

# **Лабораторная работа №6**

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

Барабанова Кристина

# Содержание

<b>Цель работы</b>	<b>4</b>
<b>Задание</b>	<b>5</b>
<b>Выполнение лабораторной работы</b>	<b>6</b>
1. Запуск лабораторной топологии . . . . .	6
2. Ограничение скорости на конечных хостах . . . . .	9
3. Ограничение скорости на коммутаторах . . . . .	10
4. Объединение NETEM и TBF . . . . .	11
5. Воспроизводимые эксперименты . . . . .	13
5. Выводы . . . . .	16

# Список иллюстраций

1	рис. 1.1	6
2	рис. 1.2	6
3	рис. 1.3	7
4	рис. 1.4	8
5	рис. 1.5	8
6	рис. 1.6	8
7	рис. 1.7	9
8	рис. 2.1	9
9	рис. 2.2	9
10	рис. 2.3	10
11	рис. 2.4	10
12	рис. 3.1	10
13	рис. 3.2	11
14	рис. 3.3	11
15	рис. 4.1	12
16	рис. 4.2	12
17	рис. 4.3	12
18	рис. 4.4	13
19	рис. 5.1	14
20	рис. 5.2	14
21	рис. 5.3	15
22	рис. 5.4	15
23	рис. 5.5	15

# 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.

# Задание

1. Задайте топологию (рис. 6.3), 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.

# Выполнение лабораторной работы

## 1. Запуск лабораторной топологии

1. Запустите виртуальную среду с mininet.
2. Из основной ОС подключитесь к виртуальной машине:

```
mininet@mininet-vm:~$ ssh -Y mininet@192.168.56.104
mininet@192.168.56.104'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

New release '22.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat Jan 31 10:28:32 2026 from 192.168.56.1
```

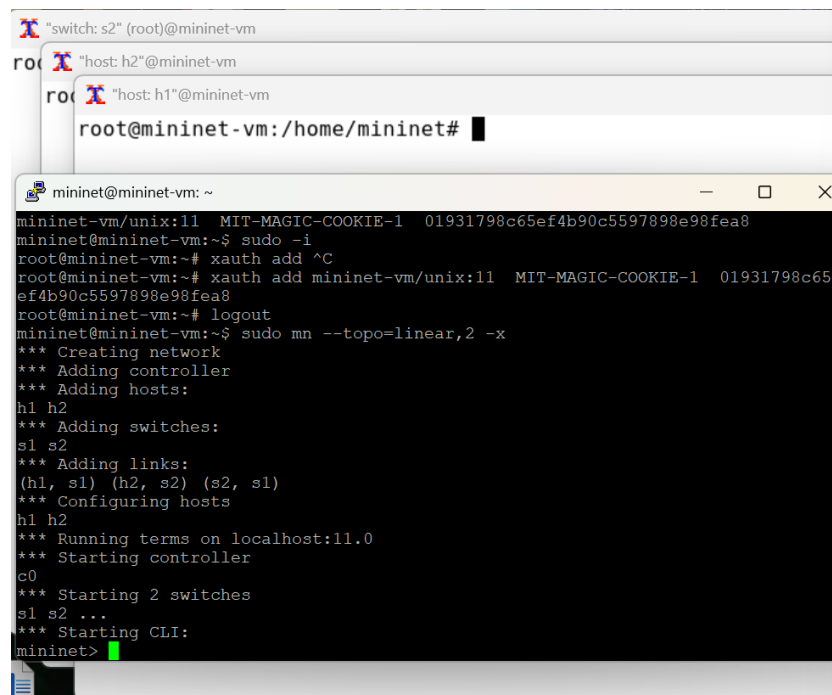
Рис. 1: рис. 1.1

3. В виртуальной машине mininet при необходимости исправьте права запуска X-соединения. Скопируйте значение куки (MIT magic cookie)<sup>1</sup> своего пользователя mininet в файл для пользователя root:

```
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:11 MIT-MAGIC-COOKIE-1 01931798c65ef4b90c5597898e98fea8
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth add ^C
root@mininet-vm:~# xauth add mininet-vm/unix:11 MIT-MAGIC-COOKIE-1 01931798c65ef4b90c5597898e98fea8
root@mininet-vm:~# logout
```

Рис. 2: рис. 1.2

4. Задайте топологию сети, состоящую из двух хостов и двух коммутаторов с назначенной по умолчанию mininet сетью 10.0.0.0/8:



```
switch: s2" (root)@mininet-vm
root "host: h2"@mininet-vm
root "host: h1"@mininet-vm
root@mininet-vm:/home/mininet#

mininet@mininet-vm: ~
mininet-vm/unix:11 MIT-MAGIC-COOKIE-1 01931798c65ef4b90c5597898e98fea8
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth add ^C
root@mininet-vm:~# xauth add mininet-vm/unix:11 MIT-MAGIC-COOKIE-1 01931798c65ef4b90c5597898e98fea8
root@mininet-vm:~# logout
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:11.0
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...
*** Starting CLI:
mininet>
```

Рис. 3: рис. 1.3

5. На хостах h1, h2 и на коммутаторах s1, s2 введите команду `ifconfig`, чтобы отобразить информацию, относящуюся к их сетевым интерфейсам и назначенным им IP-адресам.

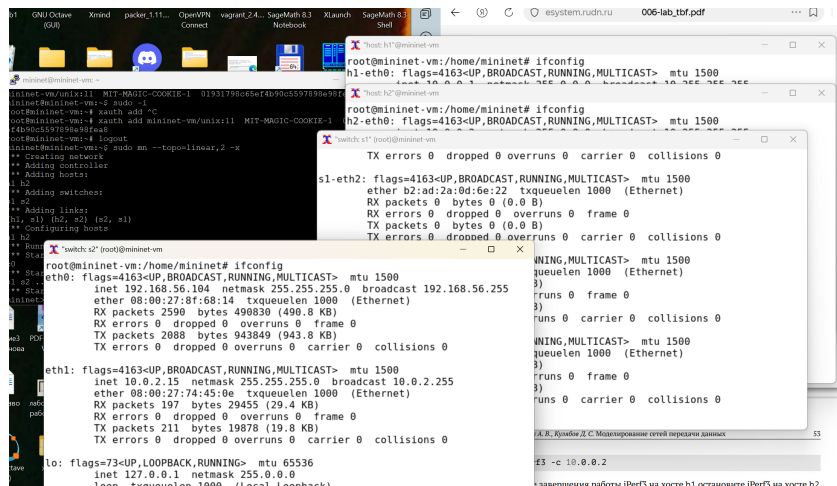


Рис. 4: рис. 1.4

- Проверьте подключение между хостами h1 и h2 с помощью команды `ping` с параметром `-c 4`.

```
root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 5
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=3.85 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.243 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.081 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.059 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.051 ms

--- 10.0.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4081ms
rtt min/avg/max/mdev = 0.051/0.856/3.848/1.497 ms
```

Рис. 5: рис. 1.5

- В терминале хоста h2 запустите `iPerf3` в режиме сервера:

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

Рис. 6: рис. 1.6

- В терминале хоста h1 запустите `iPerf3` в режиме клиента:



```

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 51122 connected to 10.0.0.2 port 5201
[ ID] Interval      Transfer    Bitrate      Retr  Cwnd
[ 7]  0.00-1.00    sec  3.30 GBytes  28.3 Gbits/sec    0   8.01 MBytes
[ 7]  1.00-2.00    sec  3.31 GBytes  28.5 Gbits/sec    0   8.01 MBytes
[ 7]  2.00-3.00    sec  3.31 GBytes  28.4 Gbits/sec    0   8.01 MBytes
[ 7]  3.00-4.00    sec  3.07 GBytes  26.3 Gbits/sec    0   8.01 MBytes
[ 7]  4.00-5.00    sec  3.36 GBytes  28.9 Gbits/sec    0   8.01 MBytes
[ 7]  5.00-6.00    sec  3.44 GBytes  29.5 Gbits/sec    0   8.01 MBytes
[ 7]  6.00-7.00    sec  3.39 GBytes  29.1 Gbits/sec    0   8.01 MBytes
[ 7]  7.00-8.00    sec  3.43 GBytes  29.5 Gbits/sec    0   8.01 MBytes
[ 7]  8.00-9.00    sec  3.44 GBytes  29.5 Gbits/sec    0   8.01 MBytes
[ 7]  9.00-10.00   sec  3.44 GBytes  29.5 Gbits/sec    0   8.01 MBytes
-----
[ ID] Interval      Transfer    Bitrate      Retr
[ 7]  0.00-10.00   sec  33.5 GBytes  28.8 Gbits/sec    0
[ 7]  0.00-10.00   sec  33.5 GBytes  28.7 Gbits/sec    0
sender
receiver

iperf Done.
-----

```

Рис. 7: рис. 1.7

## 2. Ограничение скорости на конечных хостах

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 (tbf). 1. Измените пропускную способность хоста h1, установив пропускную способность на 10 Гбит/с на интерфейсе h1-eth0 и параметры TBF-фильтра:

```

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

```

Рис. 8: рис. 2.1

3. С помощью `iPerf3` проверьте, что значение пропускной способности изменилось: – В терминале хоста h2 запустите `iPerf3` в режиме сервера:

```

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

```

Рис. 9: рис. 2.2

- В терминале хоста h2 запустите iPerf3 в режиме клиента:

```
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 51126 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    4.00 MBytes
[ 7] 1.00-2.00 sec  1.11 GBytes 9.57 Gbits/sec  0    4.00 MBytes
[ 7] 2.00-3.00 sec  1.11 GBytes 9.57 Gbits/sec  0    4.00 MBytes
[ 7] 3.00-4.00 sec  1.11 GBytes 9.56 Gbits/sec  0    4.00 MBytes
[ 7] 4.00-5.00 sec  1.11 GBytes 9.56 Gbits/sec  0    4.00 MBytes
[ 7] 5.00-6.00 sec  1.11 GBytes 9.56 Gbits/sec  0    4.00 MBytes
[ 7] 6.00-7.00 sec  1.11 GBytes 9.56 Gbits/sec  0    4.00 MBytes
[ 7] 7.00-8.00 sec  1.11 GBytes 9.58 Gbits/sec  0    4.00 MBytes
[ 7] 8.00-9.00 sec  1.11 GBytes 9.55 Gbits/sec  0    4.00 MBytes
[ 7] 9.00-10.00 sec 1.11 GBytes 9.57 Gbits/sec  0    4.00 MBytes
-----
[ ID] Interval      Transfer    Bitrate      Retr
[ 7] 0.00-10.00 sec 11.1 GBytes 9.58 Gbits/sec  9
[ 7] 0.00-10.01 sec 11.1 GBytes 9.56 Gbits/sec
sender
receiver

iperf Done.
```

Рис. 10: рис. 2.3

4. Удалите модифицированную конфигурацию на хосте h1:

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

Рис. 11: рис. 2.4

### 3. Ограничение скорости на коммутаторах

При ограничении скорости на интерфейсе s1-eth2 коммутатора s1 все сеансы связи между коммутатором s1 и коммутатором s2 будут фильтроваться в соответствии с применяемыми правилами. 1. Примените 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:

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

Рис. 12: рис. 3.1

2. Проверьте конфигурацию с помощью инструмента `iperf3` для измерения пропускной способности: – В терминале хоста `h2` запустите `iPerf3` в режиме сервера

– В терминале хоста `h2` запустите `iPerf3` в режиме клиента:

```
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 51130 connected to 10.0.0.2 port 5201
[ ID] Interval      Transfer    Bitrate      Retr    Cwnd
[ 7]  0.00-1.00    sec  1.13 GBytes  9.66 Gbits/sec    0    1.84 MBytes
[ 7]  1.00-2.00    sec  1.11 GBytes  9.58 Gbits/sec    0    2.44 MBytes
[ 7]  2.00-3.00    sec  1.11 GBytes  9.57 Gbits/sec    0    2.44 MBytes
[ 7]  3.00-4.00    sec  1.11 GBytes  9.56 Gbits/sec    0    2.62 MBytes
[ 7]  4.00-5.00    sec  1.11 GBytes  9.56 Gbits/sec    0    2.75 MBytes
[ 7]  5.00-6.00    sec  1.11 GBytes  9.57 Gbits/sec    0    2.75 MBytes
[ 7]  6.00-7.00    sec  1.11 GBytes  9.56 Gbits/sec    0    2.75 MBytes
[ 7]  7.00-8.00    sec  1.11 GBytes  9.56 Gbits/sec    0    2.75 MBytes
[ 7]  8.00-9.00    sec  1.11 GBytes  9.56 Gbits/sec    0    2.75 MBytes
[ 7]  9.00-10.00   sec  1.11 GBytes  9.56 Gbits/sec    0    2.75 MBytes
-----
[ ID] Interval      Transfer    Bitrate      Retr
[ 7]  0.00-10.00   sec  11.1 GBytes  9.58 Gbits/sec    0
[ 7]  0.00-10.01   sec  11.1 GBytes  9.56 Gbits/sec    0
                                     sender
                                     receiver

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

Рис. 13: рис. 3.2

3. Удалите модифицированную конфигурацию на коммутаторе `s1`:

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

Рис. 14: рис. 3.3

## 4. Объединение NETEM и TBF

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. 1. Объедините NETEM и TBF, введя на интерфейсе s1-eth2 коммутатора s1 задержку, джиттер, повреждение пакетов и указав скорость:

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

Рис. 15: рис. 4.1

2. Убедитесь, что соединение от хоста h1 к хосту h2 имеет заданную задержку. Для этого запустите команду ping с параметром -c 4 с терминала хоста h1

```
root@mininet-vm:/home/mininet# ping 10.0.0.2 -c 4
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
54 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=12.0 ms
54 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=10.8 ms
54 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=10.3 ms
54 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=10.1 ms

--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 10.122/10.821/12.006/0.731 ms
```

Рис. 16: рис. 4.2

3. Добавьте второе правило на коммутаторе s1, которое задаёт ограничение скорости с помощью tbf с параметрами rate=2gbit, burst=1,000,000, limit=2,000,000:

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

Рис. 17: рис. 4.3

4. Проверьте конфигурацию с помощью инструмента iperf3 для измерения пропускной способности: – В терминале хоста h2 запустите iPerf3 в режиме сервера:

– В терминале хоста h2 запустите iPerf3 в режиме клиента:

```

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 51134 connected to 10.0.0.2 port 5201
[ ID] Interval      Transfer    Bitrate      Retr  Cwnd
[ 7]  0.00-1.00    sec     219 MBytes  1.84 Gbits/sec  180  3.24 MBytes
[ 7]  1.00-2.00    sec     229 MBytes  1.92 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   0  3.84 MBytes
[ 7]  4.00-5.00    sec     228 MBytes  1.91 Gbits/sec  45  2.82 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  3.07 MBytes
[ 7]  7.00-8.00    sec     228 MBytes  1.91 Gbits/sec   0  3.16 MBytes
[ 7]  8.00-9.00    sec     229 MBytes  1.92 Gbits/sec   0  3.22 MBytes
[ 7]  9.00-10.00   sec     228 MBytes  1.91 Gbits/sec   0  3.27 MBytes
-----
[ ID] Interval      Transfer    Bitrate      Retr
[ 7]  0.00-10.00   sec    2.22 GBytes  1.91 Gbits/sec  225
[ 7]  0.00-10.01   sec    2.21 GBytes  1.90 Gbits/sec
sender
receiver

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

```

Рис. 18: рис. 4.4

5. Удалите модифицированную конфигурацию на коммутаторе s1:

## 5. Воспроизводимые эксперименты

Создала каталог для экспееримента

Написала скрипт для эксперимента

```

GNU nano 4.8                                lab tbf.py                                Modified
#!/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 switch\n' )
    s1 = net.addSwitch( 's1' )

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

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

    info( '*** Set delay\n' )
    h1.cmdPrint( 'tc qdisc add dev h1-eth0 root tbf rate 10gbit burst 5000000 limit 15000000' )
    h2.cmdPrint( 'tc qdisc add dev h2-eth0 root tbf rate 10gbit burst 5000000 limit 15000000' )

    time.sleep(10) # Wait 10 seconds

    info( '*** Ping\n' )
    h1.cmdPrint( 'ping -c 100', h2.IP(), '| grep "time=" | awk \'{print $5, $7}\'} | sed -e \'s/time=//g\' -e \'s/ /&/' )

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

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

```

Рис. 19: рис. 5.1

Написала скрипт для визуализации результатов эксперимента

```

GNU nano 4.8                                ping_plot
#!/usr/bin/gnuplot --persist

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

```

Рис. 20: рис. 5.2

Задала права доступа к файлу скрипта

Создала Makefile для управления процессом проведения эксперимента

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

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

ping.png: ping.dat
    ./ping_plot

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

Рис. 21: рис. 5.3

Выполнила эксперимент

Построила график

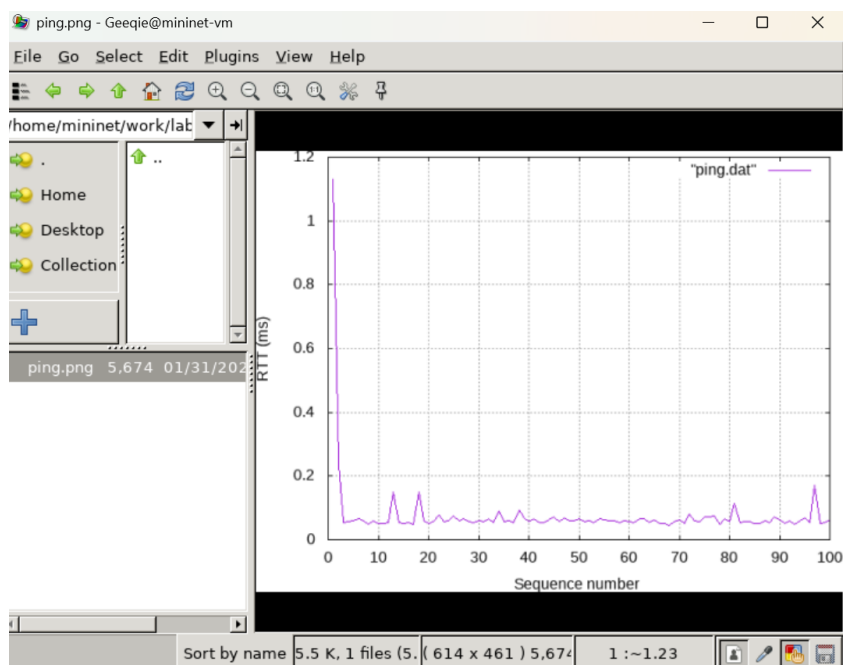


Рис. 22: рис. 5.4

Очистила каталог от результатов проведения экспериментов

```
mininet@mininet-vm:~/work/lab_tbf/simple-rate$ make clean
rm -f *.dat *.png
```

Рис. 23: рис. 5.5

---

## 5. Выводы

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.

---