

# **Лабораторная работа № 6**

**Моделирование сетей передачи данных**

**Доберштейн Алина Сергеевна**

# **Содержание**

<b>1 Цель работы</b>	<b>4</b>
<b>2 Задание</b>	<b>5</b>
<b>3 Теоретическое введение</b>	<b>6</b>
<b>4 Выполнение лабораторной работы</b>	<b>7</b>
<b>5 Интерактивные эксперименты</b>	<b>10</b>
5.1 Ограничение скорости на конечных хостах . . . . .	10
5.2 Ограничение скорости на коммутаторах . . . . .	11
5.3 Объединение NETEM и TBF . . . . .	12
<b>6 Воспроизводимые эксперименты</b>	<b>14</b>
6.1 Ограничение скорости на конечных хостах . . . . .	14
6.2 Ограничение скорости на коммутаторах . . . . .	17
6.3 Объединение NETEM и TBF . . . . .	19
<b>7 Выводы</b>	<b>21</b>
<b>Список литературы</b>	<b>22</b>

# Список иллюстраций

4.1	Задание топологии . . . . .	7
4.2	Подключение между хостами . . . . .	8
4.3	Запуск iperf3 на хостах . . . . .	9
5.1	Ограничение скорости на конечных хостах . . . . .	10
5.2	Результат эксперимента . . . . .	11
5.3	Ограничение скорости на коммутаторах . . . . .	12
5.4	Объединение NETEM и TBF . . . . .	12
5.5	Объединение NETEM и TBF . . . . .	12
5.6	Результат эксперимента . . . . .	13
6.1	Создание каталога для эксперимента . . . . .	14
6.2	Скрипт для воспроизводимого эксперимента . . . . .	15
6.3	Скрипт для отрисовки графика . . . . .	15
6.4	Запуск эксперимента . . . . .	16
6.5	График изменения скорости передачи . . . . .	16
6.6	Создание каталога для эксперимента . . . . .	17
6.7	Скрипт для воспроизводимого эксперимента . . . . .	17
6.8	Запуск эксперимента . . . . .	18
6.9	График изменения скорости передачи . . . . .	18
6.10	Создание каталога для эксперимента . . . . .	19
6.11	Скрипт для воспроизводимого эксперимента . . . . .	19
6.12	Запуск эксперимента . . . . .	20
6.13	График изменения скорости передачи . . . . .	20

# **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 назначенней 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 Теоретическое введение**

Mininet – это эмулятор компьютерной сети. Под компьютерной сетью подразумеваются простые компьютеры — хосты, коммутаторы, а так же OpenFlow-контроллеры. С помощью простейшего синтаксиса в примитивном интерпретаторе команд можно разворачивать сети из произвольного количества хостов, коммутаторов в различных топологиях и все это в рамках одной виртуальной машины(ВМ). На всех хостах можно изменять сетевую конфигурацию, пользоваться стандартными утилитами(ifconfig, ping) и даже получать доступ к терминалу. На коммутаторы можно добавлять различные правила и маршрутизировать трафик.

## 4 Выполнение лабораторной работы

Зададим простейшую топологию, состоящую из двух хостов и двух коммутаторов с назначенной по умолчанию mininet сетью 10.0.0.0/8. На хостах h1 и h2 введем команду ifconfig, чтобы отобразить информацию, относящуюся к их сетевым интерфейсам и назначенным им IP-адресам. Проверим подключение между хостами сети (рис. 4.1-4.2).

```
mininet@mininet-vm: /work$ xauth list $DISPLAY
mininet-vm/ unix:12 MIT-MAGIC-COOKIE-1 9713b86bb098fbfdb897dd6199ab032e
mininet@mininet-vm: /work$ sudo -i
root@mininet-vm:~# xauth list $DISPLAY
mininet-vm/ unix:12 MIT-MAGIC-COOKIE-1 9713b86bb098fbfdb897dd6199ab032e
root@mininet-vm:~# logout
mininet@mininet-vm:~/work$ 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:12.0
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...
*** Starting CLI:
```

Рис. 4.1: Задание топологии

```

1 switches
2                                     "host: h1"
n
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 898 bytes 254080 (254.0 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 898 bytes 254080 (254.0 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet/work# 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=6.13 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.071 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.073 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.085 ms

--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3057ms
rtt min/avg/max/mdev = 0.071/1.590/6.133/2.622 ms
root@mininet-vm:/home/mininet/work# 

0: <--> 10.0.0.2:1 loop "doubleclick" 1 mode "forward" <no data>
                                     "host: h2"
n
loop txqueuelen 1000 (Local Loopback)
RX packets 948 bytes 257136 (257.1 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 948 bytes 257136 (257.1 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet/work# ping c -4 10.0.0.1
ping: c: Temporary failure in name resolution
root@mininet-vm:/home/mininet/work# 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=9.76 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.589 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.068 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.074 ms

--- 10.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3041ms
rtt min/avg/max/mdev = 0.068/2.623/9.764/4.127 ms

```

Рис. 4.2: Подключение между хостами

Запустим iPerf3 на хостах и посмотрим результат отработки на данном этапе (рис. 4.3).

```

"host: h1"
--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3057ms
rtt min/avg/max/mdev = 0.071/1.590/6.133/2.622 ms
root@mininet-vm:/home/mininet/work# iperf3 -c 10.0.0.2
Connecting to host 10.0.0.2, port 5201
[ 7] local 10.0.0.1 port 35388 connected to 10.0.0.2 port 5201
[ ID] Interval Transfer Bitrate Retr Cwnd
[ 7] 0.00-1.00 sec 2.73 GBytes 23.4 Gbits/sec 0 8.10 MBytes
[ 7] 1.00-2.00 sec 2.80 GBytes 24.0 Gbits/sec 0 8.10 MBytes
[ 7] 2.00-3.00 sec 2.81 GBytes 24.2 Gbits/sec 0 8.10 MBytes
[ 7] 3.00-4.00 sec 2.89 GBytes 24.8 Gbits/sec 0 8.10 MBytes
[ 7] 4.00-5.00 sec 2.88 GBytes 24.7 Gbits/sec 0 8.10 MBytes
[ 7] 5.00-6.00 sec 2.79 GBytes 24.0 Gbits/sec 0 8.10 MBytes
[ 7] 6.00-7.00 sec 3.00 GBytes 25.8 Gbits/sec 0 8.10 MBytes
[ 7] 7.00-8.00 sec 2.85 GBytes 24.5 Gbits/sec 0 8.10 MBytes
[ 7] 8.00-9.00 sec 3.08 GBytes 26.5 Gbits/sec 0 8.10 MBytes
[ 7] 9.00-10.00 sec 2.82 GBytes 24.2 Gbits/sec 0 8.10 MBytes
[ 7] 0.00-10.00 sec 28.7 GBytes 24.6 Gbits/sec 0
[ 7] 0.00-10.01 sec 28.7 GBytes 24.6 Gbits/sec

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

"host: h2"
Server listening on 5201
-----
Accepted connection from 10.0.0.1, port 35386
[ 7] local 10.0.0.2 port 5201 connected to 10.0.0.1 port 35388
[ ID] Interval Transfer Bitrate
[ 7] 0.00-1.00 sec 2.73 GBytes 23.4 Gbits/sec
[ 7] 1.00-2.00 sec 2.79 GBytes 23.9 Gbits/sec
[ 7] 2.00-3.00 sec 2.82 GBytes 24.2 Gbits/sec
[ 7] 3.00-4.00 sec 2.89 GBytes 24.8 Gbits/sec
[ 7] 4.00-5.00 sec 2.87 GBytes 24.7 Gbits/sec
[ 7] 5.00-6.00 sec 2.80 GBytes 24.0 Gbits/sec
[ 7] 6.00-7.00 sec 3.01 GBytes 25.8 Gbits/sec
[ 7] 7.00-8.00 sec 2.85 GBytes 24.5 Gbits/sec
[ 7] 8.00-9.00 sec 3.08 GBytes 26.5 Gbits/sec
[ 7] 9.00-10.00 sec 2.82 GBytes 24.2 Gbits/sec
[ 7] 10.00-10.01 sec 11.1 MBytes 18.1 Gbits/sec
[ 7] 0.00-10.01 sec 28.7 GBytes 24.6 Gbits/sec

Server listening on 5201
-----
^C[iperf3: interrupted by user] the server has terminated

```

Рис. 4.3: Запуск iperf3 на хостах

# 5 Интерактивные эксперименты

## 5.1 Ограничение скорости на конечных хостах

Изменим пропускную способность хоста h1, установив пропускную способность на 10 Гбит/с на интерфейсе h1-eth0 и параметры TBF-фильтра (рис. 5.1-5.2).

```
root@mininet-vm:/home/mininet/work# sudo tc qdisc add dev h1-eth0 root tbf rate
10gbit burst 5000000 limit 15000000
root@mininet-vm:/home/mininet/work# egrep '^CONFIG_HZ_[0-9]+*' /boot/config-
`un
ame -r`
grep: /boot/config-: No such file or directory
grep: 5.4.0-42-generic: No such file or directory
root@mininet-vm:/home/mininet/work# egrep '^CONFIG_HZ_[0-9]+*' /boot/config-`una
me -r`
CONFIG_HZ_250=y
root@mininet-vm:/home/mininet/work# iperf3 -c 10.0.0.2
```

Рис. 5.1: Ограничение скорости на конечных хостах

```

"host: h1"
root@mininet-vm:/home/mininet/work# egrep '^CONFIG_HZ_[0-9]+' /boot/config-`uname -r`
CONFIG_HZ_250=y
root@mininet-vm:/home/mininet/work# iperf3 -c 10.0.0.2
Connecting to host 10.0.0.2, port 5201
[ 7] local 10.0.0.1 port 35392 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  7.34 MBytes
[ 7]  1.00-2.00  sec  1.11 GBytes  9.55 Gbits/sec  0  8.12 MBytes
[ 7]  2.00-3.00  sec  1.11 GBytes  9.58 Gbits/sec  0  8.12 MBytes
[ 7]  3.00-4.00  sec  1.11 GBytes  9.54 Gbits/sec  0  8.12 MBytes
[ 7]  4.00-5.00  sec  1.11 GBytes  9.58 Gbits/sec  0  8.12 MBytes
[ 7]  5.00-6.00  sec  1.11 GBytes  9.54 Gbits/sec  0  8.12 MBytes
[ 7]  6.00-7.00  sec  1.11 GBytes  9.60 Gbits/sec  0  8.12 MBytes
[ 7]  7.00-8.00  sec  1.11 GBytes  9.57 Gbits/sec  0  8.12 MBytes
[ 7]  8.00-9.00  sec  1.11 GBytes  9.54 Gbits/sec  0  8.12 MBytes
[ 7]  9.00-10.00 sec  1.11 GBytes  9.57 Gbits/sec  0  8.12 MBytes
[ ID] Interval      Transfer     Bitrate      Retr
[ 7]  0.00-10.00 sec  11.1 GBytes  9.57 Gbits/sec  0
[ 7]  0.00-10.01 sec  11.1 GBytes  9.56 Gbits/sec

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

"host: h2"
n= 201 "10.0.0.2" 11 grep "duplicator" 1 curl "socat SCPI/1 > /tmp/datas"
-----[REDACTED]-----
Server listening on 5201
-----[REDACTED]-----
Accepted connection from 10.0.0.1, port 35390
[ 7] local 10.0.0.2 port 5201 connected to 10.0.0.1 port 35392
[ ID] Interval      Transfer     Bitrate
[ 7]  0.00-1.00  sec  1.11 GBytes  9.56 Gbits/sec
[ 7]  1.00-2.00  sec  1.11 GBytes  9.50 Gbits/sec
[ 7]  2.00-3.00  sec  1.12 GBytes  9.56 Gbits/sec
[ 7]  3.00-4.00  sec  1.12 GBytes  9.62 Gbits/sec
[ 7]  4.00-5.01  sec  1.10 GBytes  9.45 Gbits/sec
[ 7]  5.01-6.00  sec  1.12 GBytes  9.67 Gbits/sec
[ 7]  6.00-7.00  sec  1.11 GBytes  9.58 Gbits/sec
[ 7]  7.00-8.00  sec  1.11 GBytes  9.58 Gbits/sec
[ 7]  8.00-9.00  sec  1.11 GBytes  9.52 Gbits/sec
[ 7]  9.00-10.01 sec  1.11 GBytes  9.55 Gbits/sec
[ ID] Interval      Transfer     Bitrate
[ 7]  0.00-10.01 sec  11.1 GBytes  9.56 Gbits/sec
-----[REDACTED]-----
Server listening on 5201
-----[REDACTED]-----
^CReceived interrupt - the receiver has terminated
-----[REDACTED]-----
```

Рис. 5.2: Результат эксперимента

## 5.2 Ограничение скорости на коммутаторах

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. 5.3).

The screenshot shows two terminal windows. The top window is titled "switch: s1" (root) and displays the command: `root@mininet-vm:/home/mininet/work# sudo tc qdisc add dev s1-eth2 root tbf rate 92.168.485 10gbit burst 5000000 limit 15000000`. The bottom window is titled "host: h1" and shows the output of an iperf3 test between host h1 and switch s1. The test results are as follows:

Interval	Transfer	Bitrate	Retr.
[ 7] 0.00-1.00 sec	9.64 GBytes	9.56 Gbits/sec	0 8.00 MBytes
[ 7] 1.00-2.00 sec	9.54 GBytes	9.54 Gbits/sec	0 8.00 MBytes
[ 7] 2.00-3.00 sec	9.54 GBytes	9.57 Gbits/sec	0 8.00 MBytes
[ 7] 3.00-4.00 sec	9.54 GBytes	9.56 Gbits/sec	0 8.00 MBytes
[ 7] 4.00-5.00 sec	9.54 GBytes	9.56 Gbits/sec	0 8.00 MBytes
[ 7] 5.00-6.00 sec	9.55 GBytes	9.55 Gbits/sec	0 8.00 MBytes
[ 7] 6.00-7.00 sec	9.53 GBytes	9.53 Gbits/sec	0 8.00 MBytes
[ 7] 7.00-8.00 sec	9.58 GBytes	9.58 Gbits/sec	0 8.00 MBytes
[ 7] 8.00-9.00 sec	9.55 GBytes	9.55 Gbits/sec	0 8.00 MBytes
[ 7] 9.00-10.00 sec	9.58 GBytes	9.58 Gbits/sec	0 8.00 MBytes

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

Рис. 5.3: Ограничение скорости на коммутаторах

## 5.3 Объединение NETEM и TBF

Объединим NETEM и TBF, введя на интерфейсе s1-eth2 коммутатора s1 задержку, джиттер, повреждение пакетов и указав скорость. (рис. 5.4).

The terminal window shows the configuration of a combined queue discipline (qdisc) on interface s1-eth2. It includes NETEM parameters (delay, loss, jitter) and TBF parameters (rate, burst, limit). The configuration command is:

```
root@mininet-vm:/home/mininet/work# sudo tc qdisc add dev s1-eth2 root handle 1: parent 1:0 netem delay 10ms loss 10% jitter 5ms tbf rate 10000000 burst 1000000 limit 2000000
```

Following this, an iperf3 test is run between host h1 and switch s1, showing stable bandwidth utilization at 10000000 bits/sec.

Рис. 5.4: Объединение NETEM и TBF

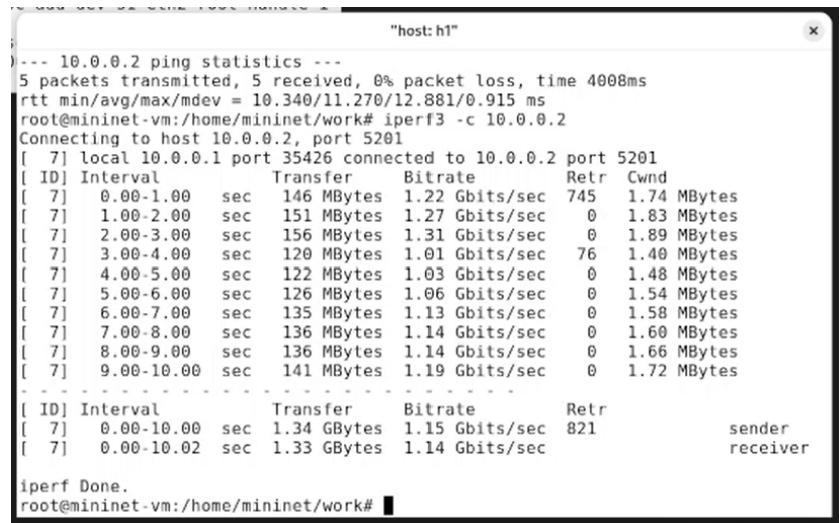
Добавим второе правило на коммутаторе s1, которое задаёт ограничение скорости с помощью tbf с параметрами rate=2gbit, burst=1,000,000, limit=2,000,000: и проверим (рис. 5.5).

The terminal window shows the addition of a second TBF rule to the queue discipline on interface s1-eth2. The command is:

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

Рис. 5.5: Объединение NETEM и TBF

Проверим конфигурацию с помощью iperf3 (рис. 5.6).



```
"host: h1"
--- 10.0.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 10.340/11.270/12.881/0.915 ms
root@mininet-vm:/home/mininet/work# iperf3 -c 10.0.0.2
Connecting to host 10.0.0.2, port 5201
[ 7] local 10.0.0.1 port 35426 connected to 10.0.0.2 port 5201
[ ID] Interval           Transfer     Bitrate      Retr  Cwnd
[ 7]  0.00-1.00   sec    146 MBytes   1.22 Gbits/sec  745  1.74 MBytes
[ 7]  1.00-2.00   sec    151 MBytes   1.27 Gbits/sec   0  1.83 MBytes
[ 7]  2.00-3.00   sec    156 MBytes   1.31 Gbits/sec   0  1.89 MBytes
[ 7]  3.00-4.00   sec    120 MBytes   1.01 Gbits/sec  76  1.40 MBytes
[ 7]  4.00-5.00   sec    122 MBytes   1.03 Gbits/sec   0  1.48 MBytes
[ 7]  5.00-6.00   sec    126 MBytes   1.06 Gbits/sec   0  1.54 MBytes
[ 7]  6.00-7.00   sec    135 MBytes   1.13 Gbits/sec   0  1.58 MBytes
[ 7]  7.00-8.00   sec    136 MBytes   1.14 Gbits/sec   0  1.60 MBytes
[ 7]  8.00-9.00   sec    136 MBytes   1.14 Gbits/sec   0  1.66 MBytes
[ 7]  9.00-10.00  sec    141 MBytes   1.19 Gbits/sec   0  1.72 MBytes
-
[ ID] Interval          Transfer     Bitrate      Retr
[ 7]  0.00-10.00  sec  1.34 GBytes  1.15 Gbits/sec  821             sender
[ 7]  0.00-10.02  sec  1.33 GBytes  1.14 Gbits/sec               receiver

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

Рис. 5.6: Результат эксперимента

# 6 Воспроизводимые эксперименты

## 6.1 Ограничение скорости на конечных хостах

В виртуальной среде mininet в своём рабочем каталоге с проектами создадим каталог host-tbf и перейдем в него. Создадим скрипт для эксперимента lab\_netem\_iii.py (рис. 6.1).

```
mininet@mininet-vm:~/work$ mkdir -p lab_netem_iii
mininet@mininet-vm:~/work$ cd lab_netem_iii
mininet@mininet-vm:~/work/lab_netem_iii$ mkdir -p host_tbf
mininet@mininet-vm:~/work/lab_netem_iii$ cd host_tbf
mininet@mininet-vm:~/work/lab_netem_iii/host_tbf$ cp ~/work/lab_netem_ii/simple-drop/Makefile ~/work/lab_netem_iii/host_tbf/Makefile
mininet@mininet-vm:~/work/lab_netem_iii/host_tbf$ ls
Makefile
mininet@mininet-vm:~/work/lab_netem_iii/host_tbf$ cp ~/work/lab_netem_ii/simple-drop/lab_netem_ii.py ~/work/lab_netem_iii/host_tbf/lab_netem_iii.py
mininet@mininet-vm:~/work/lab_netem_iii/host_tbf$ ls
lab_netem_iii.py Makefile
mininet@mininet-vm:~/work/lab_netem_iii/host_tbf$ cp ~/work/lab_netem_ii/simple-drop/Makefile ~/work/lab_netem_iii/host_tbf/Makefile
cp: cannot stat '/home/mininet/work/lab_netem_ii/simple-drop/Makefile': No such file or directory
mininet@mininet-vm:~/work/lab_netem_iii/host_tbf$ cp ~/work/lab_netem_ii/simple-delay/Makefile ~/work/lab_netem_iii/host_tbf/Makefile
mininet@mininet-vm:~/work/lab_netem_iii/host_tbf$ ls
lab_netem_iii.py Makefile
mininet@mininet-vm:~/work/lab_netem_iii/host_tbf$ cp ~/work/lab_netem_ii/simple-delay/ping_plot ~/work/lab_netem_iii/host_tbf/ping_plot
mininet@mininet-vm:~/work/lab_netem_iii/host_tbf$ ls
lab_netem_iii.py Makefile ping_plot
mininet@mininet-vm:~/work/lab_netem_iii/host_tbf$ mcedit lab_netem_iii.py
```

Рис. 6.1: Создание каталога для эксперимента

Отредактируем скрипт для проведения эксперимента (рис. 6.2).

```

/home/mininet/work/lab_neterm_iii/host_tbf/lab_neterm_iii.py [-M--] 32 L: 1+47 48/ 53 +(1142/1222b) 19 0x00A
#!/usr/bin/env python

"""

Simple experiment.
Output: ping.dat
"""

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

def emptyNet():

    "Create an empty network and add nodes to it."
    ...
    net = Mininet( controller=Controller, waitConnected=True )

    info('*** Adding controller\n')
    net.addController( 'c0' )

    info('*** Adding hosts\n')
    h1 = net.addHost( 'h1', ip='10.0.0.1' )
    h2 = net.addHost( 'h2', ip='10.0.0.2' )

    info('*** Adding switches\n')
    s1 = net.addSwitch('s1')
    s2 = net.addSwitch('s2')

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

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

    info('*** Set rate\n')
    h1.cmdPrint('sudo tc qdisc add dev h1-eth0 root tbf rate 10gbit burst 5000000 limit 15000000')

    time.sleep(10)

    info('*** Traffic generation\n')
    h2.cmdPrint('iperf3 -s &')
    h1.cmdPrint('iperf3 -c ' + h2.IP(), '| grep "NBytes" | awk \'{print $7}\'' > ping.dat')

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

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

```

Рис. 6.2: Скрипт для воспроизводимого эксперимента

Создадим также скрипт для визуализации ping\_plot результатов эксперимента (рис. 6.3).

```

/home/mininet/work/lab_neterm_iii/host_tbf/ping_plot [-M--]
#!/usr/bin/gnuplot --persist

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

```

Рис. 6.3: Скрипт для отрисовки графика

Запустим скрипт (рис. 6.4).

```

mininet@mininet-vm:~/work/lab_neterm_iii/host_tbf$ make
sudo python lab_neterm_iii.py
*** Adding controller
*** Adding hosts
*** Adding switches
*** 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 : ('sudo tc qdisc add dev h1-eth0 root tbf rate 10gbit burst 5000000 limit 15000000',)
*** Traffic generation
*** h2 : ('iperf3 -s &',)
*** h1 : ('iperf3 -c ', '10.0.0.2', '| grep "MBytes" | awk \'{print $7}\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 3 links
...
*** Stopping 2 switches
s1 s2
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_neterm_iii/host_tbf$ ls
lab_neterm_iii.py Makefile ping.dat ping_plot ping.png
mininet@mininet-vm:~/work/lab_neterm_iii/host_tbf$ geeqie ping.png

```

Рис. 6.4: Запуск эксперимента

Получим следующий график (рис. 6.5).

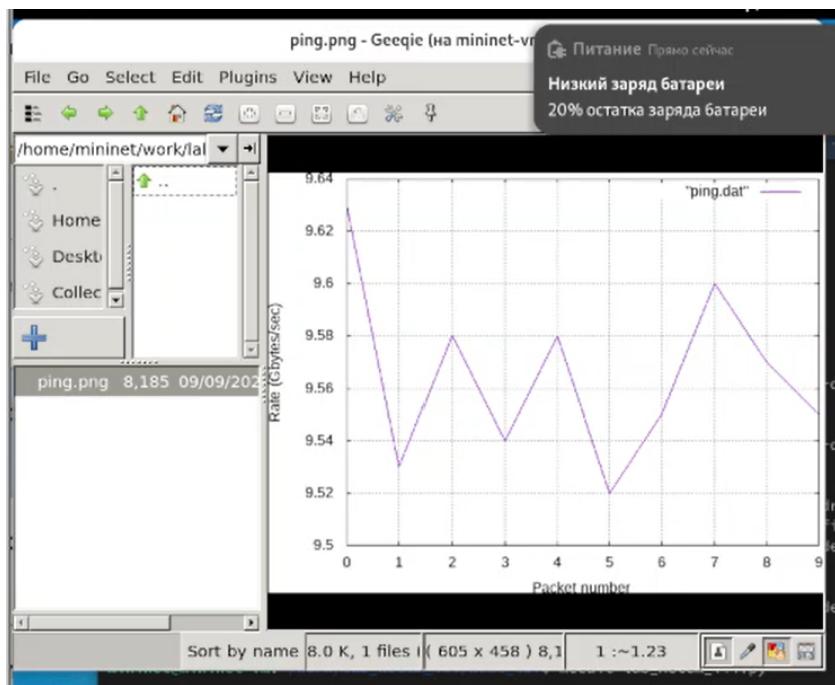


Рис. 6.5: График изменения скорости передачи

## 6.2 Ограничение скорости на коммутаторах

В виртуальной среде mininet в своём рабочем каталоге с проектами создадим каталог switch-tbf и перейдем в него. Создадим скрипт для эксперимента lab\_netem\_iii.py (рис. 6.6).

```
mininet@mininet-vm:/work/lab_netem_iii/host_tbf$ cd ..
mininet@mininet-vm:/work/lab_netem_iii$ mkdir -p switch_tbf
mininet@mininet-vm:/work/lab_netem_iii$ cd switch_tbf
mininet@mininet-vm:/work/lab_netem_iii/switch_tbf$ cp ./work/lab_netem_iii/host_tbf/lab_netem_iii.py ./work/lab_netem_iii/switch_tbf/lab_netem_iii.py
mininet@mininet-vm:/work/lab_netem_iii/switch_tbf$ cp ./work/lab_netem_iii/host_tbf/Makerfile ./work/lab_netem_iii/switch_tbf/Makerfile
mininet@mininet-vm:/work/lab_netem_iii/switch_tbf$ cp ./work/lab_netem_iii/host_tbf/ping_plot ./work/lab_netem_iii/switch_tbf/ping_plot
mininet@mininet-vm:/work/lab_netem_iii/switch_tbf$ chmod +x lab_netem_iii.py
```

Рис. 6.6: Создание каталога для эксперимента

Отредактируем скрипт для проведения эксперимента (рис. 6.7).

```
/home/mininet/work/lab_netem_iii/switch_tbf/lab_netem_iii.py  [---]  0 L:[ 1+37 38/ 53] *(803 /1222b)  16 0x00A
#!/usr/bin/env python

"""
Simple experiment.
Outputs ping.dat
"""

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

def emptyNet():
    """
    Create an empty network and add nodes to it.
    """
    net = Mininet( controller=Controller, waitConnected=True )

    info('*** Adding controller\n')
    net.addController('c0')

    info('*** Adding hosts\n')
    h1 = net.addHost('h1', ip='10.0.0.1')
    h2 = net.addHost('h2', ip='10.0.0.2')

    info('*** Adding switches\n')
    s1 = net.addSwitch('s1')
    s2 = net.addSwitch('s2')

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

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

    info('*** Set rate\n')
    s1.cmdPrint('sudo tc qdisc add dev eth2 root tbf rate 10gbit burst 5000000 limit 15000000')

    time.sleep(10)

    info('*** Traffic generation\n')
    h2.cmdPrint('iperf3 -s &')
    h1.cmdPrint('iperf3 -c ' + h2.IP(), '| grep "HBytes" | awk \'{print $7}\' > ping.dat')

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

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

Рис. 6.7: Скрипт для воспроизводимого эксперимента

Запустим эксперимент (рис. 6.8).

```

mininet@mininet-vm:~/work/lab_neterm_iii/switch_tbf$ make
sudo python lab_neterm_iii.py
*** Adding controller
*** Adding hosts
*** Adding switches
*** 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 : ('sudo tc qdisc add dev s1-eth2 root tbf rate 10gbit burst 5000000 limit 15000000',)
*** h2 : ('iperf3 -c ', '10.0.0.2', '| grep "MBytes" | awk \'{print $7\'\} > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 3 links
...
*** Stopping 2 switches
s1 s2
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_neterm_iii/switch_tbf$ geeqie ping.png
Gtk-Message: 17:05:28.022: Failed to load module "canberra-gtk-module"
Gtk-Message: 17:05:28.022: Failed to load module "pk-gtk-module"
mininet@mininet-vm:~/work/lab_neterm_iii/switch_tbf$ cat ping.dat
9.55
9.56
9.41
9.46
9.56
9.52
9.55
9.57
9.57
9.51
mininet@mininet-vm:~/work/lab_neterm_iii/switch_tbf$ cd ...

```

Рис. 6.8: Запуск эксперимента

Получим следующий график (рис. 6.9).

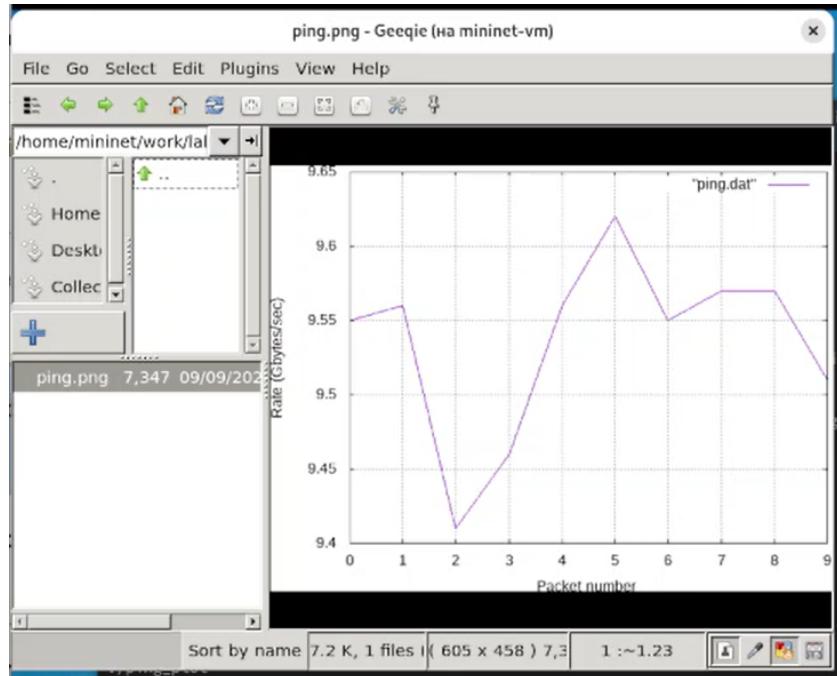


Рис. 6.9: График изменения скорости передачи

## 6.3 Объединение NETEM и TBF

В виртуальной среде mininet в своём рабочем каталоге с проектами создадим каталог simple=netem-tbf и перейдем в него. Создадим скрипт для эксперимента lab\_netem\_iii.py (рис. 6.10).

```
mininet@mininet-vm:~/work/lab_netem_iii$ cd ..
mininet@mininet-vm:~/work$ mkdir -p simple_netem_tbf
mininet@mininet-vm:~/work$ cd simple_netem_tbf
mininet@mininet-vm:~/work/simple$ cp ~/work/lab_netem_iii/host_tbf/lab_netem_iii.py ~/work/lab_netem_iii/simple_netem_tbf/lab_netem_iii.py
y
mininet@mininet-vm:~/work/lab_netem_iii/simple$ cp ~/work/lab_netem_iii/host_tbf/Makefile ~/work/lab_netem_iii/simple_netem_tbf/Makefile
mininet@mininet-vm:~/work/lab_netem_iii/simple$ cp ~/work/lab_netem_iii/host_tbf/ping_plot ~/work/lab_netem_iii/simple_netem_tbf/ping_file
mininet@mininet-vm:~/work/lab_netem_iii/simple$ chmod lab_netem_iii.py
```

Рис. 6.10: Создание каталога для эксперимента

Отредактируем скрипт для проведения эксперимента (рис. 6.11).

```
/home/mininet/work/lab_netem_iii/simple_netem_tbf/lab_netem_iii.py [-N--] 76 L{ 1+39 40/ 54}* (907 /1338b) 109 0x960
#!/usr/bin/env python

...
Simple experiment.
Output: ping.dat
...
from mininet.net import Mininet
from mininet.node import Controller
from mininet.cli import CLI
from mininet.log import setLogLevel, info
import time

def emptyNet():
    ...
    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 switches\n')
    s1 = net.addSwitch( 's1' )
    s2 = net.addSwitch( 's2' )

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

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

    info('*** Set rate\n')
    s1.cmdPrint('sudo tc qdisc add dev s1-eth2 root handle 1: netem delay 40ms corrupt 0.01%')
    s1.cmdPrint('sudo tc qdisc add dev s1-eth2 parent 1: handle 2: tbf rate 2gbit burst 1000000 limit 2000000')
    time.sleep(10)

    info('*** Traffic generation\n')
    h2.cmdPrint('iperf3 -s &')
    h1.cmdPrint('iperf3 -c ', h2.IP(), '| grep "0Bytes" | awk \'{print $7}\' > ping.dat')

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

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

Рис. 6.11: Скрипт для воспроизводимого эксперимента

Запустим эксперимент (рис. 6.12).

```

mininet@mininet-vm:~/work/lab_netem_111/simple_netem_tb$ make
sudo python lab_netem_111.py
*** Adding controller
*** Adding hosts
*** Adding switches
*** 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 : ('sudo tc qdisc add dev s1-eth2 root handle 1: netem delay 10ms corrupt 0.01%',)
*** s1 : ('sudo tc qdisc add dev s1-eth2 parent 1: handle 2: tbf rate 2gbit burst 1000000 limit 2000000',)
*** Traffic generation
*** h2 : ('iperf3 -s &')
*** h1 : ('iperf3 -c ', '10.0.0.2', '| grep "MBytes" | awk \'{print $7}\' > ping.dat')
*** Stopping network*** Stopping 1 controllers
c0
*** Stopping 3 links
...
*** Stopping 2 switches
s1 s2
*** Stopping 2 hosts
h1 h2
*** Done
sudo chown mininet:mininet ping.dat
./ping_plot
mininet@mininet-vm:~/work/lab_netem_111/simple_netem_tb$ geegle ping.png
Gtk-Message: 17:13:31.060: Failed to load module "canberra-gtk-module"
Gtk-Message: 17:13:31.060: Failed to load module "pk-gtk-module"
mininet@mininet-vm:~/work/lab_netem_111/simple_netem_tb$ cat ping.dat
1.32
1.66
1.65
1.37
1.06
954
973
1.04
1.04
1.11

```

Рис. 6.12: Запуск эксперимента

Получим следующий график (рис. 6.13).

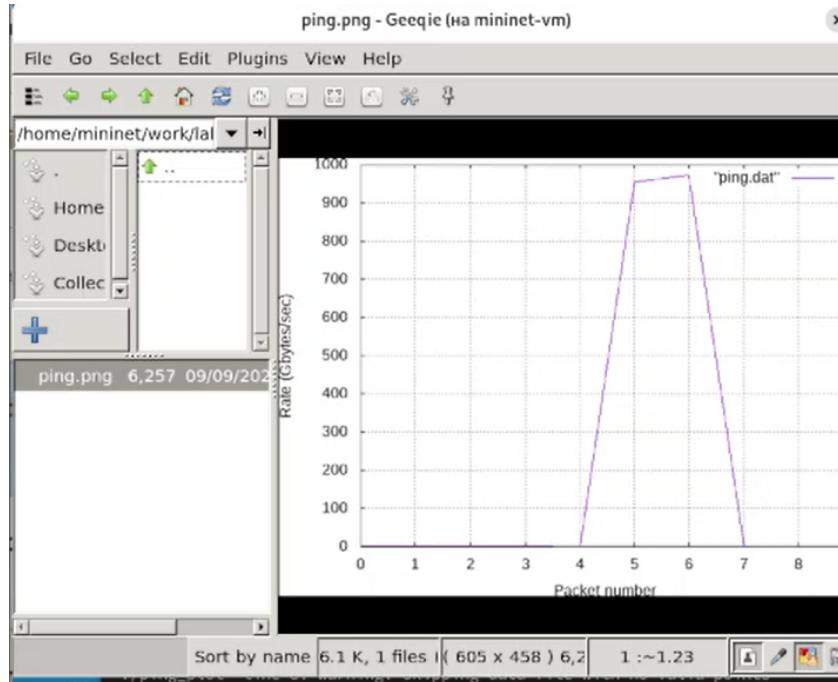


Рис. 6.13: График изменения скорости передачи

## **7 Выводы**

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

## **Список литературы**