

# Лабораторная работа №4

## Эмуляция и измерение задержек в глобальных сетях

---

Ким Реачна<sup>1</sup>

7 декабря, 2023, Москва, Россия

<sup>1</sup>Российский Университет Дружбы Народов

# Цели и 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.

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.

# Процесс выполнения лабораторной работы

---

# Запуск лабораторной топологии

```
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=2.09 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.247 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.045 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.046 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.082 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.046 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5105ms
rtt min/avg/max/mdev = 0.045/0.426/2.092/0.748 ms
```

Рис. 1: Проверка соединения от h1 к h2

# Добавление задержки в эмулируемой глобальной сети

```
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=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
--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5010ms
rtt min/avg/max/mdev = 100.543/100.781/101.103/0.233 ms
```

Рис. 2: Добавление задержку в 100 мс на h1

# Изменение задержки в эмулируемой глобальной сети

```
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=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=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=101 ms
--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 101.053/101.332/101.918/0.288 ms
```

Рис. 3: Добавление задержку в 50 мс на h1



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

```
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=1.03 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.356 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.129 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.124 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.050 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.053 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5082ms
rtt min/avg/max/mdev = 0.050/0.290/1.031/0.346 ms
```

Рис. 4: Восстановила конфигурацию по умолчанию для h1

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

```
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=110 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=104 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=94.5 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=94.6 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=98.2 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 94.499/101.029/110.419/5.781 ms
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
```

Рис. 5: Дрожания задержки

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

```
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 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=98.5 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=108 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=98.2 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=96.6 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=108 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 96.642/103.190/109.136/5.424 ms
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
```

Рис. 6: Корреляции для джиттера

# Распределение задержки в интерфейсе подключения к эмулируемой глобальной сети

```
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 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=91.2 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=104 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=94.1 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=87.0 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=109 ms

--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5003ms
rtt min/avg/max/mdev = 87.047/98.155/108.772/7.835 ms
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root netem
root@mininet-vm:/home/mininet#
```

Рис. 7: Распределение задержки в интерфейсе подключения к эмулируемой глобальной сети

# Добавление задержки для интерфейса, подключающегося к эмулируемой глобальной сети

```
GNU nano 4.2 /home/mininet/work/lab_netem_i/simple-delay/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()

    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()
```

Рис. 8: Скрипт lab\_netem\_i.py

# Добавление задержки для интерфейса, подключающегося к эмулируемой глобальной сети

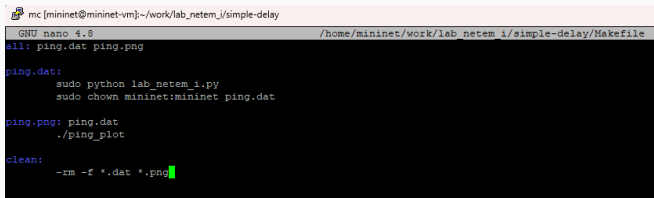


```
mc [mininet@mininet-vm]:~/work/lab_netem_i/simple-delay
GNU nano 4.8 /home/mininet/work/lab_netem_i/simple-delay/ping_plot
#!/usr/bin/gnuplot --persist

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

Рис. 9: Скрипт ping\_plot

# Добавление задержки для интерфейса, подключающегося к эмулируемой глобальной сети



The screenshot shows a terminal window with the following content:

```
mc [mininet@mininet-vm]:~/work/lab_netem_i/simple-delay
GNU nano 4.8 /home/mininet/work/lab_netem_i/simple-delay/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. 10: Makefile

# Добавление задержки для интерфейса, подключающегося к эмулируемой глобальной сети

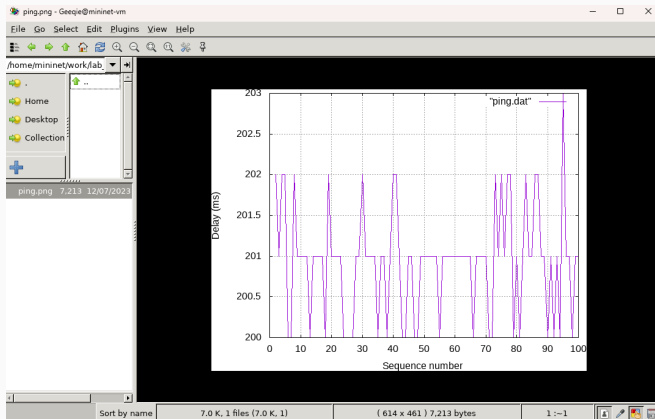


Рис. 11: График после удаления одной строки из ping.dat



# Реализование воспроизводимые эксперименты по изменению задержки

```
#!/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 50ms' )
    h2.cmdPrint( 'tc qdisc add dev h2-eth0 root netem delay 50ms' )

    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()
```

Рис. 12: Скрипт lab\_netem\_i.py для изменения задержки

# Реализование воспроизводимые эксперименты по изменению задержки

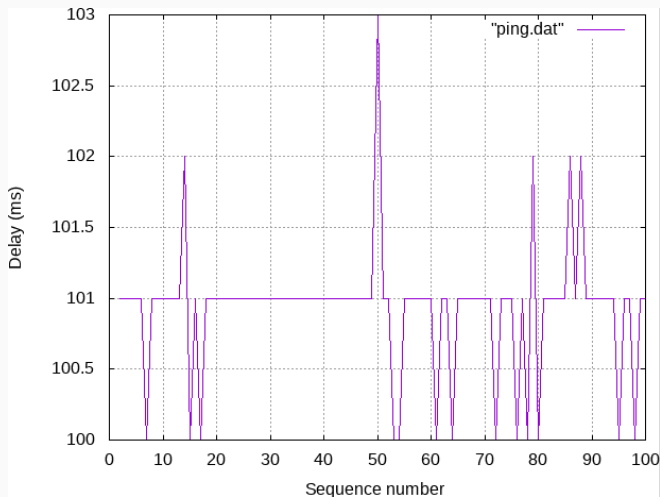


Рис. 13: График

# Реализование воспроизводимые эксперименты по джиттера, значения корреляции для джиттера и задержки

```
GNU nano 4.8 /home/mininet/work/lab_netem_i/jitter-delay/lab_netem_i.py
/usr/bin/cnv 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' )

    time.sleep(10) # Wait 10 seconds

    info( '*** Ping\n' )
    h1.cmdPrint( 'ping -c 100', h2.IP(), '| grep "time=" | awk \'{print $6, $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. 14: Скрипт lab\_netem\_i.py для джиттера

# Реализование воспроизводимые эксперименты по джиттера, значения корреляции для джиттера и задержки

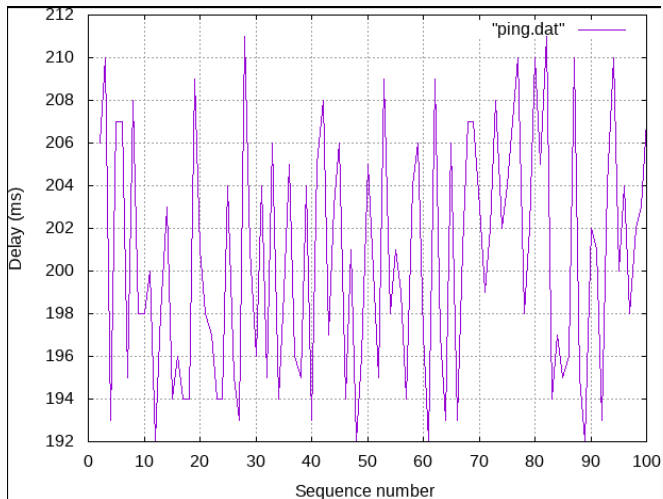


Рис. 15: График

# Реализование воспроизводимые эксперименты распределения задержки

```
GNU nano 4.8 /home/mininet/work/lab_netem_i/correlation-delay/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()

    info( '*** Set delay\n' )
    h1.cmdPrint( 'tc qdisc add dev h1-eth0 root netem delay 100ms 20ms distribution normal' )
    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()
```

Рис. 16: Скрипт lab\_netem\_i.py для распределения времени задержки в эмулируемой глобальной сети

# Реализование воспроизводимые эксперименты распределение задержки

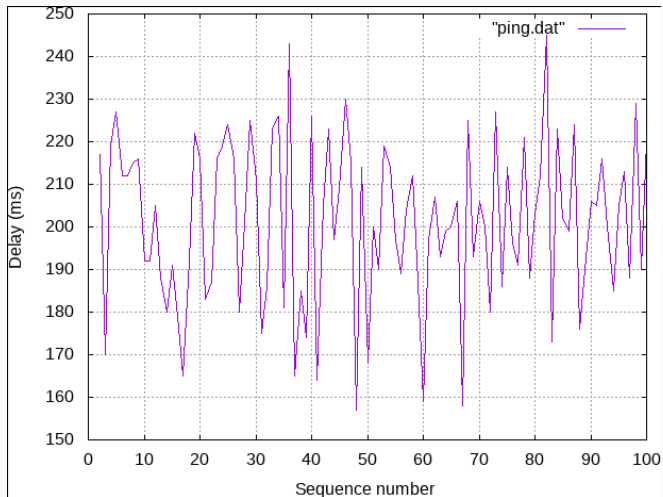


Рис. 17: График

## Выводы по проделанной работе

---

Я познакомилась 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.