

Презентация по лабораторной работе №6

Моделирование сетей передачи данных

Оразгелдиев Язгелди

2025-11-22

Содержание I

Информация

Докладчик

- Оразгелдиев Язгелди

Информация

Докладчик

- ▶ Оразгелдиев Язгелди
- ▶ 1032225075@pfur.ru

Информация

Докладчик

- ▶ Оразгелдиев Язгелди
- ▶ 1032225075@pfur.ru
- ▶ <https://github.com/YazgeldiOrazgeldiyev>

Цель

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.

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.

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

```
yazik@fedora:~$ ssh -Y mininet@192.168.56.104
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 03:10:29 2025 from 192.168.56.1
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  931d2a41599840b96ba89785c85ebf6e
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth add mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  931d2a41599840b96ba89785c85ebf6e
root@mininet-vm:~# xauth list $DISPLAY
mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  931d2a41599840b96ba89785c85ebf6e
root@mininet-vm:~# logout
```

Рисунок 1: Изменение права запуска X-соединения

Выполнение

```
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> ifconfig
*** Unknown command: ifconfig
mininet>
```

Рисунок 2: Создание топологии

Выполнение

```
"host: h1"
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 5a:be:69:de:da:89 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 970 bytes 315720 (315.7 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 970 bytes 315720 (315.7 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet# █
```

Рисунок 3: Команда *ifconfig* на хосте *h1*

Выполнение

```
"host: h2"
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 16:88:fd:84:6f:bd 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 978 bytes 326512 (326.5 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 978 bytes 326512 (326.5 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet# █
```

Рисунок 4: Команда *ifconfig* на хосте *h2*

Выполнение

```
"switch: s1" (root)
x

root@mininet-vm:/home/mininet# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.56.104 netmask 255.255.255.0 broadcast 192.168.56.255
                ether 08:00:27:78:60:33 txqueuelen 1000 (Ethernet)
                RX packets 2315 bytes 803376 (803.3 KB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 2061 bytes 1036828 (1.0 MB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
                ether 08:00:27:b4:66:9b txqueuelen 1000 (Ethernet)
                RX packets 242 bytes 32724 (32.7 KB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 267 bytes 24143 (24.1 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 5090 bytes 1654329 (1.6 MB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 5090 bytes 1654329 (1.6 MB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s1-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        ether ae:5a:61:ab:48:1a 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)
```

Выполнение

```
"switch: s2" (root)
root@mininet-vm:/home/mininet# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.56.104 netmask 255.255.255.0 broadcast 192.168.56.255
                ether 08:00:27:78:60:33 txqueuelen 1000 (Ethernet)
                RX packets 2489 bytes 832464 (832.4 KB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 2285 bytes 1225832 (1.2 MB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
                ether 08:00:27:b4:66:9b txqueuelen 1000 (Ethernet)
                RX packets 244 bytes 32874 (32.8 KB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 269 bytes 24293 (24.2 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 5465 bytes 1858557 (1.8 MB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 5465 bytes 1858557 (1.8 MB)
```

Рисунок 6: Команда *ifconfig* на коммутаторе *s2*

Выполнение

```
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.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.82 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.181 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.050 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.050 ms

--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3029ms
rtt min/avg/max/mdev = 0.050/0.526/1.823/0.750 ms
root@mininet-vm:/home/mininet# █
```

Рисунок 7: Пингование

Выполнение

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



Рисунок 8: Запуск сервера iPerf3

Выполнение

```
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 48050 connected to 10.0.0.2 port 5201
[ ID] Interval           Transfer     Bitrate      Retr  Cwnd
[ 7]  0.00-1.00   sec   4.83 GBytes  41.5 Gbits/sec   9  20.2 MBytes
[ 7]  1.00-2.00   sec   5.17 GBytes  44.4 Gbits/sec   0  20.2 MBytes
[ 7]  2.00-3.00   sec   5.08 GBytes  43.6 Gbits/sec   0  20.2 MBytes
[ 7]  3.00-4.00   sec   4.51 GBytes  38.7 Gbits/sec   0  20.2 MBytes
[ 7]  4.00-5.00   sec   4.72 GBytes  40.6 Gbits/sec   0  20.2 MBytes
[ 7]  5.00-6.00   sec   4.44 GBytes  38.2 Gbits/sec   0  20.2 MBytes
[ 7]  6.00-7.00   sec   4.60 GBytes  39.5 Gbits/sec   0  20.2 MBytes
[ 7]  7.00-8.00   sec   4.51 GBytes  38.8 Gbits/sec   0  20.2 MBytes
```

Рисунок 9: Запуск клиента iPerf3

Выполнение

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev h1-eth0 root tbf rate 10gbit burst 5000000  
limit 150000000
```

Рисунок 10: *Изменение пропускной способности на хосте*

Выполнение

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

Рисунок 11: Извлечение тактовой частоты

Выполнение

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

Рисунок 12: Запуск сервера iPerf3

Выполнение

```
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 48056 connected to 10.0.0.2 port 5201
[ ID] Interval      Transfer     Bitrate      Retr  Cwnd
[ 7]  0.00-1.00    sec   1.13 GBytes   9.69 Gbits/sec   9  3.00 MBytes
[ 7]  1.00-2.00    sec   1.11 GBytes   9.51 Gbits/sec   0  3.69 MBytes
[ 7]  2.00-3.00    sec   1.11 GBytes   9.55 Gbits/sec   0  3.69 MBytes
[ 7]  3.00-4.00    sec   1.11 GBytes   9.56 Gbits/sec   0  3.69 MBytes
[ 7]  4.00-5.00    sec   1.11 GBytes   9.50 Gbits/sec   0  3.69 MBytes
[ 7]  5.00-6.00    sec   1.11 GBytes   9.56 Gbits/sec   0  3.69 MBytes
[ 7]  6.00-7.00    sec   1.11 GBytes   9.50 Gbits/sec   0  3.69 MBytes
[ 7]  7.00-8.00    sec   1.11 GBytes   9.56 Gbits/sec   0  3.69 MBytes
[ 7]  8.00-9.00    sec   1.11 GBytes   9.56 Gbits/sec   0  3.69 MBytes
[ 7]  9.00-10.00   sec   1.11 GBytes   9.56 Gbits/sec   0  3.69 MBytes
[ ID] Interval      Transfer     Bitrate      Retr
[ 7]  0.00-10.00   sec  11.1 GBytes   9.56 Gbits/sec   9
[ 7]  0.00-10.01   sec  11.1 GBytes   9.54 Gbits/sec
sender
receiver
iperf Done.
```

Рисунок 13: Запуск клиента iPerf3

Выполнение

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

Рисунок 14: Удаление модифицированной конфигурации

Выполнение

```
root@mininet-vm:/home/mininet# sudo tc qdisc add dev s1-eth2 root tbf rate 10gbit b  
urst 5000000 limit 15000000  
root@mininet-vm:/home/mininet#
```

Рисунок 15: Изменение пропускной способности на коммутаторе

Выполнение

```
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 48868 connected to 10.0.0.2 port 5201
[ ID] Interval           Transfer     Bitrate      Retr Cwnd
[ 7]  0.00-1.00  sec   1.12 GBytes   9.65 Gbits/sec  0  1.30 MBytes
[ 7]  1.00-2.00  sec   1.11 GBytes   9.54 Gbits/sec  0  1.76 MBytes
[ 7]  2.00-3.00  sec   1.11 GBytes   9.56 Gbits/sec  0  1.76 MBytes
[ 7]  3.00-4.00  sec   1.11 GBytes   9.53 Gbits/sec  0  2.36 MBytes
[ 7]  4.00-5.00  sec   1.11 GBytes   9.52 Gbits/sec  0  2.60 MBytes
[ 7]  5.00-6.00  sec   1.11 GBytes   9.54 Gbits/sec  0  3.16 MBytes
[ 7]  6.00-7.00  sec   1.11 GBytes   9.56 Gbits/sec  0  3.16 MBytes
[ 7]  7.00-8.00  sec   1.11 GBytes   9.51 Gbits/sec  0  3.32 MBytes
[ 7]  8.00-9.00  sec   1.11 GBytes   9.57 Gbits/sec  0  3.48 MBytes
[ 7]  9.00-10.00 sec   1.11 GBytes   9.52 Gbits/sec  0  3.65 MBytes
[ ID] Interval          Transfer     Bitrate      Retr
[ 7]  0.00-10.00 sec  11.1 GBytes  9.55 Gbits/sec  0             sender
[ 7]  0.00-10.01 sec  11.1 GBytes  9.54 Gbits/sec               receiver
iperf Done.
```

Рисунок 16: Запуск клиента iPerf3

Выполнение

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

Рисунок 17: Изменение задержки на коммутаторе

Выполнение

```
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.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=13.1 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=10.9 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=10.7 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 3006ms
rtt min/avg/max/mdev = 10.695/11.351/13.116/1.021 ms
```

Рисунок 18: Пингование

Выполнение

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

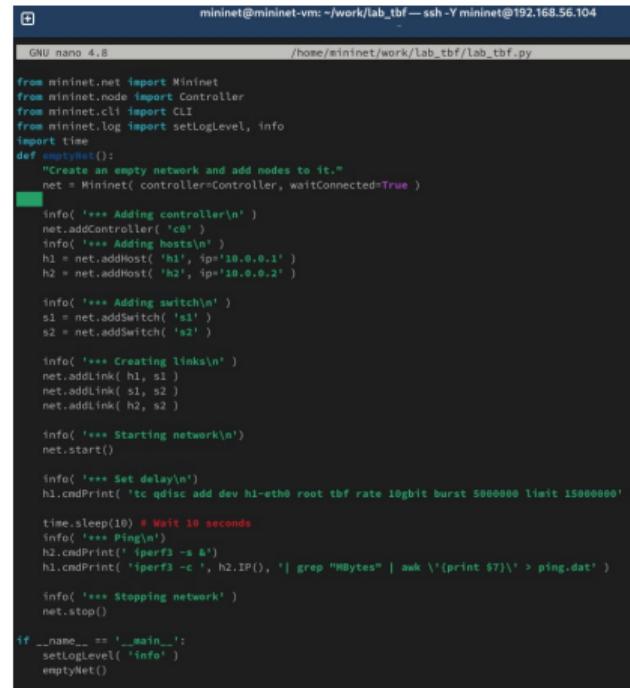
Рисунок 19: Изменение пропускной способности на коммутаторе

Выполнение

```
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 48864 connected to 10.0.0.2 port 5201
[ ID] Interval      Transfer     Bitrate    Retr  Cwnd
[ 7]  0.00-1.00  sec   217 MBytes  1.82 Gbits/sec  1260  2.91 MBytes
[ 7]  1.00-2.00  sec   228 MBytes  1.91 Gbits/sec  0  3.04 MBytes
[ 7]  2.00-3.00  sec   229 MBytes  1.92 Gbits/sec  0  3.15 MBytes
[ 7]  3.00-4.00  sec   228 MBytes  1.91 Gbits/sec  0  3.23 MBytes
[ 7]  4.00-5.00  sec   229 MBytes  1.92 Gbits/sec  0  3.29 MBytes
[ 7]  5.00-6.00  sec   228 MBytes  1.91 Gbits/sec  0  3.33 MBytes
[ 7]  6.00-7.00  sec   228 MBytes  1.91 Gbits/sec  0  3.35 MBytes
[ 7]  7.00-8.00  sec   229 MBytes  1.92 Gbits/sec  0  3.37 MBytes
[ 7]  8.00-9.00  sec   228 MBytes  1.91 Gbits/sec  0  3.37 MBytes
[ 7]  9.00-10.00 sec   229 MBytes  1.92 Gbits/sec  0  3.37 MBytes
[ 7]  0.00-10.00 sec  228 MBytes  1.91 Gbits/sec  0  3.37 MBytes
[ 7]  0.00-10.01 sec  2.21 GBytes  1.89 Gbits/sec
                                         sender
                                         receiver
iperf Done.
```

Рисунок 20: Запуск клиента iPerf3

Выполнение



The screenshot shows a terminal window titled "mininet@mininet-vm: ~/work/lab_tbf — ssh -Y mininet@192.168.56.104". The window displays the content of the lab_tbf.py script. The script creates a network with two hosts (h1, h2) and two switches (s1, s2), connects them, starts the network, and performs a ping test between h1 and h2. It then stops the network and exits.

```
mininet@mininet-vm: ~/work/lab_tbf — ssh -Y mininet@192.168.56.104

GNU nano 4.8 /home/mininet/work/lab_tbf/lab_tbf.py

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' )
    s2 = net.addSwitch( 's2' )

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

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

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

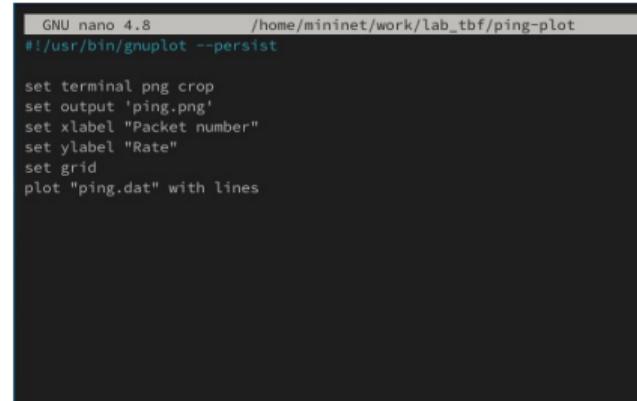
    time.sleep(10) # Wait 10 seconds
    info( '*** Ping\n' )
    h2.cmdPrint('iperf3 -s &')
    h1.cmdPrint('iperf3 -c ' + h2.IP(), '| grep "MBytes" | awk \'{print $7}\' > ping.dat')

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

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

Рисунок 21: Файл *lab_tbf.py*

Выполнение

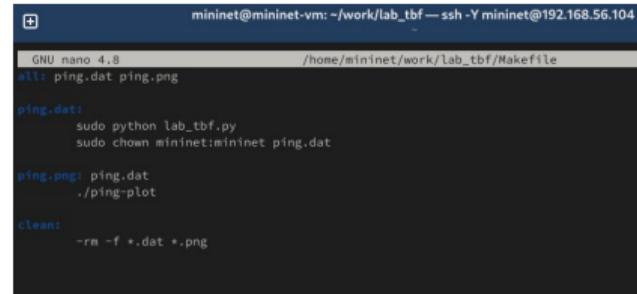


```
GNU nano 4.8          /home/mininet/work/lab_tbf/ping-plot
#!/usr/bin/gnuplot --persist

set terminal png crop
set output 'ping.png'
set xlabel "Packet number"
set ylabel "Rate"
set grid
plot "ping.dat" with lines
```

Рисунок 22: Файл *ping-plot*

Выполнение



The screenshot shows a terminal window titled "mininet@mininet-vm: ~/work/lab_tbf — ssh -Y mininet@192.168.56.104". The window displays a Makefile with the following content:

```
GNU nano 4.8 /home/mininet/work/lab_tbf/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
```

Рисунок 23: *Makefile*

Выполнение

```
mininet@mininet-vm:~/work/lab_tbf$ make
sudo python lab_tbf.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
*** h1 : ('tc qdisc add dev h1-eth0 root tbf rate 10gbps burst 5000000 limit 15000000',)
*** Ping
*** 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
```

Рисунок 24: Запуск эксперимента

Выполнение

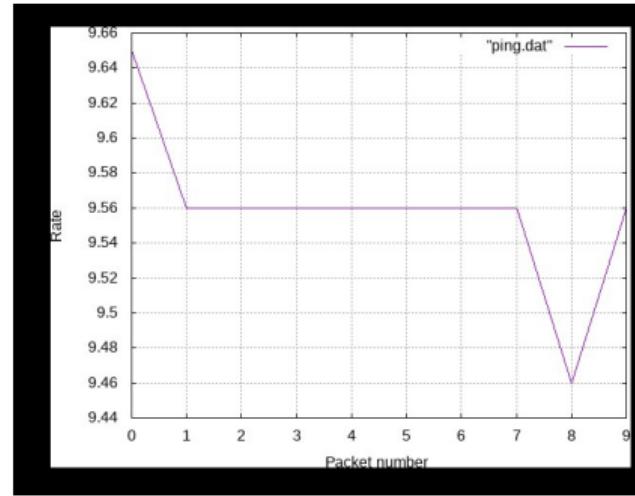


Рисунок 25: График пропускной способности

Выполнение

```
info( '*** Set delay\n')
h1.cmdPrint( 'tc qdisc add dev h1-eth2 root tbf rate 6gbit burst 5000000 limit 15000000' )
```

Рисунок 26: Изменение файла *lab_tbf.py*

Выполнение

```
mininet@mininet-vm:~/work/lab_tbf$ make
sudo python lab_tbf.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
*** h1 : ('tc qdisc add dev h1-eth2 root tbf rate 6gbit burst 5000000 limit 15000000',)
Cannot find device "h1-eth2"
*** Ping
*** h2 : ('iperf3 -s &',)
*** h1 : ('iperf3 -c ', '192.0.0.2', '| 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
```

Рисунок 27: Запуск эксперимента

Выполнение

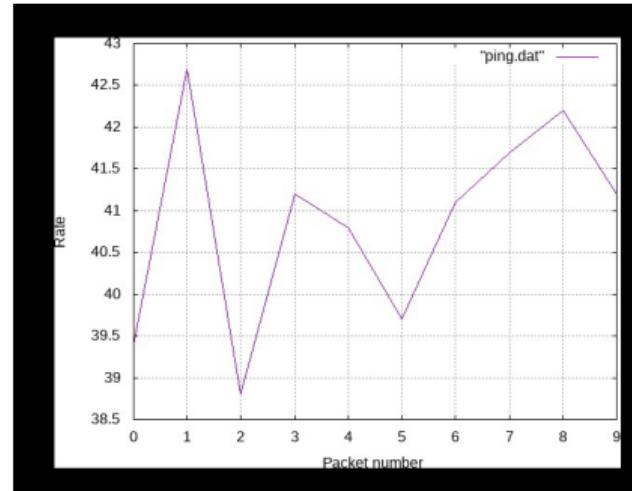


Рисунок 28: График пропускной способности

Выполнение

```
    info{ '*** Set delay\n'}
    s1.cmdPrint('tc qdisc add dev s1-eth2 root handle 1: netem delay 20ms')
    s1.cmdPrint('tc qdisc add dev s1-eth2 parent 1: tbf rate 2gbit burst 1000000 limit 2000000')
```

Рисунок 29: Изменение файла *lab_tbf.py*

Выполнение

```
mininet@mininet-vm:~/work/lab_tbf$ make
sudo python lab_tbf.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',)
*** s1 : ('tc qdisc add dev s1-eth2 parent 1: handle 2: tbf rate 2gbps burst 1000000 limit 2000000',)
*** Ping
*** 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
```

Рисунок 30: Запуск эксперимента

Выполнение

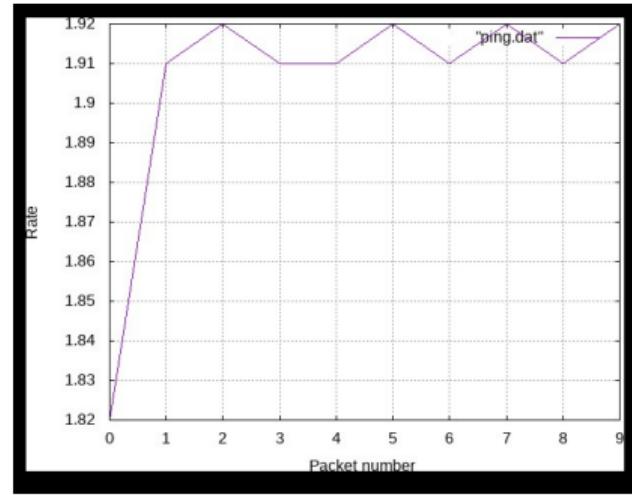


Рисунок 31: График пропускной способности

Вывод

Я ознакомился с принципами работы дисциплины очереди Token Bucket Filter, а также получил навыки моделирования и исследования поведения трафика посредством проведения интерактивного и воспроизводимого экспериментов в Mininet.