

# **Отчёт по лабораторной работе №6**

## **Моделирование сетей передачи данных**

**Настройка пропускной способности глобальной сети с помощью Token  
Bucket Filter**

Выполнил: Махорин Иван Сергеевич,  
НПИбд-02-21, 1032211221

# Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Вывод	27
4	Список литературы. Библиография	28

# Список иллюстраций

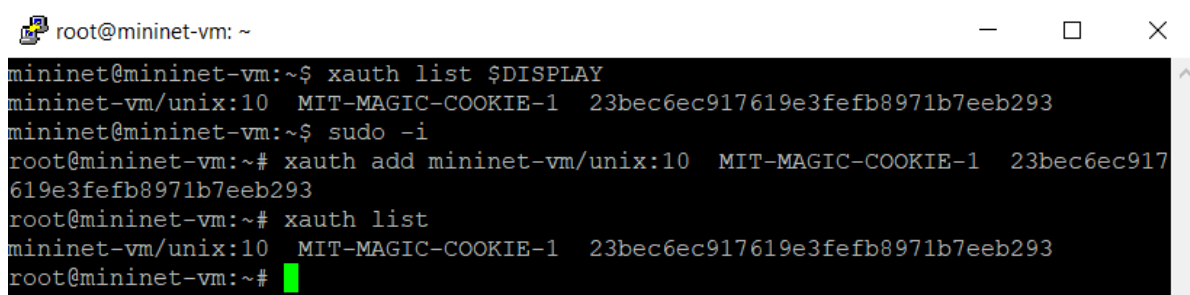
2.1	Исправление прав запуска X-соединения в виртуальной машине mininet . . . . .	5
2.2	Создание топологии с двумя хостами и двумя коммутаторами . .	6
2.3	Отображение информации сетевых интерфейсов и IP-адресов . .	7
2.4	Проверка подключения между хостами h1 и h2 . . . . .	8
2.5	Запуск iPerf3 в режиме сервера на хосте h2 . . . . .	8
2.6	Запуск iPerf3 в режиме клиента на хосте h1 . . . . .	9
2.7	Остановка iPerf3 . . . . .	10
2.8	Изменение пропускной способности хоста h1 . . . . .	10
2.9	Установка значения всплеска при ограничении скорости для филь- тра tbf . . . . .	11
2.10	Запуск iPerf3 в режиме сервера на хосте h2 . . . . .	11
2.11	Запуск iPerf3 в режиме клиента на хосте h1 . . . . .	12
2.12	Остановка iPerf3 . . . . .	13
2.13	Удаление модифицированной конфигурации на хосте h1 . . . . .	13
2.14	Применение правила ограничения скорости tbf . . . . .	13
2.15	Запуск iPerf3 в режиме сервера на хосте h2 . . . . .	14
2.16	Запуск iPerf3 в режиме клиента на хосте h1 . . . . .	14
2.17	Остановка iPerf3 . . . . .	15
2.18	Удаление модифицированной конфигурации на коммутаторе s1 .	15
2.19	Объединение NETEM и TBF . . . . .	16
2.20	Проверка задержки . . . . .	16
2.21	Добавление второго правила на коммутаторе s1 . . . . .	16
2.22	Запуск iPerf3 в режиме сервера на хосте h2 . . . . .	17
2.23	Запуск iPerf3 в режиме клиента на хосте h1 . . . . .	17
2.24	Остановка iPerf3 . . . . .	18
2.25	Удаление модифицированной конфигурации на коммутаторе s1 .	18
2.26	Создание необходимых каталогов . . . . .	19
2.27	Написание 1 скрипта . . . . .	20
2.28	Написание 2 скрипта . . . . .	21
2.29	Написание 3 скрипта . . . . .	21
2.30	Выполнение скриптов . . . . .	22
2.31	Изменение параметров . . . . .	23
2.32	Выполнение скриптов . . . . .	24
2.33	График №1 . . . . .	25
2.34	График №2 . . . . .	26

# 1 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.

## 2 Выполнение лабораторной работы

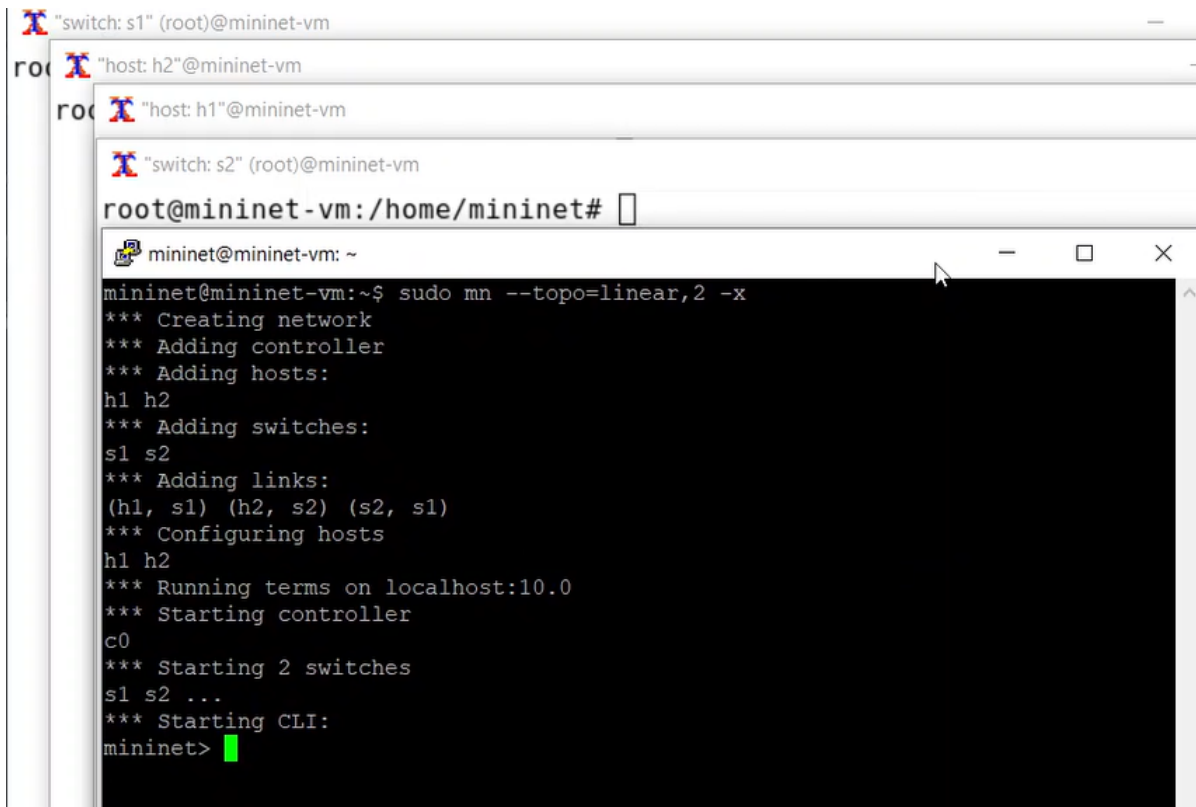
В виртуальной машине mininet исправим права запуска X-соединения (рис. 2.1):



```
root@mininet-vm: ~
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  23bec6ec917619e3fefb8971b7eeb293
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth add mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  23bec6ec917619e3fefb8971b7eeb293
root@mininet-vm:~# xauth list
mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  23bec6ec917619e3fefb8971b7eeb293
root@mininet-vm:~#
```

Рис. 2.1: Исправление прав запуска X-соединения в виртуальной машине mininet

Зададим топологию сети, состоящую из двух хостов и двух коммутаторов с назначенной по умолчанию mininet сетью 10.0.0.0/8 (рис. 2.2):



```
root@mininet-vm:~# "switch: s1" (root)@mininet-vm
root@mininet-vm:~# "host: h2" (root)@mininet-vm
root@mininet-vm:~# "host: h1" (root)@mininet-vm
root@mininet-vm:~# "switch: s2" (root)@mininet-vm
root@mininet-vm:/home/mininet# mininet@mininet-vm: ~
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. 2.2: Создание топологии с двумя хостами и двумя коммутаторами

На хостах h1, h2 и на коммутаторах s1, s2 введём команду `ifconfig`, чтобы отобразить информацию, относящуюся к их сетевым интерфейсам и назначенным им IP-адресам. В дальнейшем при работе с NETEM и командой `tc` будем использовать интерфейсы h1-eth0, h2-eth0, s1-eth2 (рис. 2.3):

```
"switch: s1" (root)@mininet-vm
root@mininet-vm:/home/mininet# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    ether 00:00:00:00:00:00 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

"switch: s2" (root)@mininet-vm
s1-eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    ether be:2c:c3:f3:96:e2 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

"host: h1" (root)@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 8a:5e:58:3f:15:e7 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

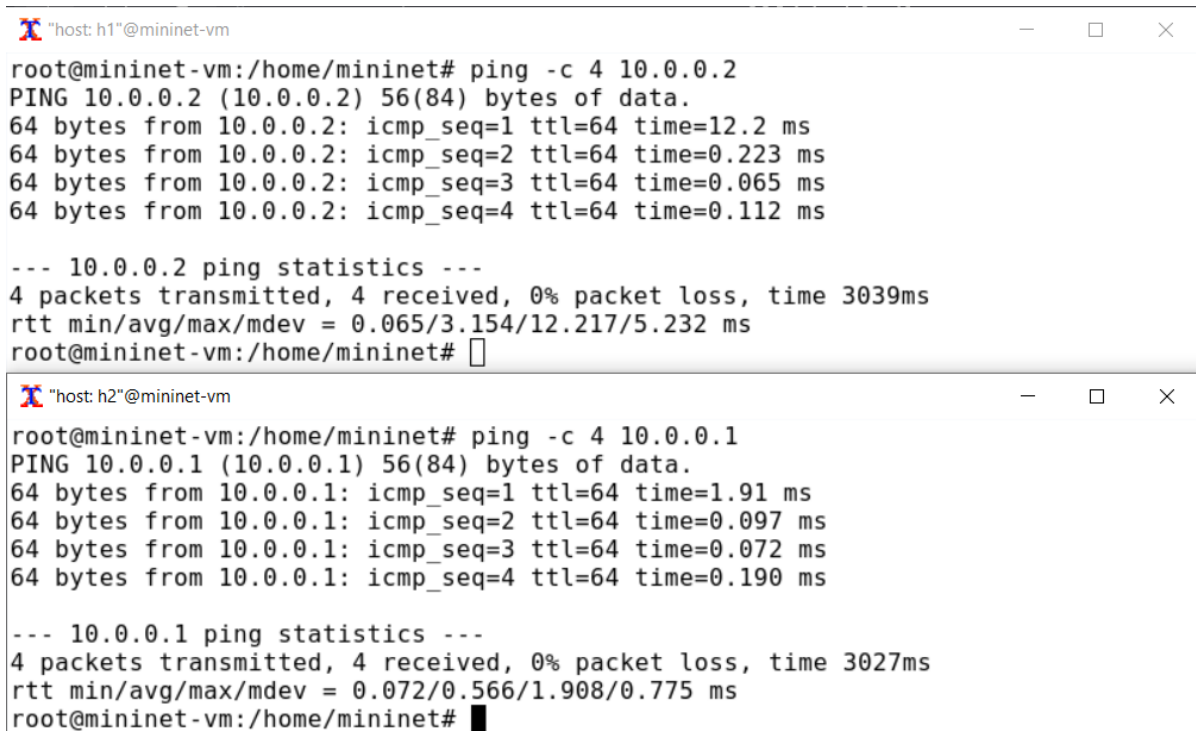
"host: h2" (root)@mininet-vm
root@mininet-vm:/home/mininet# ifconfig
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.2 netmask 255.0.0.0 broadcast 10.255.255.255
    ether 06:67:fa:ca:45:7e txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 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 707 bytes 246096 (246.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 707 bytes 246096 (246.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

Рис. 2.3: Отображение информации сетевых интерфейсов и IP-адресов

Проверим подключение между хостами h1 и h2 с помощью команды ping с параметром -c 4 (рис. 2.4):



The image shows two terminal windows from the mininet environment. The top window, titled "host: h1"@mininet-vm, shows the execution of a ping command from host h1 to host h2 (10.0.0.2). It displays four successful ping responses with varying times and a summary statistics block indicating 0% packet loss. The bottom window, titled "host: h2"@mininet-vm, shows the execution of a ping command from host h2 to host h1 (10.0.0.1). It also displays four successful ping responses and a summary statistics block indicating 0% packet loss.

```
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=12.2 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.223 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.065 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.112 ms

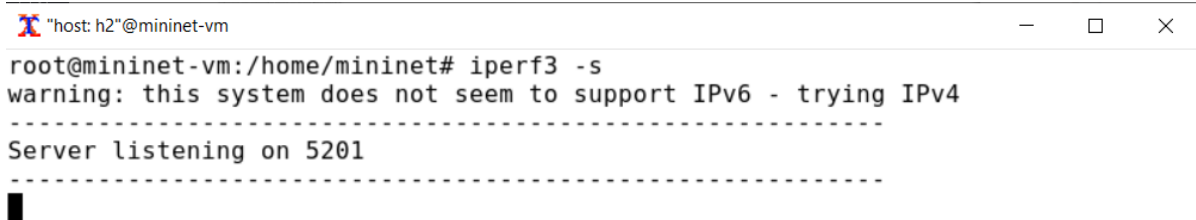
--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3039ms
rtt min/avg/max/mdev = 0.065/3.154/12.217/5.232 ms
root@mininet-vm:/home/mininet#
```

```
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=1.91 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.097 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.072 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.190 ms

--- 10.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3027ms
rtt min/avg/max/mdev = 0.072/0.566/1.908/0.775 ms
root@mininet-vm:/home/mininet#
```

Рис. 2.4: Проверка подключения между хостами h1 и h2

В терминале хоста h2 запустим iPerf3 в режиме сервера (рис. 2.5):



The image shows a terminal window titled "host: h2"@mininet-vm. The user has entered the command 'iperf3 -s' to start the iPerf3 server. The output shows a warning about IPv6 support and the server listening on port 5201.

```
root@mininet-vm:/home/mininet# iperf3 -s
warning: this system does not seem to support IPv6 - trying IPv4
-----
Server listening on 5201
-----
```

Рис. 2.5: Запуск iPerf3 в режиме сервера на хосте h2

В терминале хоста h1 запустим iPerf3 в режиме клиента (рис. 2.6):



```
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 35600 connected to 10.0.0.2 port 5201
[ ID] Interval          Transfer      Bitrate      Retr  Cwnd
[ 7]  0.00-1.00    sec   1.78 GBytes  15.2 Gbits/sec    0   8.14 MBytes
[ 7]  1.00-2.00    sec    859 MBytes  7.23 Gbits/sec    0   8.14 MBytes
[ 7]  2.00-3.00    sec   1.25 GBytes  10.8 Gbits/sec    0   8.14 MBytes
[ 7]  3.00-4.00    sec   1.01 GBytes  8.68 Gbits/sec    0   8.14 MBytes
[ 7]  4.00-5.00    sec   1.24 GBytes  10.7 Gbits/sec    0   8.14 MBytes
[ 7]  5.00-6.00    sec   1.01 GBytes  8.70 Gbits/sec    0   8.14 MBytes
[ 7]  6.00-7.00    sec   1.63 GBytes  14.0 Gbits/sec    0   8.14 MBytes
[ 7]  7.00-8.00    sec   1.59 GBytes  13.6 Gbits/sec    0   8.14 MBytes
[ 7]  8.00-9.00    sec   1.64 GBytes  14.1 Gbits/sec    0   8.14 MBytes
[ 7]  9.00-10.00   sec   1.59 GBytes  13.6 Gbits/sec    0   8.14 MBytes
- - - - -
[ ID] Interval          Transfer      Bitrate      Retr
[ 7]  0.00-10.00   sec   13.6 GBytes  11.7 Gbits/sec    0
[ 7]  0.00-10.00   sec   13.6 GBytes  11.6 Gbits/sec
sender
receiver

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

Рис. 2.6: Запуск iPerf3 в режиме клиента на хосте h1

После завершения работы iPerf3 на хосте h1 остановим iPerf3 на хосте h2, нажав Ctrl + c (рис. 2.7):

```
"host: h2"@mininet-vm
-----
Server listening on 5201
-----
Accepted connection from 10.0.0.1, port 35602
[ 7] local 10.0.0.2 port 5201 connected to 10.0.0.1 port 35604
[ ID] Interval          Transfer      Bitrate
[ 7]  0.00-1.00    sec  1.83 GBytes  15.7 Gbits/sec
[ 7]  1.00-2.00    sec  1.75 GBytes  15.0 Gbits/sec
[ 7]  2.00-3.00    sec  1.84 GBytes  15.8 Gbits/sec
[ 7]  3.00-4.00    sec  1.71 GBytes  14.7 Gbits/sec
[ 7]  4.00-5.00    sec  1.58 GBytes  13.6 Gbits/sec
[ 7]  5.00-6.00    sec  1.88 GBytes  16.1 Gbits/sec
[ 7]  6.00-7.00    sec  1.88 GBytes  16.2 Gbits/sec
[ 7]  7.00-8.00    sec  1.86 GBytes  16.0 Gbits/sec
[ 7]  8.00-9.00    sec  1.89 GBytes  16.2 Gbits/sec
[ 7]  9.00-10.00   sec  1.90 GBytes  16.3 Gbits/sec
-----
[ ID] Interval          Transfer      Bitrate
[ 7]  0.00-10.00   sec  18.1 GBytes  15.6 Gbits/sec
-----
Server listening on 5201
-----
^Ciperf3: interrupt - the server has terminated
root@mininet-vm:/home/mininet#
```

Рис. 2.7: Остановка iPerf3

Команду `tc` можно применить к сетевому интерфейсу устройства для формирования исходящего трафика. Требуется ограничить скорость отправки данных с конечного хоста с помощью фильтра Token Bucket Filter (tbfb).

Изменим пропускную способность хоста `h1`, установив пропускную способность на 10 Гбит/с на интерфейсе `h1-eth0` и параметры TBF-фильтра (рис. 2.8):

```
"host: h1"@mininet-vm
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root tbfb rate 10gb
it burst 5000000 limit 15000000
root@mininet-vm:/home/mininet#
```

Рис. 2.8: Изменение пропускной способности хоста `h1`

Фильтр `tbfb` требует установки значения всплеска при ограничении скорости. Это значение должно быть достаточно высоким, чтобы обеспечить установленную скорость. Она должна быть не ниже указанной частоты, делённой на `NZ`, где `NZ` — тактовая частота, настроенная как параметр ядра, и может быть извлечена

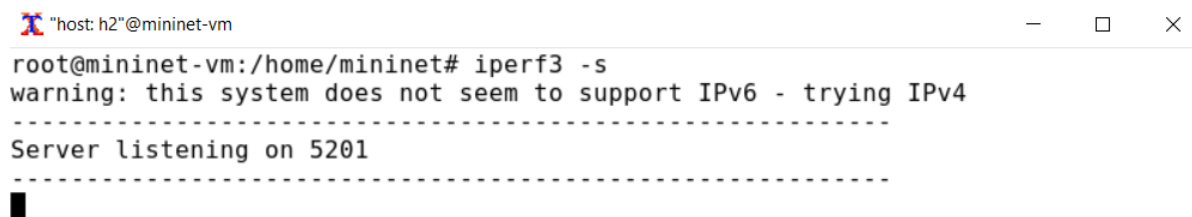
с помощью следующей команды (рис. 2.9):

```
root@mininet-vm:/home/mininet# egrep '^CONFIG_HZ_[0-9]+' /boot/config-`uname -r`  
CONFIG_HZ_250=y  
root@mininet-vm:/home/mininet# █
```

Рис. 2.9: Установка значения всплеска при ограничении скорости для фильтра tbf

С помощью iPerf3 проверим, что значение пропускной способности изменилось.

В терминале хоста h2 запустим iPerf3 в режиме сервера (рис. 2.10):



```
root@mininet-vm:/home/mininet# iperf3 -s  
warning: this system does not seem to support IPv6 - trying IPv4  
-----  
Server listening on 5201  
-----  
█
```

Рис. 2.10: Запуск iPerf3 в режиме сервера на хосте h2

В терминале хоста h1 запустим iPerf3 в режиме клиента (рис. 2.11):

```
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 35608 connected to 10.0.0.2 port 5201
[ ID] Interval          Transfer      Bitrate      Retr  Cwnd
[ 7]  0.00-1.00    sec   1.13 GBytes  9.68 Gbits/sec    9   8.86 MBytes
[ 7]  1.00-2.00    sec   1.11 GBytes  9.56 Gbits/sec    0   8.86 MBytes
[ 7]  2.00-3.00    sec   1.11 GBytes  9.58 Gbits/sec    0   8.86 MBytes
[ 7]  3.00-4.00    sec   1.11 GBytes  9.56 Gbits/sec    0   8.86 MBytes
[ 7]  4.00-5.00    sec   1.11 GBytes  9.56 Gbits/sec    0   8.86 MBytes
[ 7]  5.00-6.00    sec   1.11 GBytes  9.57 Gbits/sec    0  12.5 MBytes
[ 7]  6.00-7.00    sec  1005 MBytes  8.43 Gbits/sec    0  12.5 MBytes
[ 7]  7.00-8.00    sec   672 MBytes  5.64 Gbits/sec    0  12.5 MBytes
[ 7]  8.00-9.00    sec   1.11 GBytes  9.56 Gbits/sec    0  12.5 MBytes
[ 7]  9.00-10.00   sec   1.11 GBytes  9.56 Gbits/sec    0  12.5 MBytes
- - - - -
[ ID] Interval          Transfer      Bitrate      Retr
[ 7]  0.00-10.00   sec  10.6 GBytes  9.07 Gbits/sec    9
[ 7]  0.00-10.00   sec  10.5 GBytes  9.06 Gbits/sec    0
sender
receiver

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

Рис. 2.11: Запуск iPerf3 в режиме клиента на хосте h1

После завершения работы iPerf3 на хосте h1 остановим iPerf3 на хосте h2, нажав Ctrl + c (рис. 2.12):

```
"host: h2"@mininet-vm
Server listening on 5201
-----
Accepted connection from 10.0.0.1, port 35606
[ 7] local 10.0.0.2 port 5201 connected to 10.0.0.1 port 35608
[ ID] Interval          Transfer      Bitrate
[ 7]  0.00-1.00    sec  1.12 GBytes  9.60 Gbits/sec
[ 7]  1.00-2.00    sec  1.11 GBytes  9.56 Gbits/sec
[ 7]  2.00-3.00    sec  1.11 GBytes  9.56 Gbits/sec
[ 7]  3.00-4.00    sec  1.11 GBytes  9.57 Gbits/sec
[ 7]  4.00-5.00    sec  1.11 GBytes  9.57 Gbits/sec
[ 7]  5.00-6.00    sec  1.11 GBytes  9.56 Gbits/sec
[ 7]  6.00-7.00    sec   990 MBytes  8.29 Gbits/sec
[ 7]  7.00-8.00    sec   688 MBytes  5.78 Gbits/sec
[ 7]  8.00-9.00    sec  1.11 GBytes  9.56 Gbits/sec
[ 7]  9.00-10.00   sec  1.11 GBytes  9.56 Gbits/sec
[ 7] 10.00-10.00   sec   128 KBytes   233 Mbits/sec
-----
[ ID] Interval          Transfer      Bitrate
[ 7]  0.00-10.00   sec  10.5 GBytes  9.06 Gbits/sec
-----
Server listening on 5201
-----
^Ciperf3: interrupt - the server has terminated
root@mininet-vm:/home/mininet#
```

Рис. 2.12: Остановка iPerf3

Удалим модифицированную конфигурацию на хосте h1 (рис. 2.13):

```
"host: h1"@mininet-vm
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root
root@mininet-vm:/home/mininet#
```

Рис. 2.13: Удаление модифицированной конфигурации на хосте h1

Применим правило ограничения скорости tbf с параметрами rate = 10gbit, burst = 5,000,000, limit= 15,000,000 к интерфейсу s1-eth2 коммутатора s1, который соединяет его с коммутатором s2 (рис. 2.14):

```
"switch: s1" (root)@mininet-vm
root@mininet-vm:/home/mininet# sudo tc qdisc add dev s1-eth2 root tbf rate 10gb
it burst 5000000 limit 15000000
root@mininet-vm:/home/mininet#
```

Рис. 2.14: Применение правила ограничения скорости tbf

Проверим конфигурацию с помощью инструмента iperf3 для измерения пропускной способности.

В терминале хоста h2 запустим iPerf3 в режиме сервера (рис. 2.15):

```
"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
-----
```

Рис. 2.15: Запуск iPerf3 в режиме сервера на хосте h2

В терминале хоста h1 запустим iPerf3 в режиме клиента (рис. 2.16):

```
"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 35612 connected to 10.0.0.2 port 5201
[ ID] Interval          Transfer      Bitrate      Retr  Cwnd
[ 7]  0.00-1.00    sec   1.13 GBytes  9.68 Gbits/sec    0   3.23 MBytes
[ 7]  1.00-2.00    sec   1.11 GBytes  9.56 Gbits/sec    0   3.74 MBytes
[ 7]  2.00-3.00    sec   1.11 GBytes  9.57 Gbits/sec    0   3.74 MBytes
[ 7]  3.00-4.00    sec   1.11 GBytes  9.56 Gbits/sec    0   3.74 MBytes
[ 7]  4.00-5.00    sec   1.11 GBytes  9.57 Gbits/sec    0   3.94 MBytes
[ 7]  5.00-6.00    sec   1.11 GBytes  9.56 Gbits/sec    0   3.94 MBytes
[ 7]  6.00-7.00    sec   1.11 GBytes  9.56 Gbits/sec    0   4.14 MBytes
[ 7]  7.00-8.00    sec   1.08 GBytes  9.28 Gbits/sec    0   6.22 MBytes
[ 7]  8.00-9.00    sec    869 MBytes  7.29 Gbits/sec    0   6.67 MBytes
[ 7]  9.00-10.00   sec   1.11 GBytes  9.56 Gbits/sec    0   6.67 MBytes
-----
[ ID] Interval          Transfer      Bitrate      Retr
[ 7]  0.00-10.00   sec   10.9 GBytes  9.32 Gbits/sec    0
[ 7]  0.00-10.01   sec   10.8 GBytes  9.31 Gbits/sec    0
sender
receiver

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

Рис. 2.16: Запуск iPerf3 в режиме клиента на хосте h1

После завершения работы iPerf3 на хосте h1 остановим iPerf3 на хосте h2, нажав Ctrl + c (рис. 2.17):

```
"host: h2"@mininet-vm
Server listening on 5201
-----
Accepted connection from 10.0.0.1, port 35610
[ 7] local 10.0.0.2 port 5201 connected to 10.0.0.1 port 35612
[ ID] Interval          Transfer    Bitrate
[ 7]  0.00-1.00      sec  1.12 GBytes  9.58 Gbits/sec
[ 7]  1.00-2.00      sec  1.11 GBytes  9.58 Gbits/sec
[ 7]  2.00-3.00      sec  1.11 GBytes  9.56 Gbits/sec
[ 7]  3.00-4.00      sec  1.11 GBytes  9.57 Gbits/sec
[ 7]  4.00-5.00      sec  1.11 GBytes  9.56 Gbits/sec
[ 7]  5.00-6.00      sec  1.11 GBytes  9.57 Gbits/sec
[ 7]  6.00-7.00      sec  1.11 GBytes  9.56 Gbits/sec
[ 7]  7.00-8.00      sec  1.08 GBytes  9.27 Gbits/sec
[ 7]  8.00-9.00      sec   869 MBytes  7.29 Gbits/sec
[ 7]  9.00-10.00     sec  1.11 GBytes  9.57 Gbits/sec
[ 7] 10.00-10.01     sec   256 KBytes   319 Mbits/sec
-----
[ ID] Interval          Transfer    Bitrate
[ 7]  0.00-10.01     sec  10.8 GBytes  9.31 Gbits/sec
receiver
-----
Server listening on 5201
-----
^Ciperf3: interrupt - the server has terminated
root@mininet-vm:/home/mininet#
```

Рис. 2.17: Остановка iPerf3

Удалим модифицированную конфигурацию на коммутаторе s1 (рис. 2.18):

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

Рис. 2.18: Удаление модифицированной конфигурации на коммутаторе s1

NETEM используется для изменения задержки, джиттера, повреждения пакетов и т.д. TBF может использоваться для ограничения скорости. Утилита tc позволяет комбинировать несколько модулей. При этом первая дисциплина очереди (qdisc1) присоединяется к корневой метке, последующие дисциплины очереди можно прикрепить к своим родителям, указав правильную метку.

Объединим NETEM и TBF, введя на интерфейсе s1-eth2 коммутатора s1 задержку, джиттер, повреждение пакетов и указав скорость (рис. 2.19):



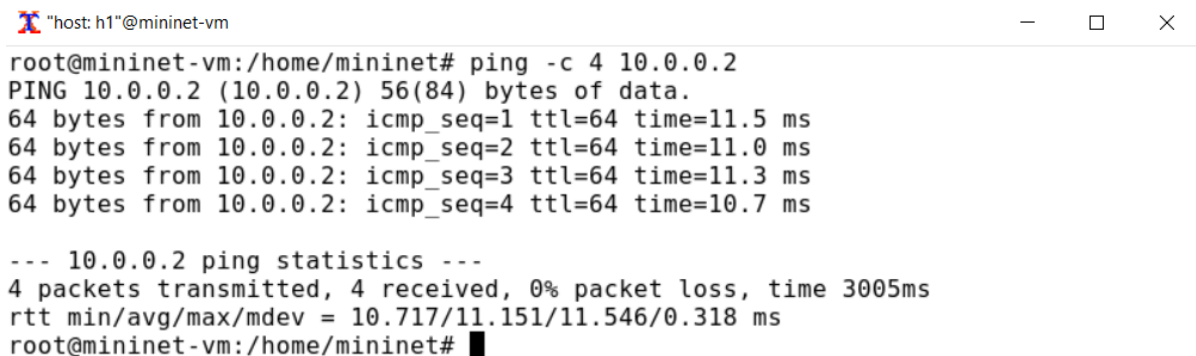
```

"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#

```

Рис. 2.19: Объединение NETEM и TBF

Убедимся, что соединение от хоста h1 к хосту h2 имеет заданную задержку. Для этого запустим команду ping с параметром -c 4 с терминала хоста h1 (рис. 2.20):



```

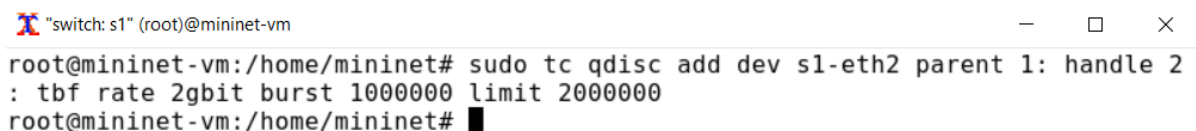
"host: h1" (root)@mininet-vm
root@mininet-vm:/home/mininet# ping -c 4 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=11.5 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=11.0 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=11.3 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=10.7 ms

--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 10.717/11.151/11.546/0.318 ms
root@mininet-vm:/home/mininet#

```

Рис. 2.20: Проверка задержки

Добавим второе правило на коммутаторе s1, которое задаёт ограничение скорости с помощью tbf с параметрами rate=2gbit, burst=1,000,000, limit=2,000,000 (рис. 2.21):



```

"switch: s1" (root)@mininet-vm
root@mininet-vm:/home/mininet# sudo tc qdisc add dev s1-eth2 parent 1: handle 2: tbf rate 2gbit burst 1000000 limit 2000000
root@mininet-vm:/home/mininet#

```

Рис. 2.21: Добавление второго правила на коммутаторе s1

Проверим конфигурацию с помощью инструмента iperf3 для измерения пропускной способности.

В терминале хоста h2 запустим iPerf3 в режиме сервера (рис. 2.22):



```
"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
-----
```

Рис. 2.22: Запуск iPerf3 в режиме сервера на хосте h2

В терминале хоста h1 запустим iPerf3 в режиме клиента (рис. 2.23):

```
"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 35616 connected to 10.0.0.2 port 5201
[ ID] Interval          Transfer      Bitrate      Retr  Cwnd
[ 7]  0.00-1.00    sec    214 MBytes  1.79 Gbits/sec    855  2.71 MBytes
[ 7]  1.00-2.00    sec    229 MBytes  1.92 Gbits/sec     0  2.86 MBytes
[ 7]  2.00-3.00    sec    228 MBytes  1.91 Gbits/sec     0  2.97 MBytes
[ 7]  3.00-4.00    sec    228 MBytes  1.91 Gbits/sec     0  3.06 MBytes
[ 7]  4.00-5.00    sec    218 MBytes  1.82 Gbits/sec    450  2.20 MBytes
[ 7]  5.00-6.00    sec    214 MBytes  1.79 Gbits/sec     0  2.33 MBytes
[ 7]  6.00-7.00    sec    225 MBytes  1.89 Gbits/sec     0  2.43 MBytes
[ 7]  7.00-8.00    sec    228 MBytes  1.91 Gbits/sec     0  2.51 MBytes
[ 7]  8.00-9.00    sec    229 MBytes  1.92 Gbits/sec     0  2.56 MBytes
[ 7]  9.00-10.00   sec    228 MBytes  1.91 Gbits/sec     0  2.60 MBytes
-----
[ ID] Interval          Transfer      Bitrate      Retr
[ 7]  0.00-10.00   sec    2.19 GBytes  1.88 Gbits/sec   1305
[ 7]  0.00-10.01   sec    2.18 GBytes  1.87 Gbits/sec
sender
receiver

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

Рис. 2.23: Запуск iPerf3 в режиме клиента на хосте h1

После завершения работы iPerf3 на хосте h1 остановим iPerf3 на хосте h2, нажав Ctrl + c (рис. 2.24):

```
"host: h2"@mininet-vm
Server listening on 5201
-----
Accepted connection from 10.0.0.1, port 35614
[ 7] local 10.0.0.2 port 5201 connected to 10.0.0.1 port 35616
[ ID] Interval          Transfer      Bitrate
[ 7] 0.00-1.00 sec      204 MBytes   1.71 Gbits/sec
[ 7] 1.00-2.00 sec      228 MBytes   1.91 Gbits/sec
[ 7] 2.00-3.00 sec      228 MBytes   1.91 Gbits/sec
[ 7] 3.00-4.00 sec      228 MBytes   1.91 Gbits/sec
[ 7] 4.00-5.00 sec      216 MBytes   1.81 Gbits/sec
[ 7] 5.00-6.00 sec      215 MBytes   1.80 Gbits/sec
[ 7] 6.00-7.00 sec      225 MBytes   1.89 Gbits/sec
[ 7] 7.00-8.00 sec      228 MBytes   1.91 Gbits/sec
[ 7] 8.00-9.00 sec      228 MBytes   1.91 Gbits/sec
[ 7] 9.00-10.00 sec     228 MBytes   1.91 Gbits/sec
[ 7] 10.00-10.01 sec    2.42 MBytes   1.83 Gbits/sec
-----
[ ID] Interval          Transfer      Bitrate
[ 7] 0.00-10.01 sec    2.18 GBytes   1.87 Gbits/sec
-----
Server listening on 5201
-----
^Ciperf3: interrupt - the server has terminated
root@mininet-vm:/home/mininet#
```

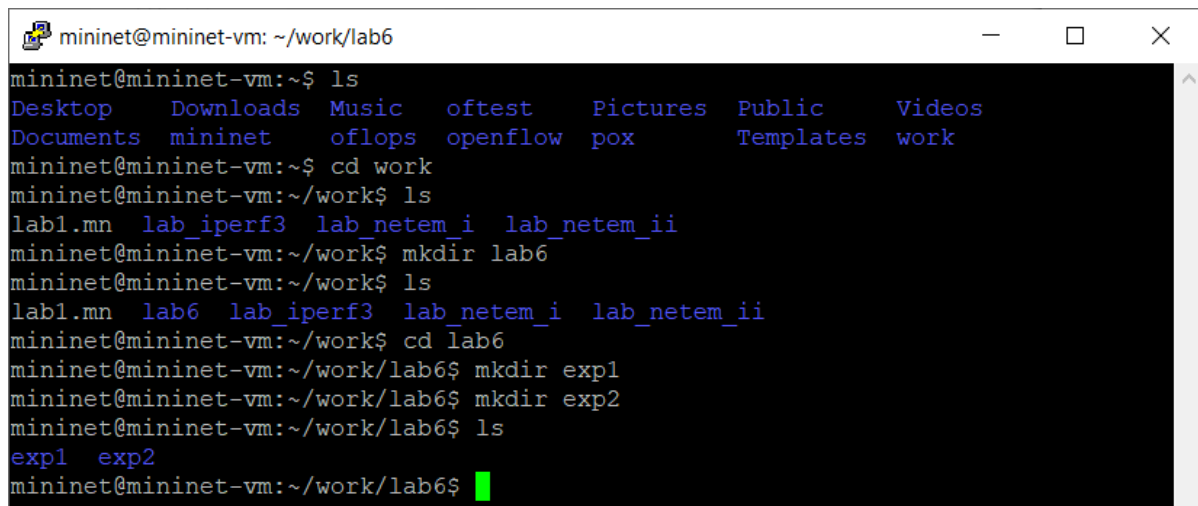
Рис. 2.24: Остановка iPerf3

Удалим модифицированную конфигурацию на коммутаторе s1 (рис. 2.25):

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

Рис. 2.25: Удаление модифицированной конфигурации на коммутаторе s1

Для самостоятельного задания создадим необходимые каталоги (рис. 2.26):

A terminal window titled 'mininet@mininet-vm: ~/work/lab6' with standard window controls. The terminal shows a sequence of commands to create a directory structure. It starts with 'ls' in the home directory, then 'cd work', then 'ls' in the work directory. Then 'mkdir lab6' is executed, followed by 'ls' to confirm its creation. Then 'cd lab6' is used to move into the new directory, and finally 'mkdir exp1' and 'mkdir exp2' are used to create subdirectories. The final 'ls' command shows 'exp1' and 'exp2' as the contents of the current directory. The prompt is always 'mininet@mininet-vm:~/work/lab6\$' and the cursor is a green block.

```
mininet@mininet-vm: ~$ ls
Desktop  Downloads  Music      oftest     Pictures   Public     Videos
Documents mininet    oflops     openflow   pox        Templates  work
mininet@mininet-vm: ~$ cd work
mininet@mininet-vm: ~/work$ ls
lab1.mn  lab_iperf3  lab_netem_i  lab_netem_ii
mininet@mininet-vm: ~/work$ mkdir lab6
mininet@mininet-vm: ~/work$ ls
lab1.mn  lab6  lab_iperf3  lab_netem_i  lab_netem_ii
mininet@mininet-vm: ~/work$ cd lab6
mininet@mininet-vm: ~/work/lab6$ mkdir exp1
mininet@mininet-vm: ~/work/lab6$ mkdir exp2
mininet@mininet-vm: ~/work/lab6$ ls
exp1  exp2
mininet@mininet-vm: ~/work/lab6$
```

Рис. 2.26: Создание необходимых каталогов

Затем напишем скрипты по примеру из прошлых лабораторных работ (рис. 2.27 - рис. 2.29):

```
mininet@mininet-vm: ~/work/lab6/exp1
/home/mininet/exp1.py [----] 50 L:[ 17+ 0 17/ 59] *(318 /1560b) 10 0x00A [*][X]
"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' )

s1.cmd('ip link del s1-eth2')
s2.cmd('ip link del s2-eth1')

info( '*** Creating links\n' )
net.addLink( h1, s1 )
net.addLink( h2, s1 )
net.addLink( s1, s2 )

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

info( '*** Set delay\n' )
s1.cmdPrint( 'tc qdisc add dev s1-eth2 root handle 1: netem delay 10ms' )
s2.cmdPrint( 'tc qdisc add dev s1-eth2 parent 1: handle 2: tbf rate 2gbit bu

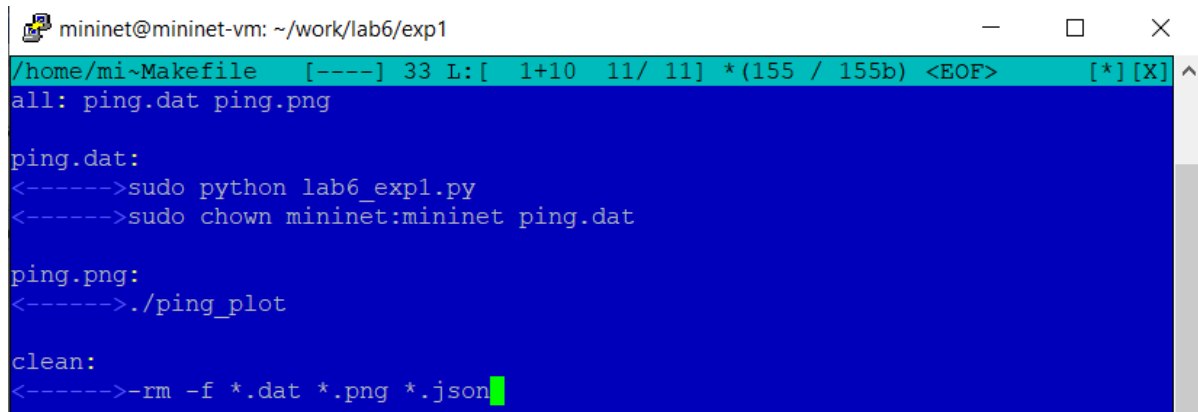
info( '*** Traffic generation\n' )
h2.cmdPrint('iperf3 -s -D -1')
time.sleep(10) # Wait 10 seconds
h1.cmdPrint( 'iperf3 -c', h2.IP(), '-J > iperf_result.json' )
h1.cmdPrint('ping -c 100', h2.IP(), '| grep "time=" | awk \'{print $5, $7}\''

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

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

1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn10Quit
```

Рис. 2.27: Написание 1 скрипта



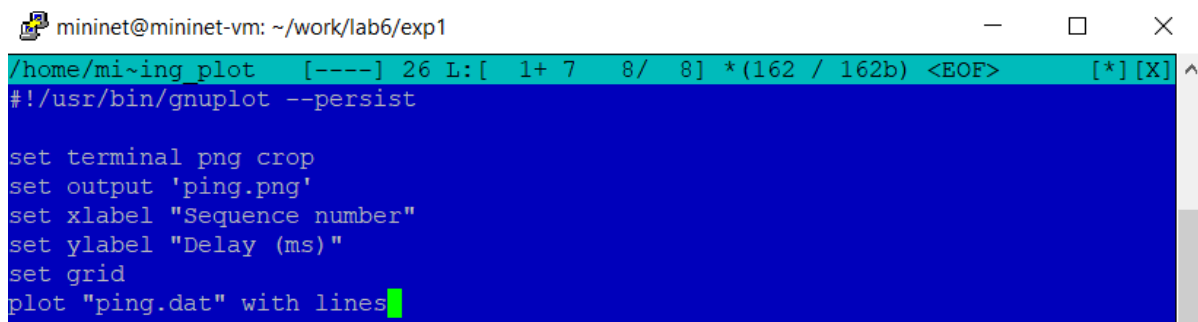
```
mininet@mininet-vm: ~/work/lab6/exp1
/home/mininet~Makefile  [-----] 33 L:[ 1+10 11/ 11] *(155 / 155b) <EOF> [*][X] ^
all: ping.dat ping.png

ping.dat:
<----->sudo python lab6_exp1.py
<----->sudo chown mininet:mininet ping.dat

ping.png:
<----->./ping_plot

clean:
<----->-rm -f *.dat *.png *.json
```

Рис. 2.28: Написание 2 скрипта



```
mininet@mininet-vm: ~/work/lab6/exp1
/home/mininet~ing_plot  [-----] 26 L:[ 1+ 7 8/ 8] *(162 / 162b) <EOF> [*][X] ^
#!/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. 2.29: Написание 3 скрипта

Запустим на выполнение скрипты для первой части самостоятельного задания (рис. 2.30):

```

mininet@mininet-vm: ~/work/lab6/exp1
mininet@mininet-vm:~/work/lab6/exp1$ make
sudo python lab6_exp1.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 delay
*** s1 : ('tc qdisc add dev s1-eth2 root handle 1: netem delay 10ms',)
*** s2 : ('tc qdisc add dev s1-eth2 parent 1: handle 2: tbfb rate 2gbit burst 1000000 limit 2000000',)
*** Traffic generation
*** h2 : ('iperf3 -s -D -1',)
*** h1 : ('iperf3 -c', '10.0.0.2', '-J > iperf_result.json')
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "time=" | awk \'{print $5, $7}\'} | sed -e \'s/time=//g\' -e \'s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 3 links
...
*** Stopping 2 switches
s1 s2
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab6/exp1$ ls
iperf_result.json lab6_exp1.py Makefile ping.dat ping_plot ping.png
mininet@mininet-vm:~/work/lab6/exp1$

```

Рис. 2.30: Выполнение скриптов

Изменим параметры в скрипте для первого задания и запустим на выполнение (рис. 2.31 - рис. 2.32):

```

mininet@mininet-vm: ~/work/lab6/exp1
/home/mininet/work/lab6/exp1/lab6_exp1.py [-M--] 92 L: [ 12+32 44/ 58] *(1029/1469b) 39 0x027
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' )
    s2 = net.addSwitch( 's2' )

    s1.cmd('ip link del s1-eth2')
    s2.cmd('ip link del s2-eth1')

    info( '*** Creating links\n' )
    net.addLink( h1, s1 )
    net.addLink( h2, s1 )
    net.addLink( s1, s2 )

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

    info( '*** Set delay\n' )
    s1.cmdPrint( 'tc qdisc add dev h1-eth0 root tbf rate 10gbit burst 5000000 limit 1300000 0 )'

    info( '*** Traffic generation\n' )
    h2.cmdPrint('iperf3 -s -B -i')
    time.sleep(10) # Wait 10 seconds
    h1.cmdPrint( 'iperf3 -c', h2.IP(), '-J > iperf_result.json' )
    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. 2.31: Изменение параметров

```

mininet@mininet-vm: ~/work/lab6/exp1
mininet@mininet-vm:~/work/lab6/exp1$ make
sudo python lab6_exp1.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 delay
*** s1 : ('tc qdisc add dev h1-eth0 root tbf rate 10gbit burst 5000000 limit 15000000',)
Cannot find device "h1-eth0"
*** Traffic generation
*** h2 : ('iperf3 -s -D -l',)
*** h1 : ('iperf3 -c', '10.0.0.2', '-J > iperf_result.json')
*** h1 : ('ping -c 100', '10.0.0.2', '| grep "time=" | awk \'{print $5, $7}\'} | sed -e \'s/time=//g\' -e \'s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 3 links
...
*** Stopping 2 switches
s1 s2
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab6/exp1$ ls
iperf_result.json lab6_exp1.py Makefile ping.dat ping_plot ping.png
mininet@mininet-vm:~/work/lab6/exp1$

```

Рис. 2.32: Выполнение скриптов

Посмотрим полученные графики (рис. 2.33 - рис. 2.34):



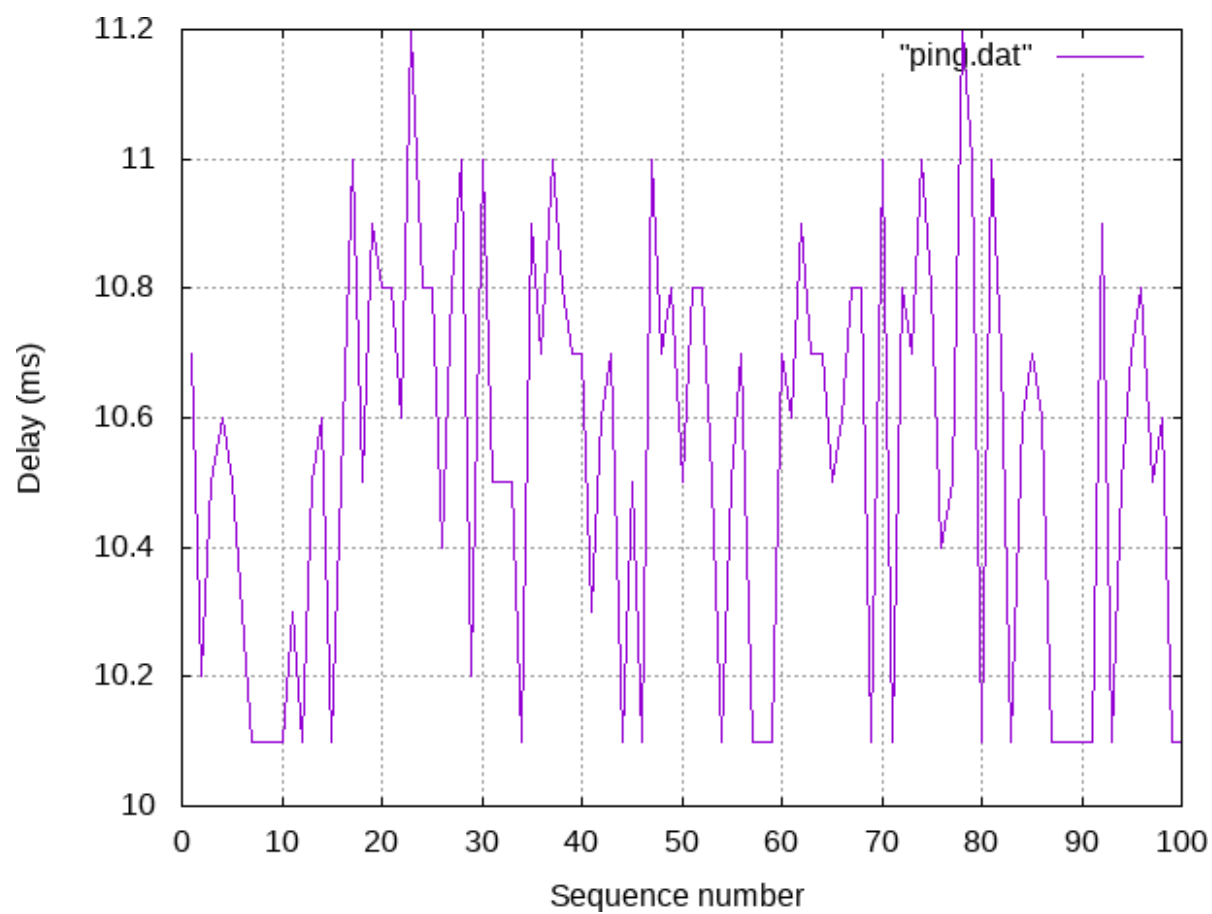


Рис. 2.33: График №1

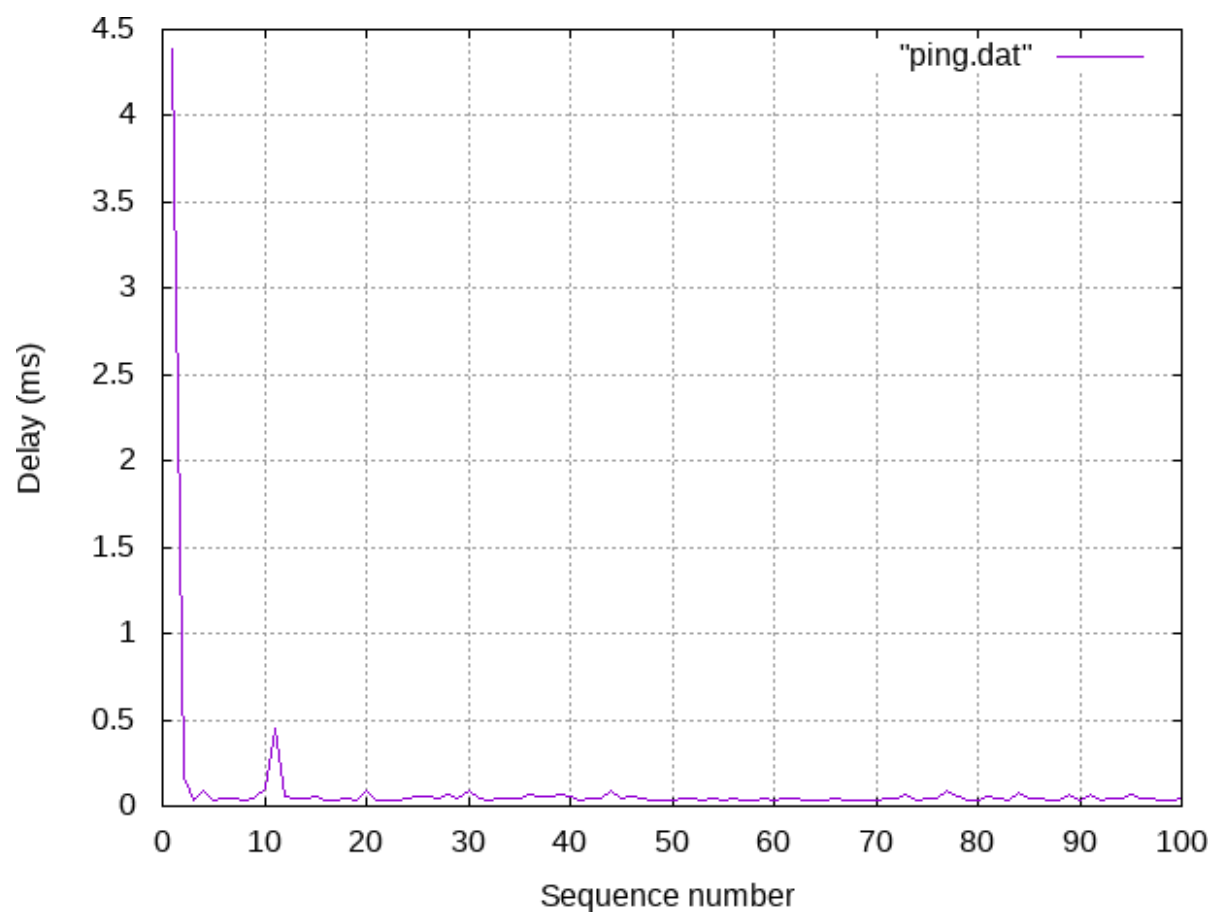


Рис. 2.34: График №2

## 3 Вывод

В ходе выполнения лабораторной работы познакомились с принципами работы дисциплины очереди Token Bucket Filter, которая формирует входящий/исходящий трафик для ограничения пропускной способности, а также получили навыки моделирования и исследования поведения трафика посредством проведения интерактивного и воспроизводимого экспериментов в Mininet.

## 4 Список литературы. Библиография

[1] Mininet: <https://mininet.org/>