

# **Отчёт по лабораторной работе №6**

**Дисциплина: Моделирование сетей передачи данных**

**Боровиков Даниил Александрович НПИбд-01-22**

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>4</b>	<b>Выводы</b>	<b>26</b>
	<b>Список литературы</b>	<b>27</b>

# Список иллюстраций

3.1	Исправление прав запуска X-соединения в виртуальной машине mininet . . . . .	7
3.2	Создание топологии с двумя хостами и двумя коммутаторами . .	8
3.3	Отображение информации сетевых интерфейсов и IP-адресов . .	9
3.4	Проверка подключения между хостами h1 и h2 . . . . .	9
3.5	Запуск iPerf3 в режиме сервера на хосте h2 . . . . .	10
3.6	Запуск iPerf3 в режиме клиента на хосте h1 . . . . .	10
3.7	Остановка iPerf3 . . . . .	11
3.8	Изменение пропускной способности хоста h1 . . . . .	11
3.9	Установка значения всплеска при ограничении скорости для филь- тра tbf . . . . .	12
3.10	Запуск iPerf3 в режиме сервера на хосте h2 . . . . .	12
3.11	Запуск iPerf3 в режиме клиента на хосте h1 . . . . .	12
3.12	Остановка iPerf3 . . . . .	13
3.13	Удаление модифицированной конфигурации на хосте h1 . . . . .	13
3.14	Применение правила ограничения скорости tbf . . . . .	13
3.15	Запуск iPerf3 в режиме сервера на хосте h2 . . . . .	14
3.16	Запуск iPerf3 в режиме клиента на хосте h1 . . . . .	14
3.17	Остановка iPerf3 . . . . .	15
3.18	Удаление модифицированной конфигурации на коммутаторе s1 .	15
3.19	Объединение NETEM и TBF . . . . .	16
3.20	Проверка задержки . . . . .	16
3.21	Добавление второго правила на коммутаторе s1 . . . . .	16
3.22	Запуск iPerf3 в режиме сервера на хосте h2 . . . . .	17
3.23	Запуск iPerf3 в режиме клиента на хосте h1 . . . . .	17
3.24	Остановка iPerf3 . . . . .	18
3.25	Удаление модифицированной конфигурации на коммутаторе s1 .	18
3.26	Создание необходимых каталогов . . . . .	19
3.27	Написание 1 скрипта . . . . .	20
3.28	Написание 2 скрипта . . . . .	21
3.29	Написание 3 скрипта . . . . .	21
3.30	Выполнение скриптов . . . . .	22
3.31	Изменение параметров . . . . .	23
3.32	Выполнение скриптов . . . . .	23
3.33	График №1 . . . . .	24
3.34	График №2 . . . . .	25

# Список таблиц

[1]

# 1 Цель работы

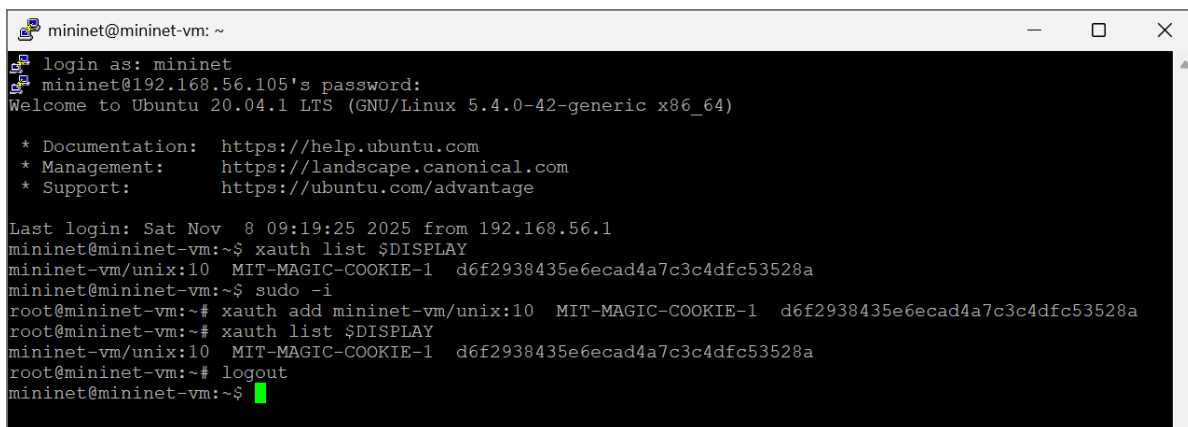
Основной целью работы является знакомство с принципами работы дисциплины очереди Token Bucket Filter, которая формирует входящий/исходящий трафик для ограничения пропускной способности, а также получение навыков моделирования и исследования поведения трафика посредством проведения интерактивного и воспроизводимого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

## 2 Задание

1. Задайте топологию, состоящую из двух хостов и двух коммутаторов с назначенной по умолчанию mininet сетью 10.0.0.0/8.
2. Проведите интерактивные эксперименты по ограничению пропускной способности сети с помощью TBF в эмулируемой глобальной сети.
3. Самостоятельно реализуйте воспроизводимые эксперимент по применению TBF для ограничения пропускной способности. Постройте соответствующие графики.

### 3 Выполнение лабораторной работы

В виртуальной машине mininet исправим права запуска X-соединения (рис. 3.1):



```
mininet@mininet-vm: ~  
login as: mininet  
mininet@192.168.56.105's password:  
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  
  
Last login: Sat Nov  8 09:19:25 2025 from 192.168.56.1  
mininet@mininet-vm:~$ xauth list $DISPLAY  
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 d6f2938435e6ecad4a7c3c4dfc53528a  
mininet@mininet-vm:~$ sudo -i  
root@mininet-vm:~# xauth add mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 d6f2938435e6ecad4a7c3c4dfc53528a  
root@mininet-vm:~# xauth list $DISPLAY  
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 d6f2938435e6ecad4a7c3c4dfc53528a  
root@mininet-vm:~# logout  
mininet@mininet-vm:~$
```

Рис. 3.1: 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. 3.2):

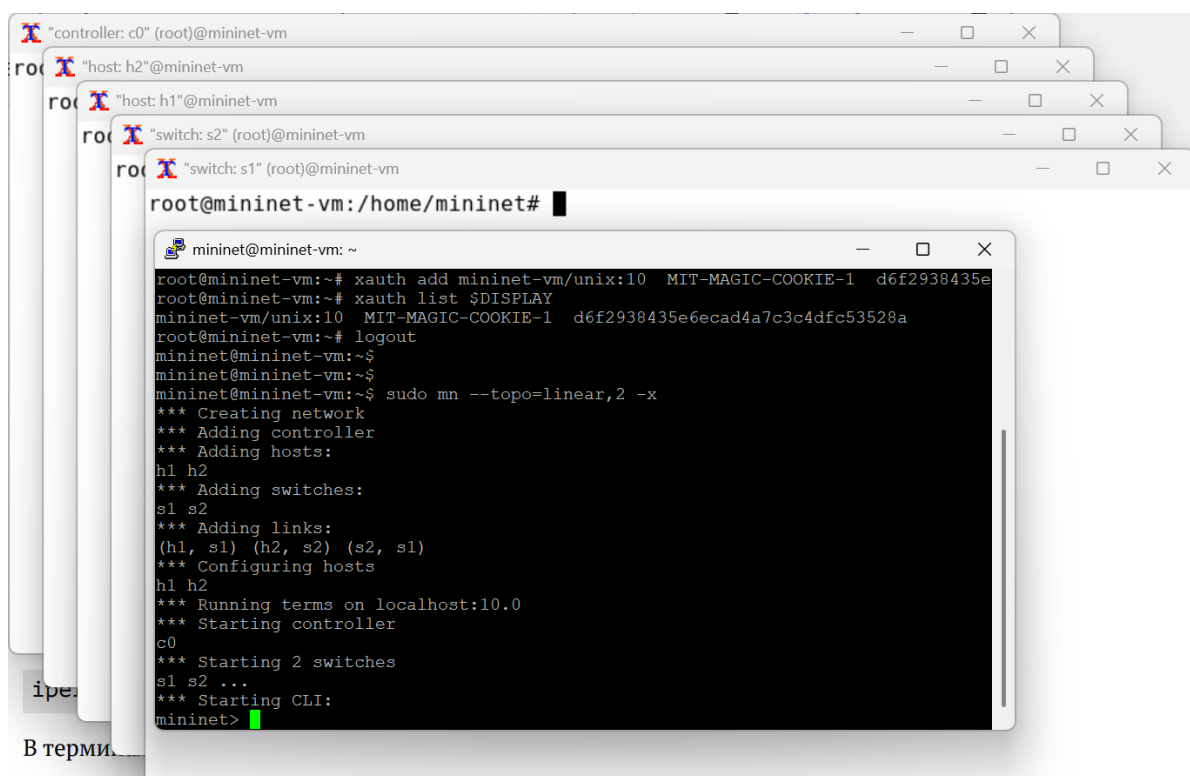


Рис. 3.2: Создание топологии с двумя хостами и двумя коммутаторами

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. 3.3):



```
root@mininet-vm:/home/mininet# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.105 netmask 255.255.255.0 broadcast 192.168.56.255
    ether 08:00:27:e8:34:b9 txqueuelen 1000 (Ethernet)
    RX packets 1494 bytes 517805 (517.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1319 bytes 882631 (882.6 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 08:00:27:d0:e5:b3 txqueuelen 1000 (Ethernet)
    RX packets 335 bytes 43935 (43.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 354 bytes 32550 (32.5 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 1228 bytes 283284 (283.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1228 bytes 283284 (283.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet#

root@mininet-vm:/home/mininet# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.105 netmask 255.255.255.0 broadcast 192.168.56.255
    ether 08:00:27:e8:34:b9 txqueuelen 1000 (Ethernet)
    RX packets 1696 bytes 553751 (553.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1526 bytes 924915 (924.9 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 08:00:27:d0:e5:b3 txqueuelen 1000 (Ethernet)
    RX packets 336 bytes 44025 (44.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 355 bytes 32640 (32.6 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 5092 bytes 1341806 (1.3 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 5092 bytes 1341806 (1.3 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet#

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 6a:aa:ec:fa:3b:6c 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 1228 bytes 283284 (283.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1228 bytes 283284 (283.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet#

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 62:77:e9:40:c8:a8 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 976 bytes 262452 (262.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 976 bytes 262452 (262.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

Рис. 3.3: Отображение информации сетевых интерфейсов и IP-адресов

Проверим подключение между хостами h1 и h2 с помощью команды ping с параметром -c 4 (рис. 3.4):

```
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=1.88 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.090 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.078 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.096 ms

--- 10.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3046ms
rtt min/avg/max/mdev = 0.078/0.535/1.878/0.775 ms
root@mininet-vm:/home/mininet#

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=6.52 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.309 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.245 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.076 ms

--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3059ms
rtt min/avg/max/mdev = 0.076/1.788/6.524/2.735 ms
root@mininet-vm:/home/mininet#
```

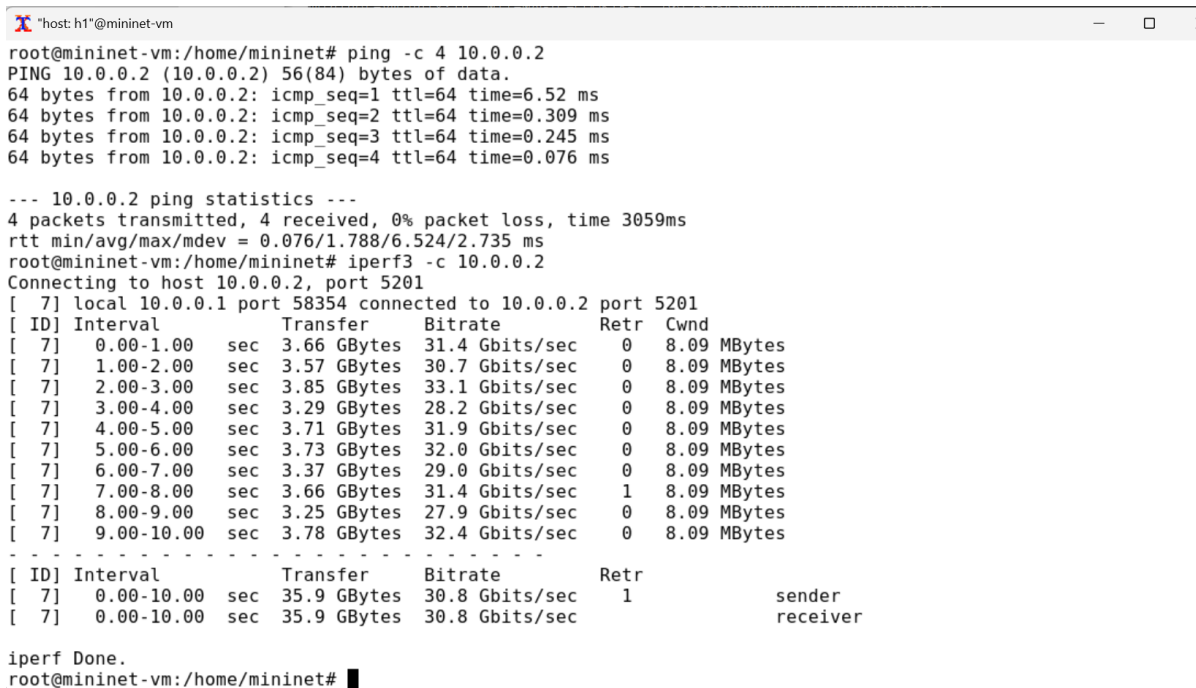
Рис. 3.4: Проверка подключения между хостами h1 и h2

В терминале хоста h2 запустим iPerf3 в режиме сервера (рис. 3.5):

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

Рис. 3.5: Запуск iPerf3 в режиме сервера на хосте h2

В терминале хоста h1 запустим iPerf3 в режиме клиента (рис. 3.6):



```
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=6.52 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.309 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.245 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.076 ms

--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3059ms
rtt min/avg/max/mdev = 0.076/1.788/6.524/2.735 ms
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 58354 connected to 10.0.0.2 port 5201
[ ID] Interval           Transfer     Bitrate      Retr  Cwnd
[ 7]  0.00-1.00   sec   3.66 GBytes  31.4 Gbits/sec    0   8.09 MBytes
[ 7]  1.00-2.00   sec   3.57 GBytes  30.7 Gbits/sec    0   8.09 MBytes
[ 7]  2.00-3.00   sec   3.85 GBytes  33.1 Gbits/sec    0   8.09 MBytes
[ 7]  3.00-4.00   sec   3.29 GBytes  28.2 Gbits/sec    0   8.09 MBytes
[ 7]  4.00-5.00   sec   3.71 GBytes  31.9 Gbits/sec    0   8.09 MBytes
[ 7]  5.00-6.00   sec   3.73 GBytes  32.0 Gbits/sec    0   8.09 MBytes
[ 7]  6.00-7.00   sec   3.37 GBytes  29.0 Gbits/sec    0   8.09 MBytes
[ 7]  7.00-8.00   sec   3.66 GBytes  31.4 Gbits/sec    1   8.09 MBytes
[ 7]  8.00-9.00   sec   3.25 GBytes  27.9 Gbits/sec    0   8.09 MBytes
[ 7]  9.00-10.00  sec   3.78 GBytes  32.4 Gbits/sec    0   8.09 MBytes
-----
[ ID] Interval           Transfer     Bitrate      Retr
[ 7]  0.00-10.00  sec   35.9 GBytes  30.8 Gbits/sec    1
[ 7]  0.00-10.00  sec   35.9 GBytes  30.8 Gbits/sec    0
                                     sender
                                     receiver

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

Рис. 3.6: Запуск iPerf3 в режиме клиента на хосте h1

После завершения работы iPerf3 на хосте h1 остановим iPerf3 на хосте h2, нажав Ctrl + c (рис. 3.7):

```
"host: h2"@mininet-vm
Server listening on 5201
-----
Accepted connection from 10.0.0.1, port 58352
[ 7] local 10.0.0.2 port 5201 connected to 10.0.0.1 port 58354
[ ID] Interval      Transfer    Bitrate
[ 7]  0.00-1.00    sec  3.66 GBytes 31.5 Gbits/sec
[ 7]  1.00-2.00    sec  3.56 GBytes 30.6 Gbits/sec
[ 7]  2.00-3.00    sec  3.85 GBytes 33.1 Gbits/sec
[ 7]  3.00-4.00    sec  3.29 GBytes 28.3 Gbits/sec
[ 7]  4.00-5.00    sec  3.71 GBytes 31.9 Gbits/sec
[ 7]  5.00-6.00    sec  3.73 GBytes 32.0 Gbits/sec
[ 7]  6.00-7.00    sec  3.37 GBytes 29.0 Gbits/sec
[ 7]  7.00-8.00    sec  3.66 GBytes 31.5 Gbits/sec
[ 7]  8.00-9.00    sec  3.25 GBytes 27.9 Gbits/sec
[ 7]  9.00-10.00   sec  3.78 GBytes 32.4 Gbits/sec
[ 7] 10.00-10.00   sec  4.00 MBytes 12.9 Gbits/sec
-----
[ ID] Interval      Transfer    Bitrate
[ 7]  0.00-10.00   sec  35.9 GBytes 30.8 Gbits/sec
-----
Server listening on 5201
-----
^Ciperf3: interrupt - the server has terminated
root@mininet-vm:/home/mininet#
```

Рис. 3.7: 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 (tb) [1].

Изменим пропускную способность хоста `h1`, установив пропускную способность на 10 Гбит/с на интерфейсе `h1-eth0` и параметры TBF-фильтра (рис. 3.8):

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root tb
f rate 10gbit burst 5000000 limit 1500000
Error: Exclusivity flag on, cannot modify.
root@mininet-vm:/home/mininet#
```

Рис. 3.8: Изменение пропускной способности хоста `h1`

Фильтр `tb` 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. 3.9):

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

Свернуть все с

Рис. 3.9: Установка значения всплеска при ограничении скорости для фильтра tb

С помощью iPerf3 проверим, что значение пропускной способности изменилось.

В терминале хоста h2 запустим iPerf3 в режиме сервера (рис. 3.10):

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

Рис. 3.10: Запуск iPerf3 в режиме сервера на хосте h2

В терминале хоста h1 запустим iPerf3 в режиме клиента (рис. 3.11):

```
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 58358 connected to 10.0.0.2 port 5201
[ ID] Interval           Transfer     Bitrate      Retr   Cwnd
[ 7]  0.00-1.00      sec   1.05 GBytes  9.05 Gbits/sec  121   1.12 MBy
[ 7]  1.00-2.00      sec   1.12 GBytes  9.58 Gbits/sec   0    1.62 MBy
[ 7]  2.00-3.00      sec   1.11 GBytes  9.58 Gbits/sec   0    1.93 MBy
[ 7]  3.00-4.00      sec   1.11 GBytes  9.57 Gbits/sec   0    2.17 MBy
[ 7]  4.00-5.00      sec   1.11 GBytes  9.57 Gbits/sec   0    2.32 MBy
[ 7]  5.00-6.00      sec   1.11 GBytes  9.56 Gbits/sec   0    2.48 MBy
[ 7]  6.00-7.00      sec   1.11 GBytes  9.58 Gbits/sec   0    2.61 MBy
[ 7]  7.00-8.00      sec   1.11 GBytes  9.54 Gbits/sec   0    2.75 MBy
[ 7]  8.00-9.00      sec   1.11 GBytes  9.55 Gbits/sec   0    2.82 MBy
[ 7]  9.00-10.00     sec   1.09 GBytes  9.37 Gbits/sec   0    2.92 MBy
-----
[ ID] Interval           Transfer     Bitrate      Retr
[ 7]  0.00-10.00      sec  11.1 GBytes  9.49 Gbits/sec  121
[ 7]  0.00-10.00      sec  11.0 GBytes  9.48 Gbits/sec
iperf Done.
root@mininet-vm:/home/mininet#
```

Рис. 3.11: Запуск iPerf3 в режиме клиента на хосте h1

После завершения работы iPerf3 на хосте h1 остановим iPerf3 на хосте h2, нажав Ctrl + c (рис. 3.12):

```
"host: h2"@mininet-vm
-----
Server listening on 5201
-----
Accepted connection from 10.0.0.1, port 58356
[ 7] local 10.0.0.2 port 5201 connected to 10.0.0.1 port 58358
[ ID] Interval          Transfer      Bitrate
[ 7] 0.00-1.00 sec      1.03 GBytes  8.89 Gbits/sec
[ 7] 1.00-2.00 sec      1.12 GBytes  9.64 Gbits/sec
[ 7] 2.00-3.00 sec      1.12 GBytes  9.60 Gbits/sec
[ 7] 3.00-4.00 sec      1.11 GBytes  9.47 Gbits/sec
[ 7] 4.00-5.00 sec      1.12 GBytes  9.67 Gbits/sec
[ 7] 5.00-6.00 sec      1.11 GBytes  9.52 Gbits/sec
[ 7] 6.00-7.00 sec      1.10 GBytes  9.47 Gbits/sec
[ 7] 7.00-8.00 sec      1.13 GBytes  9.67 Gbits/sec
[ 7] 8.00-9.00 sec      1.10 GBytes  9.44 Gbits/sec
[ 7] 9.00-10.00 sec     1.10 GBytes  9.43 Gbits/sec
-----
[ ID] Interval          Transfer      Bitrate
[ 7] 0.00-10.00 sec    11.0 GBytes  9.48 Gbits/sec
-----
Server listening on 5201
-----
^Ciperf3: interrupt - the server has terminated
root@mininet-vm:/home/mininet#
```

Рис. 3.12: Остановка iPerf3

Удалим модифицированную конфигурацию на хосте h1 (рис. 3.13):

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

Рис. 3.13: 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. 3.14):

```
"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. 3.14: 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. 3.15):

```
"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. 3.15: Запуск iPerf3 в режиме сервера на хосте h2

В терминале хоста h1 запустим iPerf3 в режиме клиента (рис. 3.16):

```
"host: h1"@mininet-vm
iperf Done.
root@mininet-vm:/home/mininet# sudo tc qdisc del dev h1-eth0 root
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 58362 connected to 10.0.0.2 port 5201
[ ID] Interval          Transfer      Bitrate      Retr  Cwnd
[ 7]  0.00-1.00    sec   1.13 GBytes  9.67 Gbits/sec    0   5.69 MBytes
[ 7]  1.00-2.00    sec   1.11 GBytes  9.56 Gbits/sec    0   7.39 MBytes
[ 7]  2.00-3.00    sec   1.11 GBytes  9.56 Gbits/sec    0   8.26 MBytes
[ 7]  3.00-4.00    sec   1.11 GBytes  9.58 Gbits/sec    0   8.26 MBytes
[ 7]  4.00-5.00    sec   1.11 GBytes  9.55 Gbits/sec    0   8.26 MBytes
[ 7]  5.00-6.00    sec   1.11 GBytes  9.58 Gbits/sec    0   8.26 MBytes
[ 7]  6.00-7.00    sec   1.11 GBytes  9.56 Gbits/sec    0   8.26 MBytes
[ 7]  7.00-8.00    sec   1.11 GBytes  9.56 Gbits/sec    0   8.26 MBytes
[ 7]  8.00-9.00    sec   1.11 GBytes  9.55 Gbits/sec    0   8.26 MBytes
[ 7]  9.00-10.00   sec   1.12 GBytes  9.58 Gbits/sec    0   8.26 MBytes
-----
[ ID] Interval          Transfer      Bitrate      Retr
[ 7]  0.00-10.00    sec  11.1 GBytes  9.58 Gbits/sec    0
[ 7]  0.00-10.02    sec  11.1 GBytes  9.55 Gbits/sec
                                     sender
                                     receiver

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

Рис. 3.16: Запуск iPerf3 в режиме клиента на хосте h1

После завершения работы iPerf3 на хосте h1 остановим iPerf3 на хосте h2, нажав Ctrl + c (рис. 3.17):



```
"host: h2"@mininet-vm
-----
Server listening on 5201
-----
Accepted connection from 10.0.0.1, port 58360
[ 7] local 10.0.0.2 port 5201 connected to 10.0.0.1 port 58362
[ ID] Interval          Transfer      Bitrate
[ 7] 0.00-1.00 sec      1.11 GBytes  9.57 Gbits/sec
[ 7] 1.00-2.00 sec      1.11 GBytes  9.57 Gbits/sec
[ 7] 2.00-3.00 sec      1.11 GBytes  9.54 Gbits/sec
[ 7] 3.00-4.00 sec      1.12 GBytes  9.55 Gbits/sec
[ 7] 4.00-5.00 sec      1.11 GBytes  9.60 Gbits/sec
[ 7] 5.00-6.00 sec      1.11 GBytes  9.54 Gbits/sec
[ 7] 6.00-7.00 sec      1.12 GBytes  9.58 Gbits/sec
[ 7] 7.00-8.00 sec      1.11 GBytes  9.57 Gbits/sec
[ 7] 8.00-9.00 sec      1.11 GBytes  9.56 Gbits/sec
[ 7] 9.00-10.00 sec     1.11 GBytes  9.57 Gbits/sec
[ 7] 10.00-10.02 sec    1.93 MBytes  1.04 Gbits/sec
-----
[ ID] Interval          Transfer      Bitrate
[ 7] 0.00-10.02 sec    11.1 GBytes  9.55 Gbits/sec
-----
Server listening on 5201
-----
^Ciperf3: interrupt - the server has terminated
root@mininet-vm:/home/mininet#
```

Рис. 3.17: Остановка iPerf3

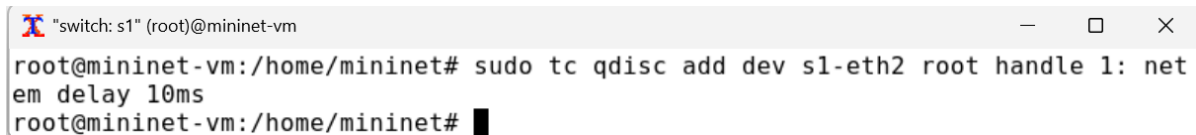
Удалим модифицированную конфигурацию на коммутаторе s1 (рис. 3.18):

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

Рис. 3.18: 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.

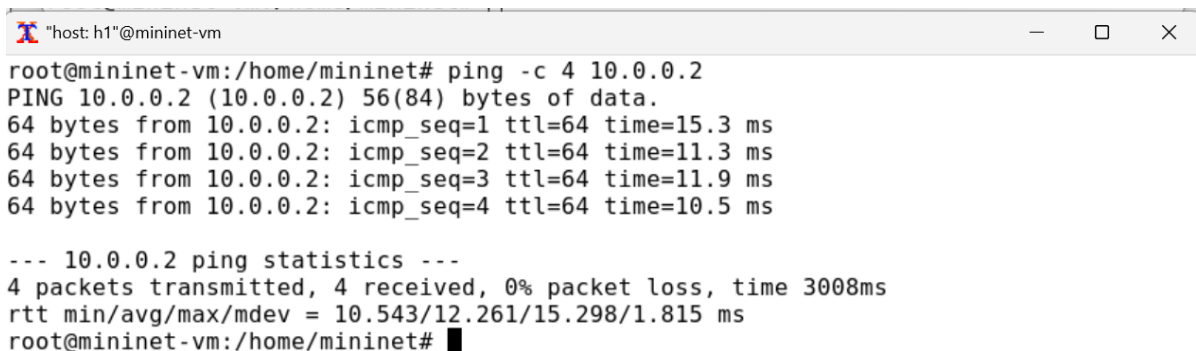
Объединим NETEM и TBF, введя на интерфейсе s1-eth2 коммутатора s1 задержку, джиттер, повреждение пакетов и указав скорость (рис. 3.19):



```
"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. 3.19: 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. 3.20):

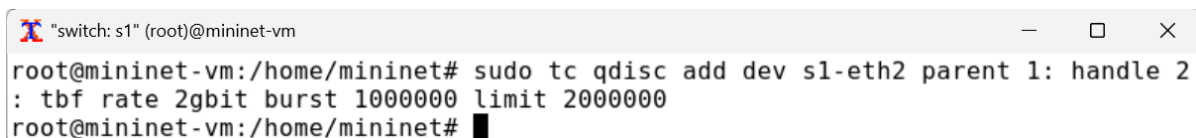


```
"host: h1" (root)@mininet-vm
root@mininet-vm:/home/mininet# ping -c 4 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=15.3 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=11.3 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=11.9 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=10.5 ms

--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3008ms
rtt min/avg/max/mdev = 10.543/12.261/15.298/1.815 ms
root@mininet-vm:/home/mininet#
```

Рис. 3.20: Проверка задержки

Добавим второе правило на коммутаторе s1, которое задаёт ограничение скорости с помощью `tbw` с параметрами `rate=2gbit`, `burst=1,000,000`, `limit=2,000,000` (рис. 3.21):



```
"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. 3.21: 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. 3.22):



```
"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. 3.22: Запуск iPerf3 в режиме сервера на хосте h2

В терминале хоста h1 запустим iPerf3 в режиме клиента (рис. 3.23):

```
"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 58366 connected to 10.0.0.2 port 5201
[ ID] Interval            Transfer        Bitrate        Retr  Cwnd
[ 7]  0.00-1.00      sec    215 MBytes    1.80 Gbits/sec    225   3.24 MBytes
[ 7]  1.00-2.00      sec    228 MBytes    1.91 Gbits/sec     0   3.48 MBytes
[ 7]  2.00-3.00      sec    228 MBytes    1.91 Gbits/sec     0   3.68 MBytes
[ 7]  3.00-4.00      sec    229 MBytes    1.92 Gbits/sec    45   2.71 MBytes
[ 7]  4.00-5.00      sec    228 MBytes    1.91 Gbits/sec     0   2.85 MBytes
[ 7]  5.00-6.00      sec    229 MBytes    1.92 Gbits/sec     0   2.96 MBytes
[ 7]  6.00-7.00      sec    228 MBytes    1.91 Gbits/sec     0   3.04 MBytes
[ 7]  7.00-8.00      sec    228 MBytes    1.91 Gbits/sec     0   3.10 MBytes
[ 7]  8.00-9.00      sec    229 MBytes    1.92 Gbits/sec     0   3.14 MBytes
[ 7]  9.00-10.00     sec    228 MBytes    1.91 Gbits/sec     0   3.17 MBytes
-----
[ ID] Interval            Transfer        Bitrate        Retr
[ 7]  0.00-10.00     sec    2.21 GBytes    1.90 Gbits/sec    270
[ 7]  0.00-10.02     sec    2.20 GBytes    1.89 Gbits/sec
                                     sender
                                     receiver

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

Рис. 3.23: Запуск iPerf3 в режиме клиента на хосте h1

После завершения работы iPerf3 на хосте h1 остановим iPerf3 на хосте h2, нажав Ctrl + c (рис. 3.24):

```
"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 58364
[ 7] local 10.0.0.2 port 5201 connected to 10.0.0.1 port 58366
[ ID] Interval           Transfer     Bitrate
[ 7] 0.00-1.00   sec      203 MBytes  1.70 Gbits/sec
[ 7] 1.00-2.01   sec      228 MBytes  1.91 Gbits/sec
[ 7] 2.01-3.00   sec      228 MBytes  1.92 Gbits/sec
[ 7] 3.00-4.00   sec      228 MBytes  1.91 Gbits/sec
[ 7] 4.00-5.00   sec      228 MBytes  1.91 Gbits/sec
[ 7] 5.00-6.00   sec      228 MBytes  1.91 Gbits/sec
[ 7] 6.00-7.00   sec      228 MBytes  1.92 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] 10.00-10.02 sec      1.30 MBytes  461 Mbits/sec
-----
[ ID] Interval           Transfer     Bitrate
[ 7] 0.00-10.02  sec      2.20 GBytes  1.89 Gbits/sec
-----
Server listening on 5201
-----
^Ciperf3: interrupt - the server has terminated
root@mininet-vm:/home/mininet#
```

Рис. 3.24: Остановка iPerf3

Удалим модифицированную конфигурацию на коммутаторе s1 (рис. 3.25):

```
"switch: s1" (root)@mininet-vm
: tbf 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. 3.25: Удаление модифицированной конфигурации на коммутаторе s1

Для самостоятельного задания создадим необходимые каталоги (рис. 3.26):

```
mininet@mininet-vm:~$ ls
Desktop  Downloads  mininet.orig  oflops  openflow  pox  Templates  work
Documents  mininet  Music        oftest  Pictures  Public  Videos
mininet@mininet-vm:~$ cd work
mininet@mininet-vm:~/work$ ls
lab_iperf3  lab_netem_i  lab_netem_ii  lesson1.mn
mininet@mininet-vm:~/work$ mkdir lab6
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. 3.26: Создание необходимых каталогов

Затем напишем скрипты по примеру из прошлых лабораторных работ (рис. 3.27 - рис. 3.29):

```

GNU nano 4.8                                exp1.py                                Modified
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 s1-eth2 root handle 1: netem delay 10ms' )
    s2.cmdPrint( 'tc qdisc add dev s1-eth2 parent 1: handle 2: tbf rate 2gbit burst 10>

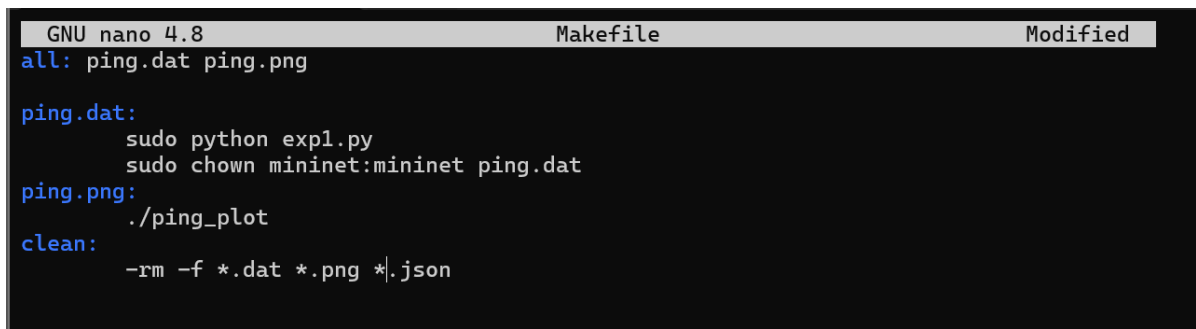
    info( '*** Traffic generation\n' )
    h2.cmdPrint( 'iperf3 -s -D -1' )
    time.sleep(10) # Wait 10 seconds for servers to start
    h1.cmdPrint( 'iperf3 -c', h2.IP(), '-J > iperf_result.json' )
    h1.cmdPrint( 'ping -c 100', h2.IP(), '| grep "time=" | awk \'{print $5, $7}\'' | se>

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

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

```

Рис. 3.27: Написание 1 скрипта



```
GNU nano 4.8 Makefile Modified
all: ping.dat ping.png

ping.dat:
    sudo python exp1.py
    sudo chown mininet:mininet ping.dat
ping.png:
    ./ping_plot
clean:
    -rm -f *.dat *.png *.json
```

Рис. 3.28: Написание 2 скрипта



```
mininet@mininet-vm: ~/work x + v
GNU nano 4.8 ping_plot Modified
#!/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. 3.29: Написание 3 скрипта

Запустим на выполнение скрипты для первой части самостоятельного задания (рис. 3.30):

```

mininet@mininet-vm:~/work/lab6/exp1$ make
sudo python 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: tbf rate 2gbit burst 1000000 li
mit 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, $7}\'' | sed -e
\'s/time=//g\' -e \'s/icmp_seq=//g\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 3 links
...
*** Stopping 2 switches
s1 s2
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab6/exp1$ |

```

Свернуть все окна

Рис. 3.30: Выполнение скриптов

Изменим параметры в скрипте для первого задания и запустим на выполнение (рис. 3.31 - рис. 3.32):

```
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 15000000' )

    info( '*** Traffic generation\n' )
    h2.cmdPrint( 'iperf3 -s -D -1' )
    time.sleep(10) # Wait 10 seconds for servers to start
    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. 3.31: Изменение параметров

```
mininet@mininet-vm:~/work/lab6/exp1$ make
sudo python expl.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 -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
expl.py iperf_result.json Makefile ping.dat ping_plot ping.png
mininet@mininet-vm:~/work/lab6/exp1$
```

Рис. 3.32: Выполнение скриптов

Посмотрим полученные графики (рис. 3.33 - рис. 3.34):

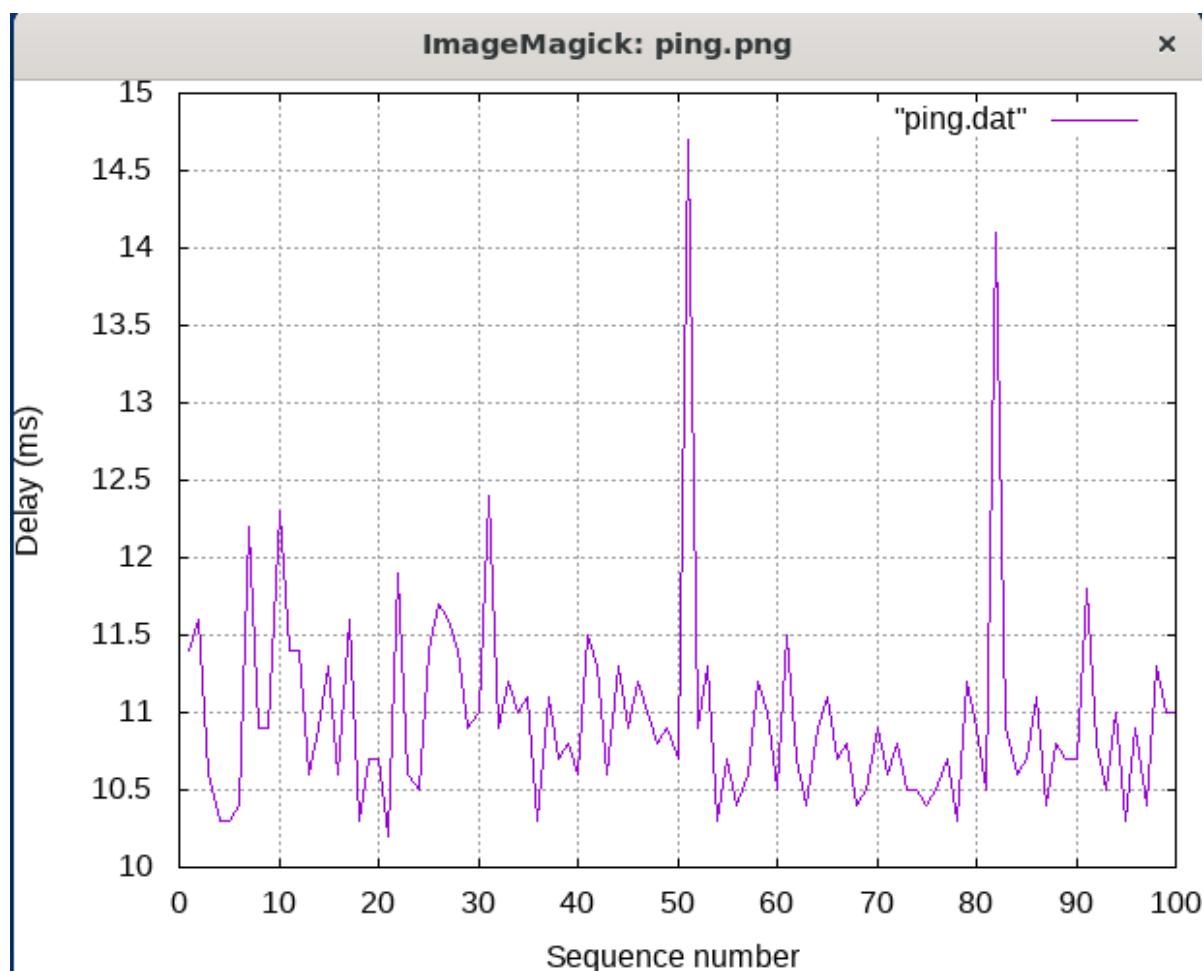


Рис. 3.33: График №1



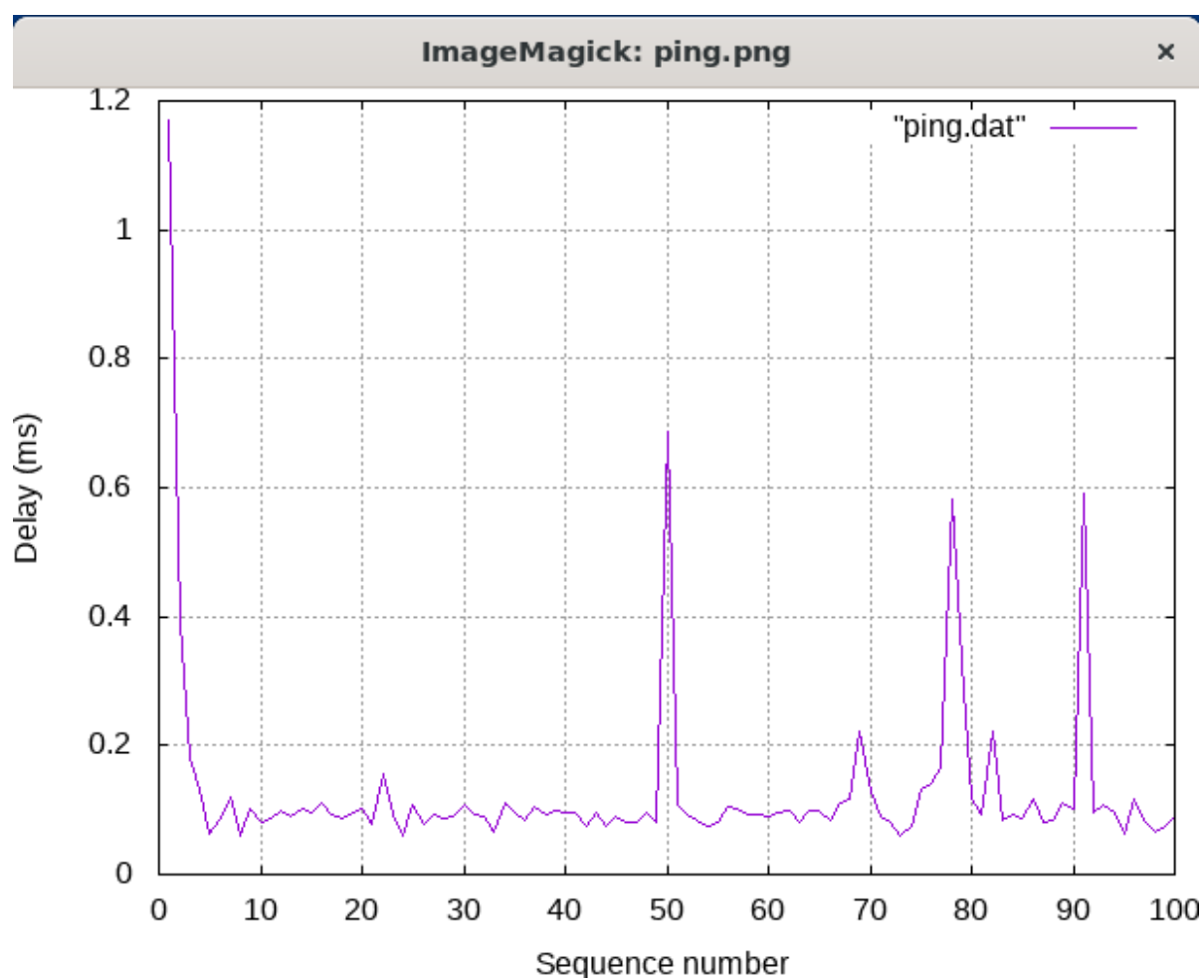


Рис. 3.34: График №2

## 4 Выводы

В ходе выполнения лабораторной работы мы познакомились 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.

# Список литературы

1. Token Bucket Filter [Электронный ресурс]. URL: [https://wiki.archlinux.org/title/Advanced\\_traffic\\_control](https://wiki.archlinux.org/title/Advanced_traffic_control).