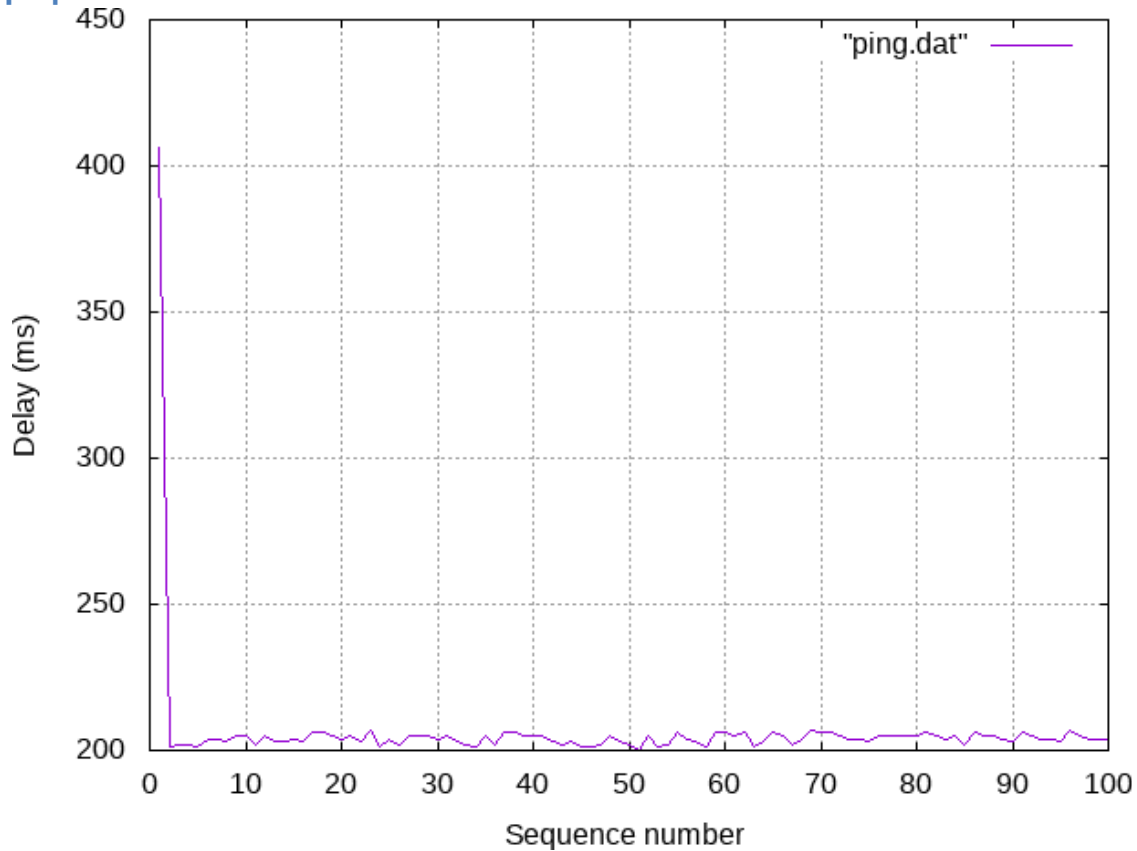
ping.dat:
    sudo python lab_netem_i.py
    sudo chown mininet:mininet ping.dat

ping.png: ping.dat
    ./ping_plot

clean:
    -rm -f *.dat *.png
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nvim Makefile
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make
./ping_plot
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ ls
lab_netem_i.py Makefile ping.dat ping_plot ping.png
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ nvim Makefile
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make ping.png
make: 'ping.png' is up to date.
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$ make clean
rm -f *.dat *.png
mininet@mininet-vm:~/work/lab_netem_i/simple-delay$
```

Step 12

Графики



ping

Выводы

Я познакомился 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.

Список литературы