

Лабораторная работа №6

Моделирование сетей передачи данных

Матюхин П.А.

Российский университет дружбы народов, Москва, Россия

22 ноября 2025

Докладчик

- ▶ Матюхин Павел
Андреевич
- ▶ Студент группы
НПИбд-01-22
- ▶ Студ. билет
1132226527
- ▶ Российский
университет
дружбы народов
имени Патриса
Лумумбы

Цель лабораторной 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.

Выполнение лабораторной работы

Подготовка инструментария к работе

Запуск лабораторной топологии

The image shows four terminal windows running on a host machine. The windows are titled "switch:s2" (root), "switch:s1" (root), "host:h1", and "host:h2". Each window displays the output of the 'ifconfig' command, showing network interface statistics for MTU, broadcast address, and various packet counters (RX bytes, TX bytes, errors, dropped, overruns, collisions).

switch:s2 (root)

```
0 dropped 0 overruns 0 carrier 0 collisions 0
B<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
ether 3:b3:a2:47:fd txqueuelen 1000 (Ethernet)
0 bytes 0 (0.0 B)
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

s2-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
ether 8a:51:cf:33:a4:10 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

s2-eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
ether 4e:9a:f4:76:17:56 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

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

switch:s1 (root)

```
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
s1-eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
ether aa:03:b3:a2:47:fd 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

s2-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
ether 8a:51:cf:33:a4:10 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

s2-eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
ether 4e:9a:f4:76:17:56 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

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

host:h1

```
@mininet-vm:/home/mininet# ifconfig
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 7e:b5:d3:8a:29:e7 txqueuelen 1000 (Ethernet)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
loop txqueuelen 1000 (Local Loopback)
RX packets 1161 bytes 577888 (577.8 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 1161 bytes 577888 (577.8 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

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 52:c6:4d:81:11:f9 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

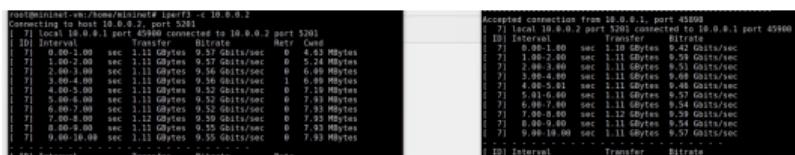
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
loop txqueuelen 1000 (Local Loopback)
RX packets 936 bytes 554364 (554.3 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 936 bytes 554364 (554.3 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

Интерактивные эксперименты с ограничением скорости на конечных хостах

```
[root@mininet-vm ~]# tc qdisc add dev h1-eth0 root tbf rate 10gb  
it burst 5000000 limit 15000000  
root@mininet-vm:~/home/mininet# grep 'CONFIG_HZ_[0-9]' /boot/config-'uname -r'  
grep: /boot/config-uname -r: No such file or directory  
root@mininet-vm:~/home/mininet# egrep '^CONFIG_HZ_[0-9]' /boot/config-'uname -r'  
CONFIG_HZ_250=y  
root@mininet-vm:~/home/mininet#
```

Измените пропускную способность хоста h1, установив пропускную способность на 10 Гбит/с на интерфейсе h1-eth0 и параметры TBF-фильтра. Фильтр tbf 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 HZ, где HZ — тактовая частота, настроенная как параметр ядра, и может быть извлечена с помощью следующей команды



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

Самостоятельная работа

```
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 )

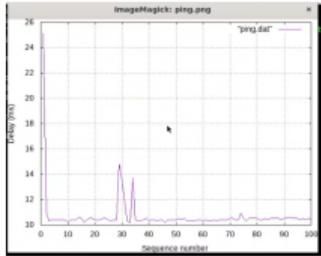
info( '*** Starting network\n' )
net.start()

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 1000000 limit 2000000' )

info( '*** Set traffic modulation\n' )
h2.cmd('nt(iperf3 -s -D )')
time.sleep(10)
h1.cmdPrint('iperf3 -c ' + h2.IP(), '- - >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\n' )
net.stop()
```

```
#!/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
```



```
all: ping.dat ping.png
ping.dat:
    sudo python lab_neterm_1.py
    sudo chown mininet:mininet ping.dat
ping.png:
    ./ping_plot
clean:
    rm *.dat *.png
```

ping.dat script(pyton)

Вывод

Вывод

- ▶ Ознакомился с принципами работы дисциплины очереди Token Bucket Filter, которая формирует входящий/исходящий трафик для ограничения пропускной способности, а также получил навыки моделирования и исследования поведения трафика посредством проведения интерактивного и воспроизводимого экспериментов в Mininet.