

Отчёт по лабораторной работе №4

Эмуляция и измерение задержек в глобальных сетях.

**Выполнил: Матюхин Павел Андреевич,
НПИбд-01-22, 1132226527**

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Вывод	9

Список иллюстраций

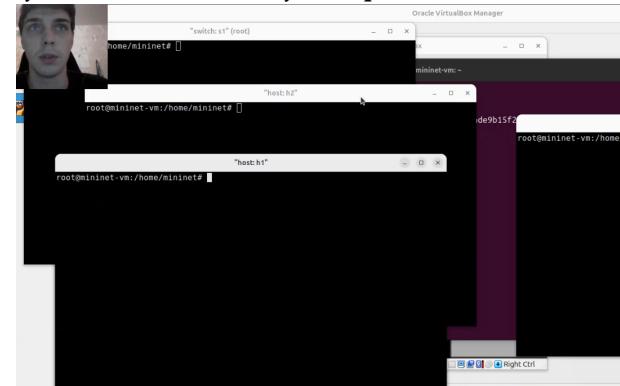
1 Цель работы

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.

2 Выполнение лабораторной работы

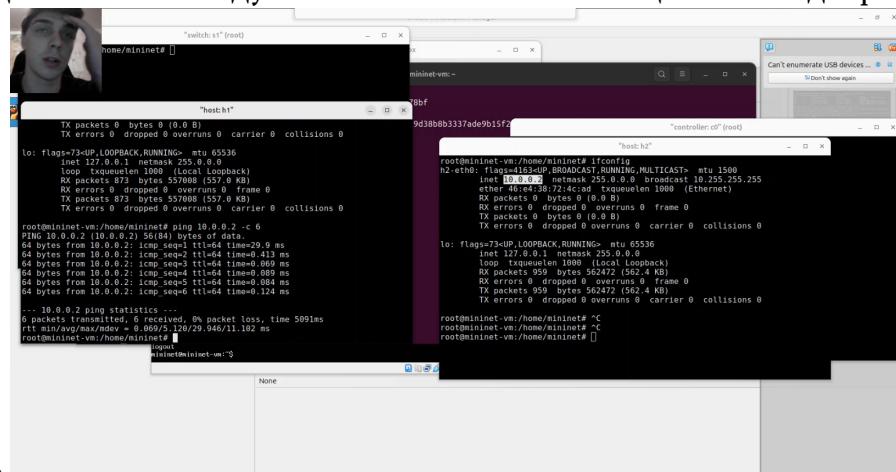
2.1

Задайте простейшую топологию, состоящую из двух хостов и коммутатора с



назначенной по умолчанию mininet сетью 10.0.0.0/8:

Проверьте подключение между хостами h1 и h2 с помощью команды ping с



параметром -c 6

```
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 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.000 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.000 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.000 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.000 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.000 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.000 ms
...
--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss
rtt min/avg/max/mdev = 100.255/101.000/101.000/0.000 ms
root@mininet-vm:/home/mininet#
```

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 50ms
root@mininet-vm:/home/mininet# 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=0.000 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.000 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.000 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.000 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.000 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.000 ms
...
--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss
rtt min/avg/max/mdev = 101.011/101.600/101.600/0.000 ms
root@mininet-vm:/home/mininet# ping 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
```

Измените задержку со 100 мс до 50 мс для отправителя h1 h2

Добавьте на интерфейсе хоста h1 задержку в 100 мс с вариацией ± 10 мс и

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 100ms 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=107 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=96.0 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=96.9 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=105 ms
```

значением корреляции в 25%

Задайте нормальное распределение задержки на узле h1 в эмулируемой сети

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root netem delay 0ms 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=75.3 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=71.3 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=98.1 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=71.0 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=55.5 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=123 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=85.5 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=111 ms
...
--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9018ms
rtt min/avg/max/mdev = 55.533/90.562/123.496/20.918 ms
root@mininet-vm:/home/mininet#
```

```

    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()
-- INSERT --

```

Создаёте скрипт для эксперимента lab_netem_i.py:

Создайте Makefile для управления процессом проведения эксперимента:

```

... ping.dat ping.png

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

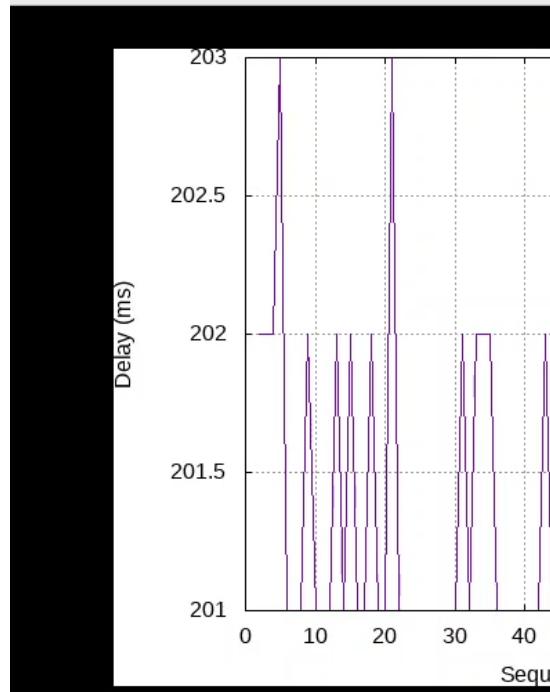
ping.png: ping.dat
    ./ping_plot

clean:
    -rm -f *.dat *.png
~
~
~
~
```

```

    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
./ping_plot
```

Выполните эксперимент



Продемонстрируйте построенный в результате график

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.

```

mininet@mininet-vm:~/work/lab_1$ ./usr/bin/env python3
# Read numbers from file (second column)
with open('ping.dat', 'r') as f:
    numbers = [float(line.strip().split()[1]) for line in f]

# Calculate stats
min_val = min(numbers)
max_val = max(numbers)
avg_val = sum(numbers) / len(numbers)
std_dev = (sum((x - avg_val) ** 2 for x in numbers) / len(numbers)) ** 0.5

# Print results
print(f"Minimum: {min_val:.3f} ms")
print(f"Average: {avg_val:.3f} ms")
print(f"Maximum: {max_val:.3f} ms")
print(f"Std dev: {std_dev:.3f} ms")
~
```

3 Вывод

Ознакомился с инструментом для измерения пропускной способности сети в режиме реального времени — iPerf3, а также получил навыки проведения интерактивного эксперимента по измерению пропускной способности моделируемой сети в среде Mininet.