

## Лабораторная работа №6

Настройка пропускной способности глобальной сети с помощью Token Bucket Filter

---

Хрусталев В.Н.

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Хрусталев Влад Николаевич
- студент
- Российский университет дружбы народов
- 1132222011@pfur.ru

## Цель работы

---

## Цель работы

---

Основной целью работы является знакомство с принципами работы дисциплины очереди Token Bucket Filter, которая формирует входящий/исходящий трафик для ограничения пропускной способности, а также получение навыков моделирования и исследования поведения трафика посредством проведения интерактивного и воспроизводимого экспериментов в Mininet.

## Выполнение лабораторной работы

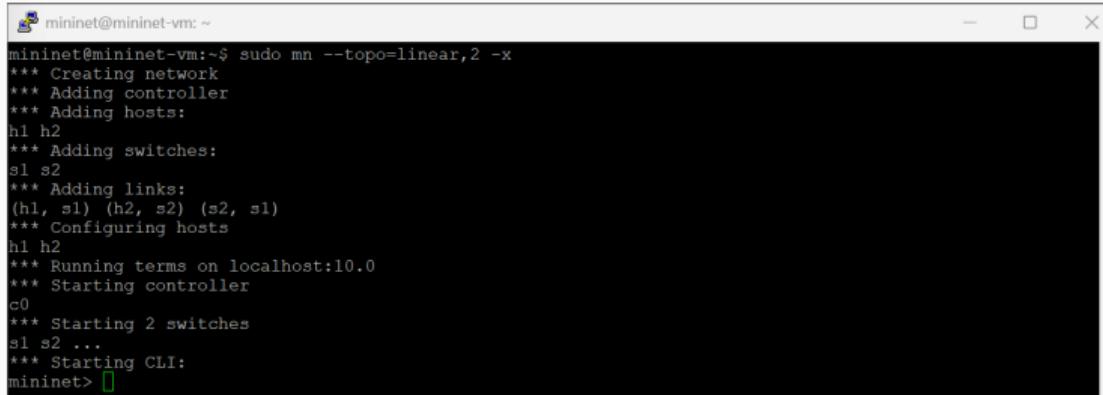
---

# Настройка X-соединения

```
Last login: Sat Nov 22 02:25:37 2023 from 192.168.56.1
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  25dbbe81293c4843b339fb714283e440
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth add mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  25dbbe81293c4843b339fb714283e440
root@mininet-vm:~# xauth list $DISPLAY
mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  25dbbe81293c4843b339fb714283e440
root@mininet-vm:~# exit
logout
mininet@mininet-vm:~$ █
```

Рис. 1: Настройка X-соединения

## Запуск простейшей топологии



```
mininet@mininet-vm:~$ sudo mn --topo=linear,2 -x
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1 s2
*** Adding links:
(h1, s1) (h2, s2) (s2, s1)
*** Configuring hosts
h1 h2
*** Running terms on localhost:10.0
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...
*** Starting CLI:
mininet> 
```

Рис. 2: Запуск простейшей топологии

# Информация на хостах и коммутаторах об интерфейсах

The image shows five terminal windows from a Linux host (Ubuntu 14.04 LTS) running a Mininet network. The windows are arranged in two columns. The left column contains three windows for hosts (host1, host2, host3) and the right column contains two windows for switches (switch1, switch2). Each window displays the output of the 'ifconfig' command.

**Host1 Information:**

```
root@mininet-vm:~/home/mininet# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.56.105 brd 255.255.255.0 broadcast 192.168.56.255
          ether 00:0c:29:b4:9e:53 txqueuelen 1000  (Ethernet)
            RX packets 1948 bytes 496874 (496.6 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 1930 bytes 985673 (985.6 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.0.2.15 brd 255.255.255.0 broadcast 10.0.2.255
          ether 00:0c:29:b4:9e:53 txqueuelen 1000  (Ethernet)
            RX packets 501 bytes 53261 (53.2 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 614 bytes 54355 (54.3 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 5829 bytes 1336189 (1.3 MB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 5829 bytes 1336189 (1.3 MB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

**Host2 Information:**

```
root@mininet-vm:~/home/mininet# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.56.105 brd 255.255.255.0 broadcast 192.168.56.255
          ether 00:0c:29:b4:9e:53 txqueuelen 1000  (Ethernet)
            RX packets 2048 bytes 509182 (509.1 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 2045 bytes 1038841 (1.0 MB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
          ether 00:0c:29:b4:9e:53 txqueuelen 1000  (Ethernet)
            RX packets 591 bytes 53261 (53.2 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 614 bytes 54355 (54.3 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 6886 bytes 1388537 (1.3 MB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 6886 bytes 1388537 (1.3 MB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

**Host3 Information:**

```
root@mininet-vm:~/home/mininet# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.56.105 brd 255.255.255.0 broadcast 192.168.56.255
          ether 00:0c:29:b4:9e:53 txqueuelen 1000  (Ethernet)
            RX packets 1948 bytes 496874 (496.6 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 1930 bytes 985673 (985.6 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
          ether 00:0c:29:b4:9e:53 txqueuelen 1000  (Ethernet)
            RX packets 501 bytes 53261 (53.2 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 614 bytes 54355 (54.3 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 1243 bytes 266844 (266.8 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 1243 bytes 266844 (266.8 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

**Switch1 Information:**

```
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 12:3a:33:03:65:34 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

h1-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.0.0.2 netmask 255.0.0.0 broadcast 10.255.255.255
          ether 12:3a:33:03:65:34 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
```

**Switch2 Information:**

```
root@mininet-vm:~/home/mininet# ifconfig
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.0.0.3 netmask 255.0.0.0 broadcast 10.255.255.255
          ether 46:45:a6:f4:e2:8d 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

h2-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.0.0.4 netmask 255.0.0.0 broadcast 10.255.255.255
          ether 46:45:a6:f4:e2:8d 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
```

Рис. 3: Информация на хостах и коммутаторах об интерфейсах

## Тестирование соединения между h1 и h2

```
"host: h1"@mininet-vm
root@mininet-vm:/home/mininet# ping -c 4 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=0.057 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.106 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.114 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.116 ms

--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3069ms
rtt min/avg/max/mdev = 0.057/0.098/0.116/0.024 ms
root@mininet-vm:/home/mininet# "
```

```
"host: h2"@mininet-vm
root@mininet-vm:/home/mininet# ping -c 4 10.0.0.1
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=4.46 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.155 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.148 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.108 ms

--- 10.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3052ms
rtt min/avg/max/mdev = 0.108/1.217/4.460/1.872 ms
root@mininet-vm:/home/mininet# "
```

# Проверка изначальной пропускной способности

```
X "host: h1"@mininet-vm
root@mininet-vm:/home/mininet# iperf3 -c 10.0.0.2
Connecting to host 10.0.0.2, port 5201
[ 7] local 10.0.0.1 port 50864 connected to 10.0.0.2 port 5201
[ ID] Interval           Transfer     Bitrate
[ 7]  0.00-1.00  sec   1.29 GBytes   11.1 Gbits/sec  0  8.22 MBytes
[ 7]  1.00-2.00  sec   1.23 GBytes   10.6 Gbits/sec  0  8.22 MBytes
[ 7]  2.00-3.00  sec   792 MBytes    6.65 Gbits/sec  0  8.22 MBytes
[ 7]  3.00-4.00  sec   726 MBytes    6.11 Gbits/sec  0  8.22 MBytes
[ 7]  4.00-5.01  sec   702 MBytes    5.82 Gbits/sec  0  8.22 MBytes
[ 7]  5.01-6.00  sec   762 MBytes    6.48 Gbits/sec  0  8.22 MBytes
[ 7]  6.00-7.00  sec   726 MBytes    6.07 Gbits/sec  0  8.22 MBytes
[ 7]  7.00-8.02  sec   838 MBytes    6.95 Gbits/sec  1  8.22 MBytes
[ 7]  8.02-9.00  sec   906 MBytes    7.70 Gbits/sec  0  8.22 MBytes
[ 7]  9.00-10.00 sec   839 MBytes    7.04 Gbits/sec  1  8.22 MBytes
[-----]
[ ID] Interval           Transfer     Bitrate
[ 7]  0.00-10.00 sec  8.67 GBytes   7.45 Gbits/sec  2          sender
[ 7]  0.00-10.01 sec  8.67 GBytes   7.44 Gbits/sec          receiver

iperf Done.
root@mininet-vm:/home/mininet#
```

```
X "host: h2"@mininet-vm
root@mininet-vm:/home/mininet# iperf3 -s
warning: this system does not seem to support IPv6 - trying IPv4
-----
Server listening on 5201
-----
Accepted connection from 10.0.0.1, port 50862
[ 7] local 10.0.0.2 port 5201 connected to 10.0.0.1 port 50864
[ ID] Interval           Transfer     Bitrate
[ 7]  0.00-1.00  sec   1.29 GBytes   11.1 Gbits/sec
[ 7]  1.00-2.02  sec   1.23 GBytes   10.3 Gbits/sec
[ 7]  2.02-3.00  sec   796 MBytes    6.82 Gbits/sec
[ 7]  3.00-4.01  sec   721 MBytes    6.01 Gbits/sec
[ 7]  4.01-5.00  sec   696 MBytes    5.88 Gbits/sec
[ 7]  5.00-6.00  sec   773 MBytes    6.50 Gbits/sec
[ 7]  6.00-7.00  sec   727 MBytes    6.10 Gbits/sec
[ 7]  7.00-8.00  sec   827 MBytes    6.92 Gbits/sec
[ 7]  8.00-9.00  sec   917 MBytes    7.71 Gbits/sec
[ 7]  9.00-10.00 sec   840 MBytes    7.05 Gbits/sec
[ 7]  10.00-10.01 sec  64.4 KBytes   89.3 Mbytes/sec
[-----]
[ ID] Interval           Transfer     Bitrate
[ 7]  0.00-10.01 sec  8.67 GBytes   7.44 Gbits/sec          receiver

Server listening on 5201
```

# Ограничение скорости на конечных хостах и тест

```
X "host: h1"@mininet-vm
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root tbft rate 10gbit b
urst 5000000 limit 15000000
root@mininet-vm:/home/mininet# iperf3 -c 10.0.0.2
Connecting to host 10.0.0.2, port 5201
[ 7] local 10.0.0.1 port 50868 connected to 10.0.0.2 port 5201
[ ID] Interval Transfer Bitrate Retr Cwnd
[ 7]  0.00-1.00  sec 1.06 GBytes 9.09 Gbits/sec 0 8.11 MBytes
[ 7] 1.00-2.01  sec 1.07 GBytes 9.17 Gbits/sec 0 8.11 MBytes
[ 7] 2.01-3.00  sec 800 MBytes 6.74 Gbits/sec 0 8.11 MBytes
[ 7] 3.00-4.01  sec 778 MBytes 6.49 Gbits/sec 0 8.11 MBytes
[ 7] 4.01-5.00  sec 815 MBytes 6.87 Gbits/sec 1 8.11 MBytes
[ 7] 5.00-6.00  sec 789 MBytes 6.64 Gbits/sec 0 8.11 MBytes
[ 7] 6.00-7.00  sec 664 MBytes 5.57 Gbits/sec 0 8.11 MBytes
[ 7] 7.00-8.00  sec 761 MBytes 6.39 Gbits/sec 0 8.11 MBytes
[ 7] 8.00-9.00  sec 831 MBytes 6.95 Gbits/sec 0 8.11 MBytes
[ 7] 9.00-10.01 sec 785 MBytes 6.52 Gbits/sec 0 8.11 MBytes
[ ID] Interval Transfer Bitrate Retr
[ 7]  0.00-10.01 sec 8.21 GBytes 7.04 Gbits/sec 1
[ 7]  0.00-10.02 sec 8.21 GBytes 7.04 Gbits/sec
sender
receiver
iperf Done.
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root
root@mininet-vm:/home/mininet#
```

```
X "host: h2"@mininet-vm
Server listening on 5201
-----
Accepted connection from 10.0.0.1, port 50868
[ 7] local 10.0.0.2 port 5201 connected to 10.0.0.1 port 50868
[ ID] Interval Transfer Bitrate
[ 7]  0.00-1.00  sec 1.06 GBytes 9.11 Gbits/sec
[ 7] 1.00-2.00  sec 1.05 GBytes 9.05 Gbits/sec
[ 7] 2.00-3.00  sec 818 MBytes 6.87 Gbits/sec
[ 7] 3.00-4.00  sec 769 MBytes 6.45 Gbits/sec
[ 7] 4.00-5.00  sec 824 MBytes 6.91 Gbits/sec
[ 7] 5.00-6.01  sec 782 MBytes 6.53 Gbits/sec
[ 7] 6.01-7.01  sec 662 MBytes 5.56 Gbits/sec
[ 7] 7.01-8.01  sec 758 MBytes 6.36 Gbits/sec
[ 7] 8.01-9.02  sec 840 MBytes 6.93 Gbits/sec
[ 7] 9.02-10.00 sec 778 MBytes 6.64 Gbits/sec
[ 7] 10.00-10.02 sec 9.38 MBytes 7.11 Gbits/sec
[ ID] Interval Transfer Bitrate
[ 7]  0.00-10.02 sec 8.21 GBytes 7.04 Gbits/sec
receiver
-----
Server listening on 5201
-----
^Ciperf3: interrupt - the server has terminated
root@mininet-vm:/home/mininet#
```

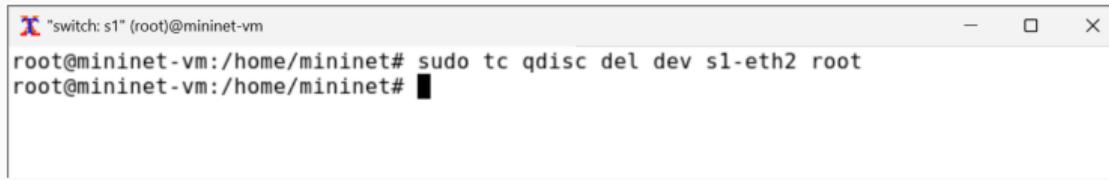
# Ограничение скорости на коммутаторе и тест

The screenshot displays four terminal windows from a Linux environment, likely Mininet, illustrating the configuration of a switch port and its impact on network performance.

- Terminal 1 (switch s1):** Shows the command `sudo tc qdisc add dev s1-eth2 root tbf rate 10gbit burst 5000000 limit 15000000` being run to set up traffic shaping rules on interface s1-eth2.
- Terminal 2 (host h1):** Shows the command `iperf3 -c 10.0.0.2` being run to perform a bandwidth test from host h1 to host h2. The output shows a stable transfer rate of approximately 8.37 MBytes/sec over 10 seconds.
- Terminal 3 (host h2):** Shows the command `iperf3 -s` being run to start an iperf3 server on host h2. The output shows a connection from host h1 and a stable transfer rate of approximately 7.28 Gbytes/sec over 10 seconds.
- Terminal 4 (switch s1):** Shows the command `tc filter add dev s1-eth2 bpf ...` being run to apply a BPF filter to the queue discipline on interface s1-eth2.

Рис. 7: Ограничение скорости на коммутаторе и тест

## Удаление модифицированной конфигурации на коммутаторе s1



```
"switch: s1" (root)@mininet-vm
root@mininet-vm:/home/mininet# sudo tc qdisc del dev s1-eth2 root
root@mininet-vm:/home/mininet#
```

Рис. 8: Удаление модифицированной конфигурации на коммутаторе s1

## Добавление NETEM правила задержки пакетов

The screenshot shows two terminal windows side-by-side. The left window, titled "switch: s1" (root)@mininet-vm, displays the command: `sudo tc qdisc add dev s1-eth2 root handle 1: netem delay 10ms`. The right window, titled "host: h1" (root)@mininet-vm, shows the output of a ping test from host h1 to IP 10.0.0.2. The ping command fails with an invalid argument error. Subsequent successful pings show four round-trip times (RTTs) of approximately 11-14 ms. The final statistics show 4 packets transmitted, 4 received, 0% loss, and an average RTT of 12.296 ms.

```
switch: s1" (root)@mininet-vm
root@mininet-vm:/home/mininet# sudo tc qdisc add dev s1-eth2 root handle 1: netem delay 10ms
root@mininet-vm:/home/mininet# 

"host: h1" (root)@mininet-vm
root@mininet-vm:/home/mininet# ping -c 10.0.0.2
ping: invalid argument: '10.0.0.2'
root@mininet-vm:/home/mininet# ping -c 4 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=14.7 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=11.6 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=11.7 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=11.2 ms

--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 11.183/12.296/14.686/1.393 ms
root@mininet-vm:/home/mininet#
```

Рис. 9: Добавление NETEM правила задержки пакетов

## Добавление 2-го правила ограничения скорости и тест

```
root@mininet-vm:~/home/mininet# sudo tc qdisc add dev s1-eth2 root handle 1: metem d
elay 10ms
root@mininet-vm:~/home/mininet# sudo tc qdisc add dev s1-eth2 parent 1: handle 2: tb
f rate 2gbit burst 1000000 limit 2000000
root@mininet-vm:~/home/mininet# █
```

```
X root@mininet-vm:~/home/mininet# iperf3 -c 10.0.0.2
Connecting to host 10.0.0.2, port 5201
[ 7] local 10.0.0.1 port 50876 connected to 10.0.0.2 port 5201
[ ID] Interval           Transfer     Bitrate      Retr
[ 7]  0.00-1.00   sec    191 MBytes   1.68 Gbits/sec  270  3.50 MBbytes
[ 7]  1.00-2.00   sec    228 MBytes   1.91 Gbits/sec  0   3.60 MBbytes
[ 7]  2.00-3.00   sec    160 MBytes   1.34 Gbits/sec  525  1.31 MBbytes
[ 7]  3.00-4.00   sec    111 MBytes   0.935 Gbits/sec  0   1.48 MBbytes
[ 7]  4.00-5.00   sec    121 MBytes   1.02 Gbits/sec  0   1.47 MBbytes
[ 7]  5.00-6.00   sec    100 MBytes   0.896 Gbits/sec  0   1.48 MBbytes
[ 7]  6.00-7.00   sec    112 MBytes   0.941 Gbits/sec  0   1.55 MBbytes
[ 7]  7.00-8.00   sec    124 MBytes   1.04 Gbits/sec  0   1.57 MBbytes
[ 7]  8.00-9.00   sec    131 MBytes   1.18 Gbits/sec  0   1.63 MBbytes
[ 7]  9.00-10.00  sec    140 MBytes   1.17 Gbits/sec  0   1.68 MBbytes
[ ...]
[ ID] Interval           Transfer     Bitrate      Retr
[ 7]  0.00-10.00  sec   1.41 GBytes  1.21 Gbits/sec  795
[ 7]  0.00-10.02  sec   1.40 GBytes  1.20 Gbits/sec
sender
receiver

iperf Done.
root@mininet-vm:~/home/mininet# █
```

```
X "mininet-vm" (root)@mininet-vm
root@mininet-vm:~/home/mininet# █
```

```
X root@mininet-vm:~/home/mininet# iperf3 -s
Server listening on 5201
[ ...]
Accepted connection from 10.0.0.1, port 50874
[ 7] local 10.0.0.2 port 5201 connected to 10.0.0.1 port 50876
[ ID] Interval           Transfer     Bitrate      Retr
[ 7]  0.00-1.00   sec    180 MBytes   1.51 Gbits/sec
[ 7]  1.00-2.00   sec    228 MBytes   1.91 Gbits/sec
[ 7]  2.00-3.01   sec    159 MBytes   1.33 Gbits/sec
[ 7]  3.01-4.00   sec    112 MBytes   0.938 Gbits/sec
[ 7]  4.00-5.00   sec    122 MBytes   1.02 Gbits/sec
[ 7]  5.00-6.00   sec    121 MBytes   1.01 Gbits/sec
[ 7]  6.00-7.00   sec    118 MBytes   0.975 Gbits/sec
[ 7]  7.00-8.00   sec    124 MBytes   1.04 Gbits/sec
[ 7]  8.00-9.00   sec    131 MBytes   1.18 Gbits/sec
[ 7]  9.00-10.00  sec    139 MBytes   1.17 Gbits/sec
[ 7]  10.00-10.02  sec   128 KBytes  59.3 Mbites/sec
[ ...]
[ ID] Interval           Transfer     Bitrate      Retr
[ 7]  0.00-10.02  sec   1.49 GBytes  1.20 Gbits/sec
receiver

Server listening on 5201
[ ...]
```

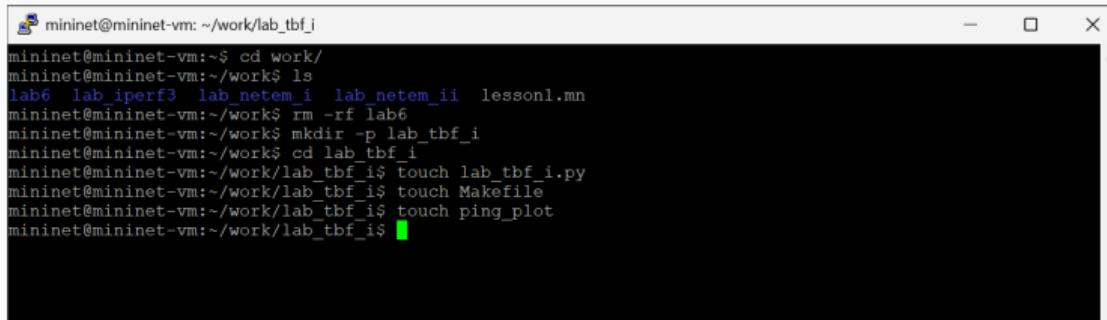
Рис. 10: Добавление 2-го правила ограничения скорости и тест

## Удаление модифицированной конфигурации на коммутаторе s1

```
X "switch: s1" (root)@mininet-vm
root@mininet-vm:/home/mininet# sudo tc qdisc add dev s1-eth2 root handle 1: netem d
elay 10ms
root@mininet-vm:/home/mininet# sudo tc qdisc add dev s1-eth2 parent 1: handle 2: tb
f rate 2gbit burst 1000000 limit 2000000
root@mininet-vm:/home/mininet# sudo tc qdisc del dev s1-eth2 root
root@mininet-vm:/home/mininet#
```

Рис. 11: Удаление модифицированной конфигурации на коммутаторе s1

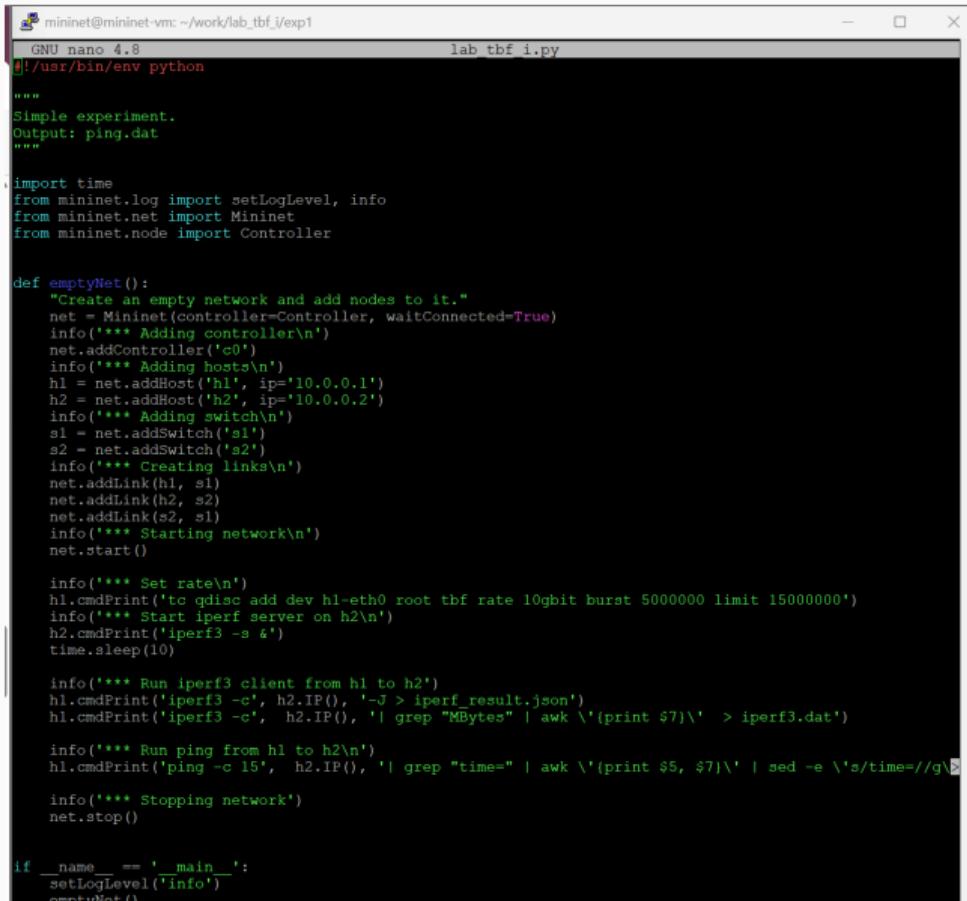
## Создание необходимых файлов



```
mininet@mininet-vm:~/work/lab_tbf_i
mininet@mininet-vm:~/work$ ls
lab6 lab_iperf3 lab_netem_i lab_netem_ii lesson1.mn
mininet@mininet-vm:~/work$ rm -rf lab6
mininet@mininet-vm:~/work$ mkdir -p lab_tbf_i
mininet@mininet-vm:~/work$ cd lab_tbf_i
mininet@mininet-vm:~/work/lab_tbf_i$ touch lab_tbf_i.py
mininet@mininet-vm:~/work/lab_tbf_i$ touch Makefile
mininet@mininet-vm:~/work/lab_tbf_i$ touch ping_plot
mininet@mininet-vm:~/work/lab_tbf_i$
```

Рис. 12: Создание необходимых файлов

## Листинг exp1/lab\_tbf\_i.py



```
mininet@mininet-vm: ~/work/lab_tbf_i/exp1
GNU nano 4.8                                     lab_tbf_i.py
#!/usr/bin/env python

"""
Simple experiment.
Output: ping.dat
"""

import time
from mininet.log import setLogLevel, info
from mininet.net import Mininet
from mininet.node import Controller

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')
    s2 = net.addSwitch('s2')
    info('*** Creating links\n')
    net.addLink(h1, s1)
    net.addLink(h2, s2)
    net.addLink(s2, s1)
    info('*** Starting network\n')
    net.start()

    info('*** Set rate\n')
    h1.cmdPrint('tc qdisc add dev h1-eth0 root tbf rate 10gbit burst 5000000 limit 15000000')
    info('*** Start iperf server on h2\n')
    h2.cmdPrint('iperf3 -s &')
    time.sleep(10)

    info('*** Run iperf3 client from h1 to h2')
    h1.cmdPrint('iperf3 -c', h2.IP(), '-J > iperf_result.json')
    h1.cmdPrint('iperf3 -c', h2.IP(), '| grep "MBytes" | awk \'(print $7)\' > iperf3.dat')

    info('*** Run ping from h1 to h2\n')
    h1.cmdPrint('ping -c 15', h2.IP(), '| grep "time=" | awk \'(print $5, $7)\' | sed -e \'s/time=/g\'')

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

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

## Листинг Makefile и ping\_plot

```
mininet@mininet-vm:~/work/lab_tbf_i/expl$ cat Makefile
all: ping.dat ping.png

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

ping.png:
    ./ping_plot

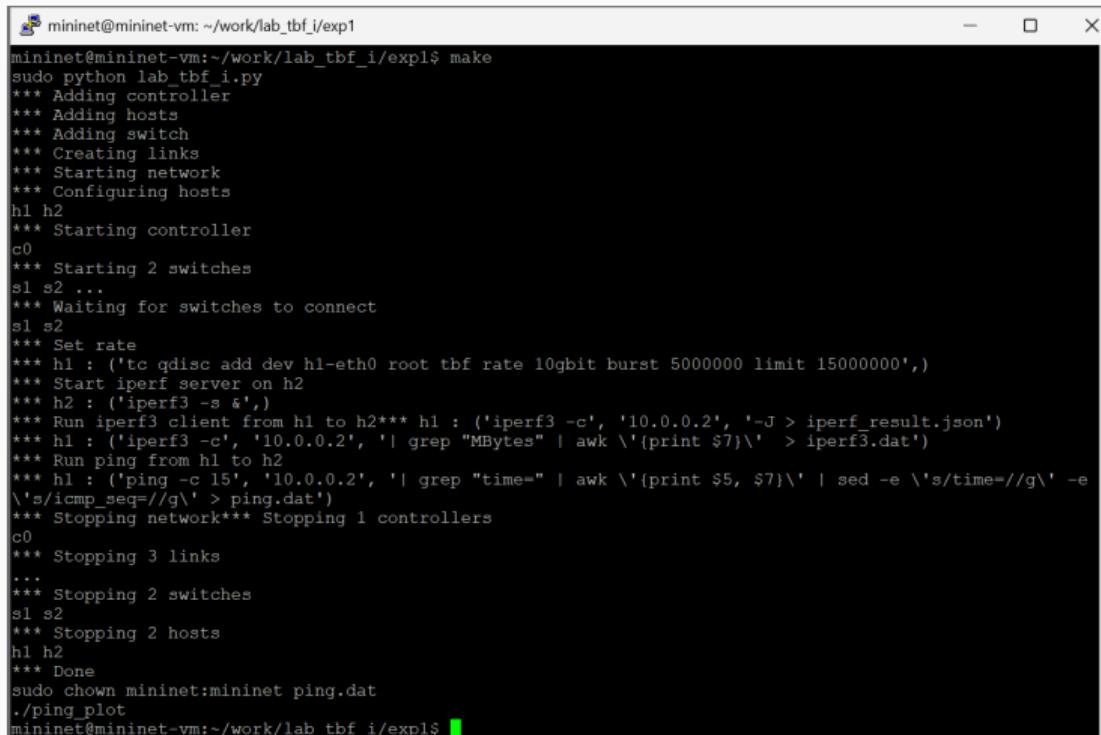
clean:
    -rm -f *.dat *.png
mininet@mininet-vm:~/work/lab_tbf_i/expl$ cat ping_plot
#!/usr/bin/gnuplot --persist

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

set terminal png crop
set output 'iperf3.png'
set xlabel "Packet number"
set ylabel "Rate (Gbytes/sec)"
set grid
plot "iperf3.dat" with lines
```

Рис. 14: Листинг Makefile и ping\_plot

## Запуск exp1



mininet@mininet-vm: ~/work/lab\_tbf\_i/exp1\$ make  
sudo python lab\_tbf\_i.py  
\*\*\* Adding controller  
\*\*\* Adding hosts  
\*\*\* Adding switch  
\*\*\* Creating links  
\*\*\* Starting network  
\*\*\* Configuring hosts  
h1 h2  
\*\*\* Starting controller  
c0  
\*\*\* Starting 2 switches  
s1 s2 ...  
\*\*\* Waiting for switches to connect  
s1 s2  
\*\*\* Set rate  
\*\*\* h1 : ('tc qdisc add dev h1-eth0 root tbf rate 10gbit burst 5000000 limit 15000000',)  
\*\*\* Start iperf server on h2  
\*\*\* h2 : ('iperf3 -s &',)  
\*\*\* Run iperf3 client from h1 to h2\*\*\* h1 : ('iperf3 -c', '10.0.0.2', '-J > iperf\_result.json')  
\*\*\* h1 : ('iperf3 -c', '10.0.0.2', '| grep "MBytes" | awk \'{print \$7}\' > iperf3.dat')  
\*\*\* Run ping from h1 to h2  
\*\*\* h1 : ('ping -c 15', '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 3 links  
...  
\*\*\* Stopping 2 switches  
s1 s2  
\*\*\* Stopping 2 hosts  
h1 h2  
\*\*\* Done  
sudo chown mininet:mininet ping.dat  
./ping\_plot  
mininet@mininet-vm:~/work/lab\_tbf\_i/exp1\$

Рис. 15: Запуск exp1

## График iperf3.png из exp1

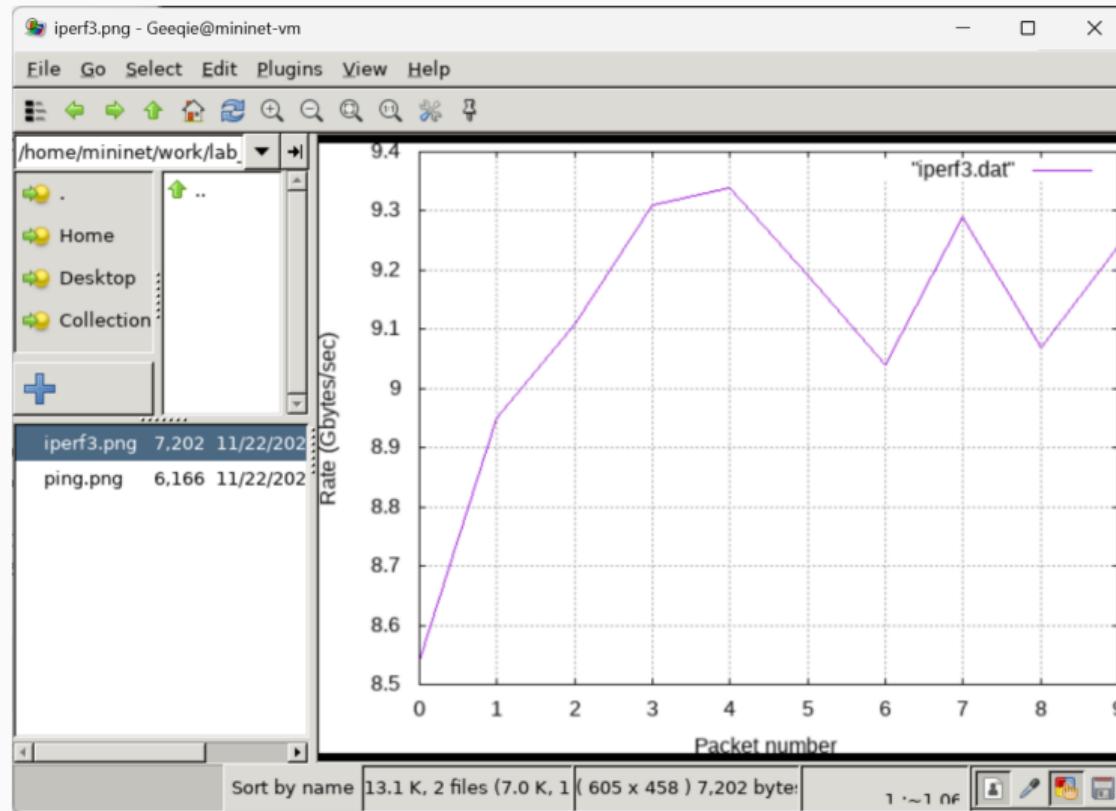


Рис. 16: График iperf3.png из exp1

## График ping.png из exp1

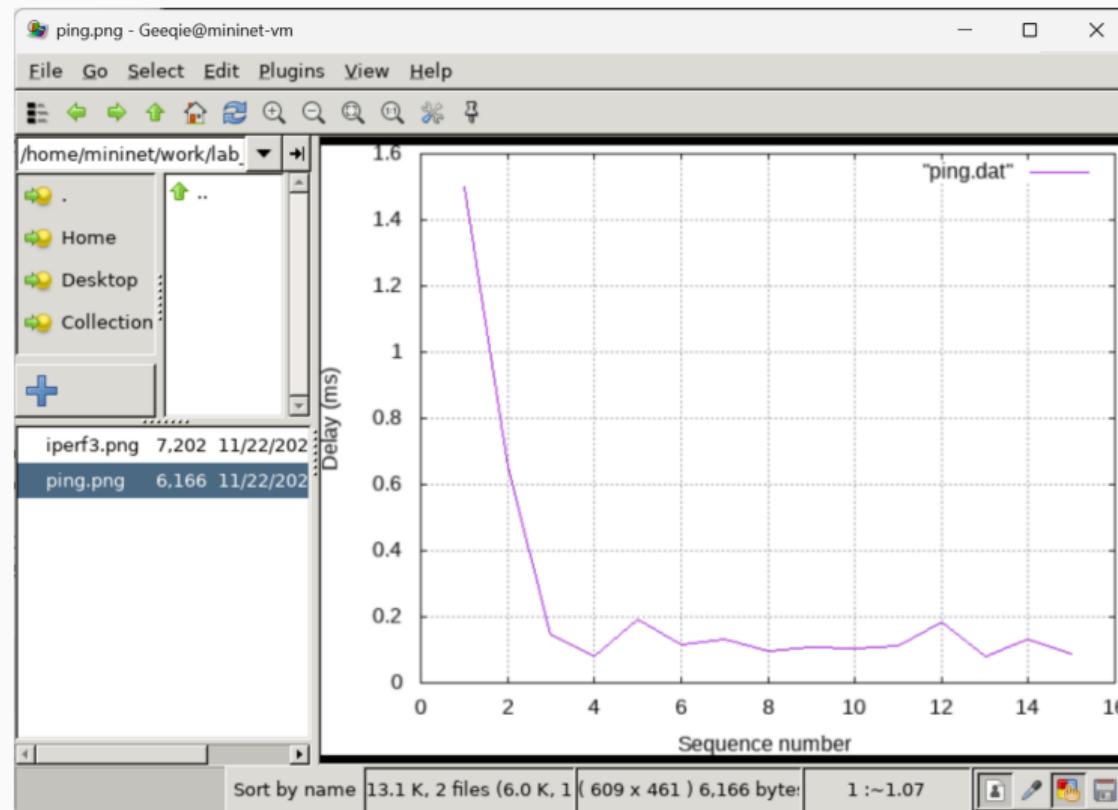


Рис. 17: График ping.png из exp1

## Листинг exp2/lab\_tbf\_i.py

```
mininet@mininet-vm: ~/work/lab_tbf_i/exp2
GNU nano 4.8                               lab_tbf_i.py                                Modified
#!/usr/bin/env python

"""
Simple experiment.
Output: ping.dat
"""

import time
from mininet.log import setLogLevel, info
from mininet.net import Mininet
from mininet.node import Controller

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')
    s2 = net.addSwitch('s2')
    info('*** Creating links\n')
    net.addLink(h1, s1)
    net.addLink(h2, s2)
    net.addLink(s2, s1)
    info('*** Starting network\n')
    net.start()

    info('*** Set rate\n')
    s1.cmdPrint('tc qdisc add dev s1-eth2 root tbf rate 10gbit burst 5000000 lim 15000000')
    info('*** Start iperf server on h2\n')
    h2.cmdPrint('iperf3 -s &')
    time.sleep(10)

    info('*** Run iperf3 client from h1 to h2')
    h1.cmdPrint('iperf3 -c', h2.IP(), '-J > iperf_result.json')
    h1.cmdPrint('iperf3 -c', h2.IP(), '| grep "MBytes" | awk \'{print $7}\' > iperf3.dat')

    info('*** Run ping from h1 to h2\n')
    h1.cmdPrint('ping -c 15', h2.IP(), '| grep "time=" | awk \'{print $5, $7}\' | sed -e \'/s/time=/g\'')
    info('*** Stopping network')
    net.stop()

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

## Запуск exp2

```
mininet@mininet-vm:~/work/lab_tbf_i/exp2$ make
sudo python lab_tbf_i.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
*** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...
*** Waiting for switches to connect
s1 s2
*** Set rate
*** s1 : ('tc qdisc add dev s1-eth2 root tbf rate 10gbit burst 5000000 limit 15000000',)
*** Start iperf server on h2
*** h2 : ('iperf3 -s &')
*** Run iperf3 client from h1 to h2
*** h1 : ('iperf3 -c', '10.0.0.2', '-J > iperf_result.json')
*** h1 : ('iperf3 -c', '10.0.0.2', '| grep "MBytes" | awk \'{print $7}\' > iperf3.dat')
*** Run ping from h1 to h2
*** h1 : ('ping -c 15', '10.0.0.2', '| grep "time=" | awk \'{print $5, $7}\' | sed -e \'s/time=/g\' -e \'s/icmp_seq=/g\' > ping.dat')
*** Stopping network
*** Stopping 1 controllers
c0
*** Stopping 3 links
...
*** Stopping 2 switches
s1 s2
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_tbf_i/exp2$ ls
iperf3.dat  iperf3.png  iperf_result.json  lab_tbf_i.py  Makefile  ping.dat  ping_plot  ping.png
mininet@mininet-vm:~/work/lab_tbf_i/exp2$
```

Рис. 19: Запуск exp2

## График iperf3.png из exp2

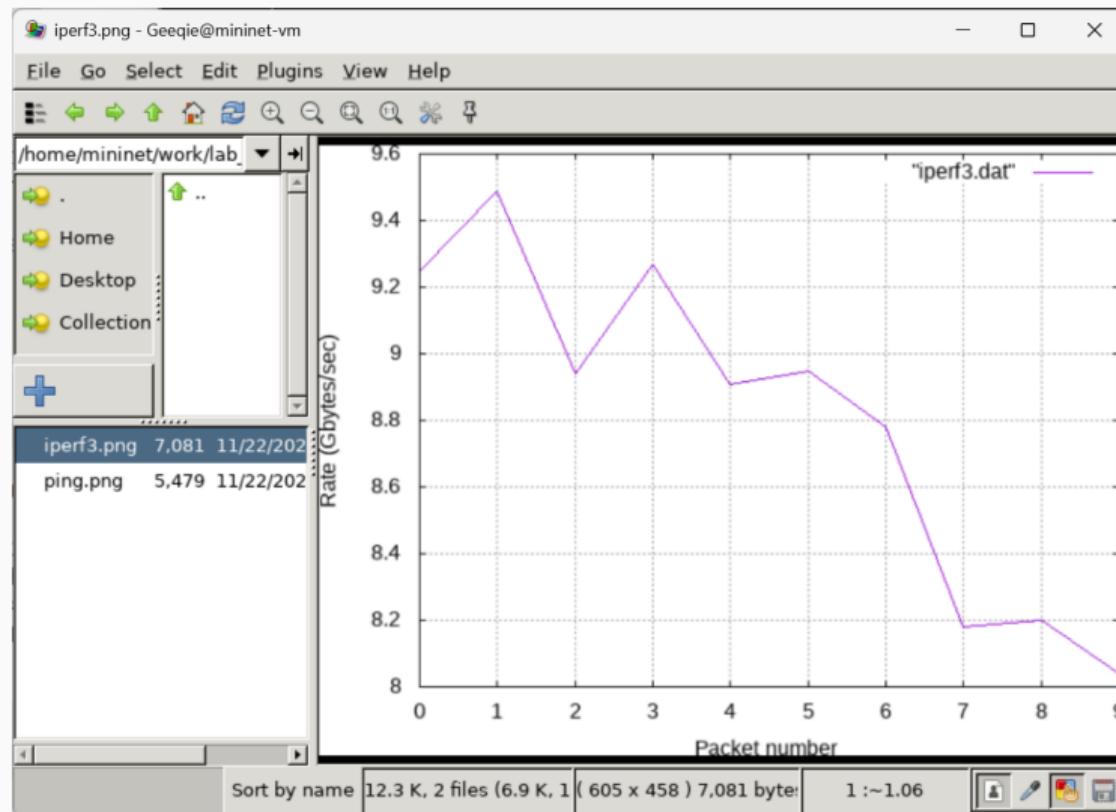


Рис. 20: График iperf3.png из exp2

## График ping.png из exp2

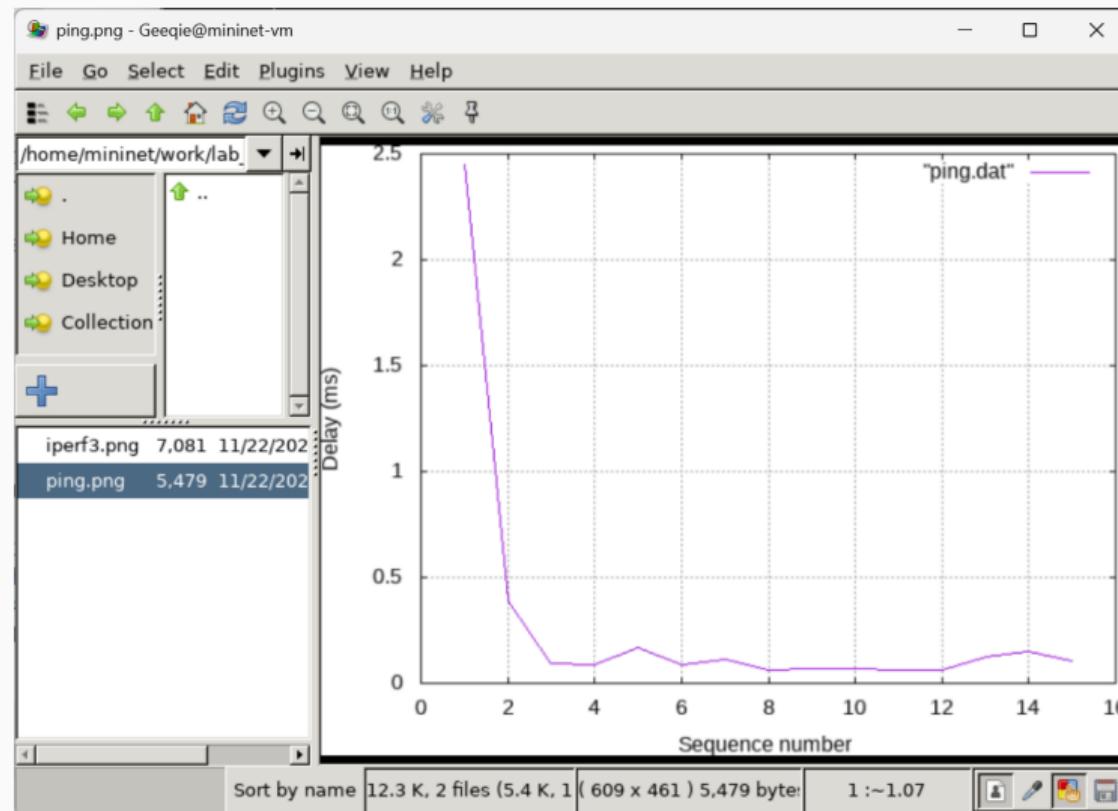


Рис. 21: График ping.png из exp2

## Листинг exp3/lab\_tbf\_i.py

```
mininet@mininet-vm: ~/work/lab_tbf_i/exp3          lab_tbf_i.py          Modified
GNU nano 4.8
#!/usr/bin/env python

"""
Simple experiment.
Output: ping.dat
"""

import time
from mininet.log import setLogLevel, info
from mininet.net import Mininet
from mininet.node import Controller

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')
    s2 = net.addSwitch('s2')
    info('*** Creating links\n')
    net.addLink(h1, s1)
    net.addLink(h2, s2)
    net.addLink(s2, s1)
    info('*** Starting network\n')
    net.start()

    info('*** Set rate\n')
    s1.cmdPrint('tc qdisc add dev s1-eth2 root handle 1: netem delay 10ms')
    s1.cmdPrint('tc qdisc add dev s1-eth2 root parent 1: tbf rate 2gbit burst 1000000 limit 2000000')
    info('*** Start iperf server on h2\n')
    h2.cmdPrint('iperf3 -s')
    time.sleep(10)

    info('*** Run iperf3 client from h1 to h2')
    h1.cmdPrint('iperf3 -e', h2.IP(), '-J > iperf_result.json')
    h1.cmdPrint('iperf3 -e', h2.IP(), '| grep "NBytes" | awk \'(print $7)\' > iperf3.dat')

    info('*** Run ping from h1 to h2\n')
    h1.cmdPrint('ping -c 15', h2.IP(), '| grep "time=" | awk \'(print $5, $7)\' | sed -e \'/s/time=/g\' -e \'/s/icmp_/')
    info('*** Stopping network')
    net.stop()

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

# Запуск exp3

```
mininet@mininet-vm:~/work/lab_tbf_i/exp3$ nano lab_tbf_i.py
mininet@mininet-vm:~/work/lab_tbf_i/exp3$ make
sudo python lab_tbf_i.py
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
*** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...
*** Waiting for switches to connect
s1 s2
*** Set rate
*** s1 : ('tc qdisc add dev s1-eth2 root handle 1: netem delay 10ms',)
*** s1 : ('tc qdisc add dev s1-eth2 root parent 1: handle 2: tbf rate 2gbit burst 1000000 limit 2000000',)
Error: duplicate "parent": "1:" is the second value.
*** Start iperf server on h2
*** h2 : ('iperf3 -s &')
*** Run iperf3 client from h1 to h2*** h1 : ('iperf3 -c', '10.0.0.2', '-J > iperf_result.json')
*** h1 : ('iperf3 -c', '10.0.0.2', '| grep "MBytes" | awk \'{print $7}\' > iperf3.dat')
*** Run ping from h1 to h2
*** h1 : ('ping -c 15', '10.0.0.2', '| grep "time=" | awk \'{print $5, $7}\' | sed -e \'s/time=/\\n\\t/ -e \'s/icmp_seq//\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 3 links
*** Stopping 2 switches
s1 s2
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_tbf_i/exp3$ ls
iperf3.dat iperf3.png iperf_result.json lab_tbf_i.py Makefile ping.dat ping_plot ping.png
mininet@mininet-vm:~/work/lab_tbf_i/exp3$
```

## График iperf3.png из exp3

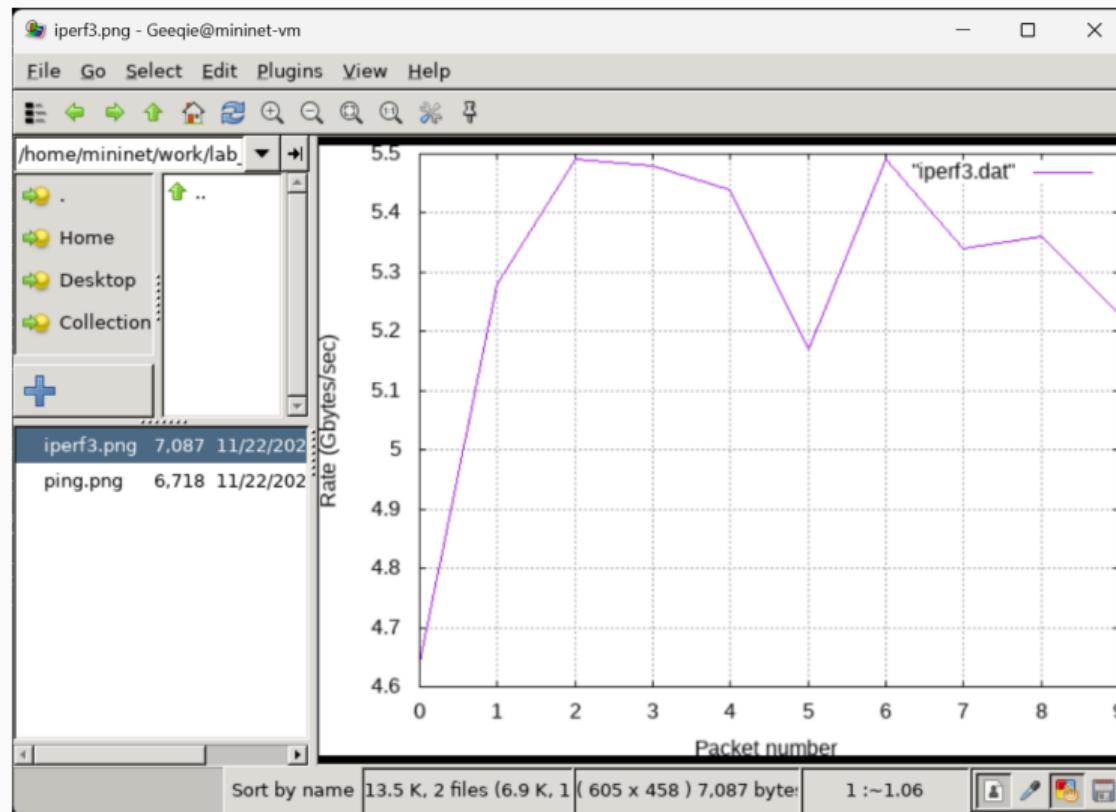


Рис. 24: График iperf3.png из exp3

## График ping.png из exp3

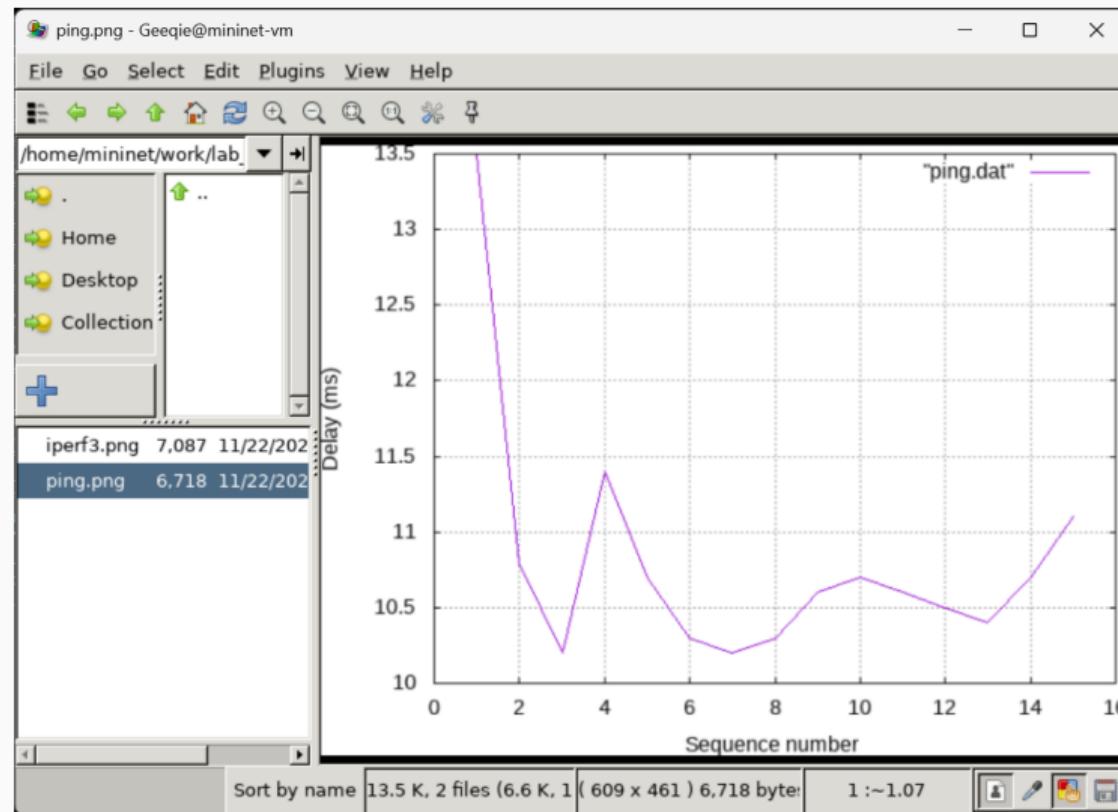


Рис. 25: График ping.png из exp3

## Выводы

---

## Выводы

---

В результате выполнения работы я познакомился с принципами работы Token Bucket Filter, а также получили навыки моделирования и исследования поведения трафика посредством проведения интерактивного и воспроизводимого экспериментов в Mininet.

## Список литературы

---

## Список литературы

---

1. Mininet [Электронный ресурс]. Mininet Project Contributors. URL: <http://mininet.org/> (дата обращения: 06.10.2025).

