# Лабораторная работа №6

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

Ким Реачна[1]

21 декабрь, 2023, Москва, Россия

[1]Российский Университет Дружбы Народов

# Цели и задачи

# 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.

# 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.

# Процесс выполнения лабораторной работы

# Запуск лабораторной топологии

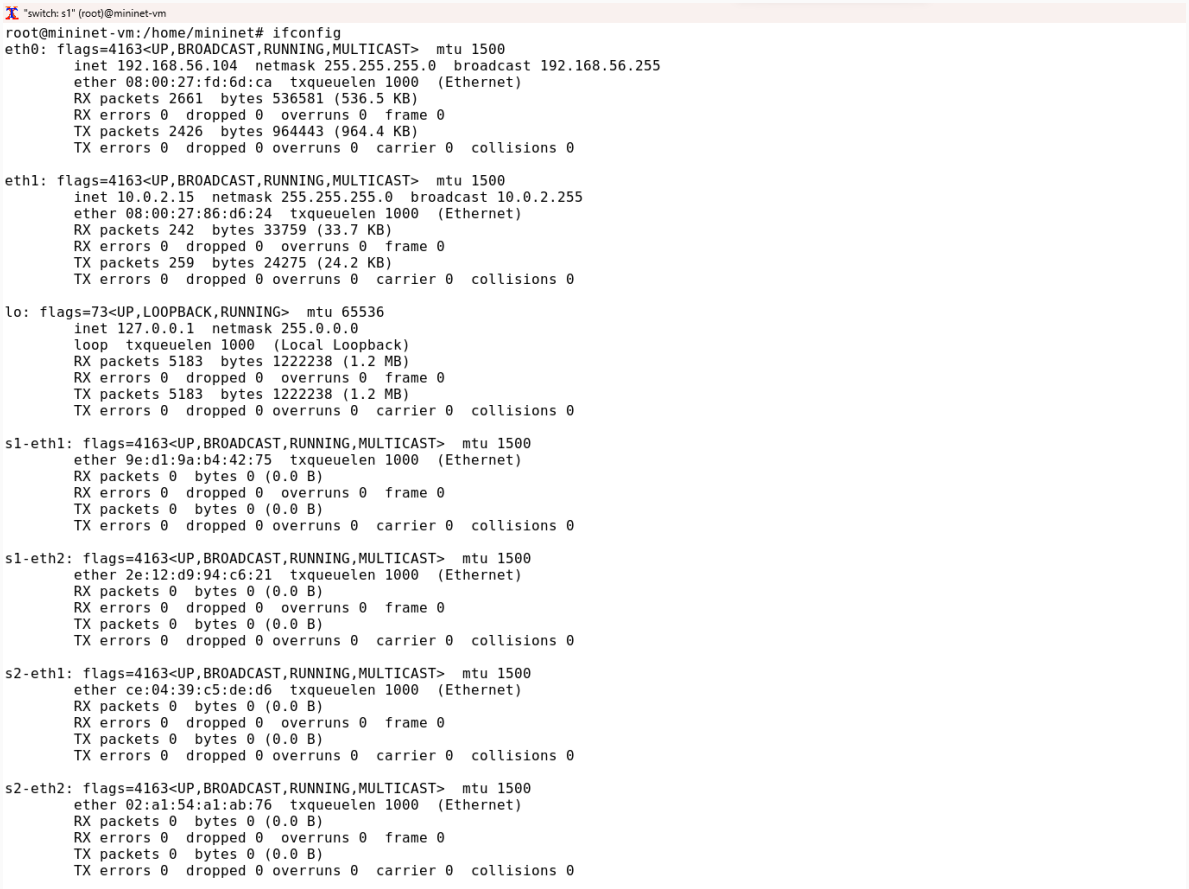


Рис. 1: Информация о сетевом интерфейсе и IP-адресе s1

# Запуск лабораторной топологии

```
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=2.47 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.221 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.058 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.127 ms

--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3045ms
rtt min/avg/max/mdev = 0.058/0.718/2.467/1.011 ms
root@mininet-vm:/home/mininet#
```

Рис. 2: Проверка подключение от h2 к h1

# Ограничение скорости на конечных хостах

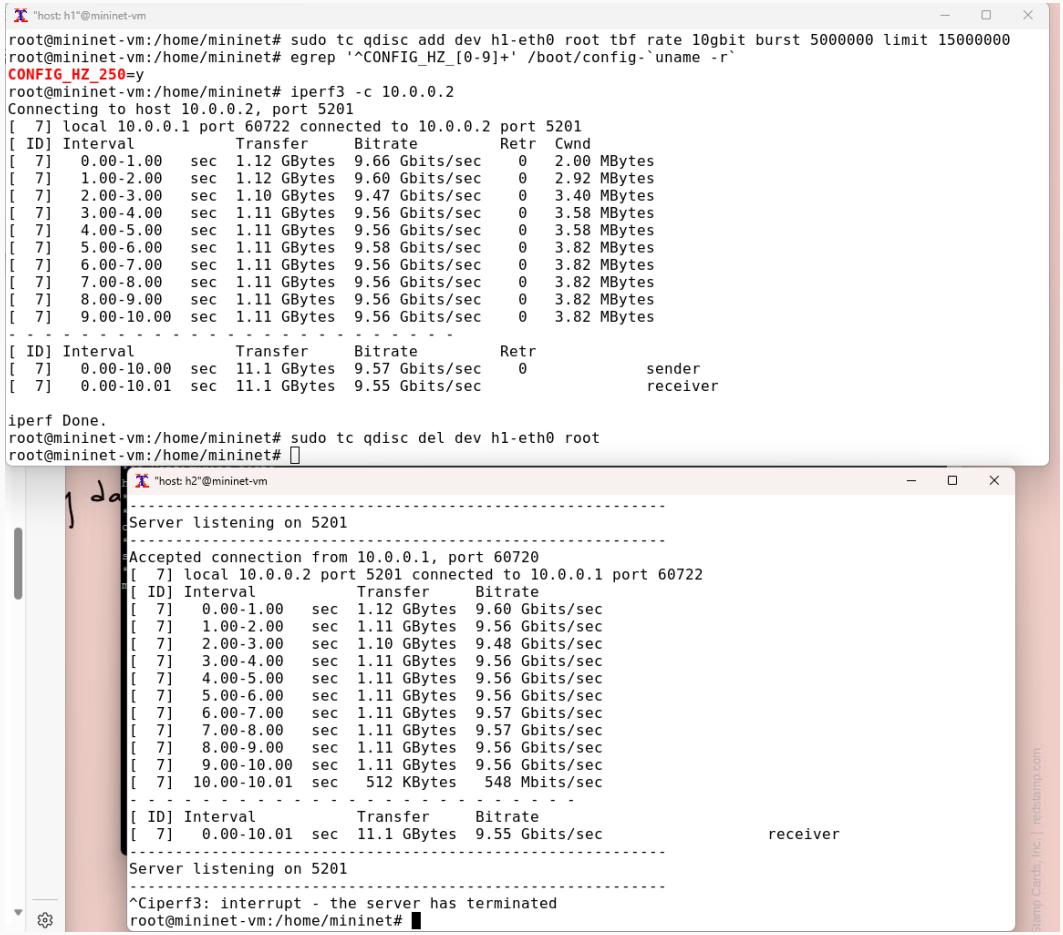


Рис. 3: Настройка tbf на конечных хостах и проверки

# Ограничение скорости на коммутаторах

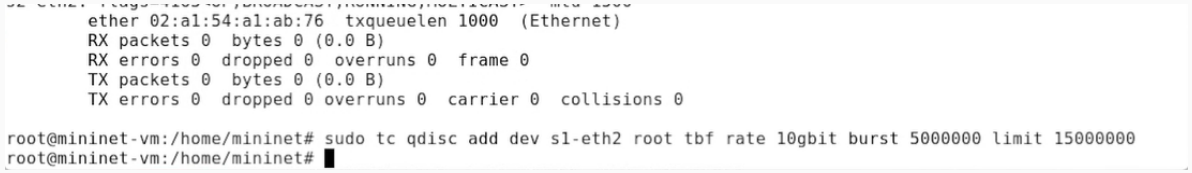


Рис. 4: Настройка tbf на коммутаторах

# Ограничение скорости на коммутаторах

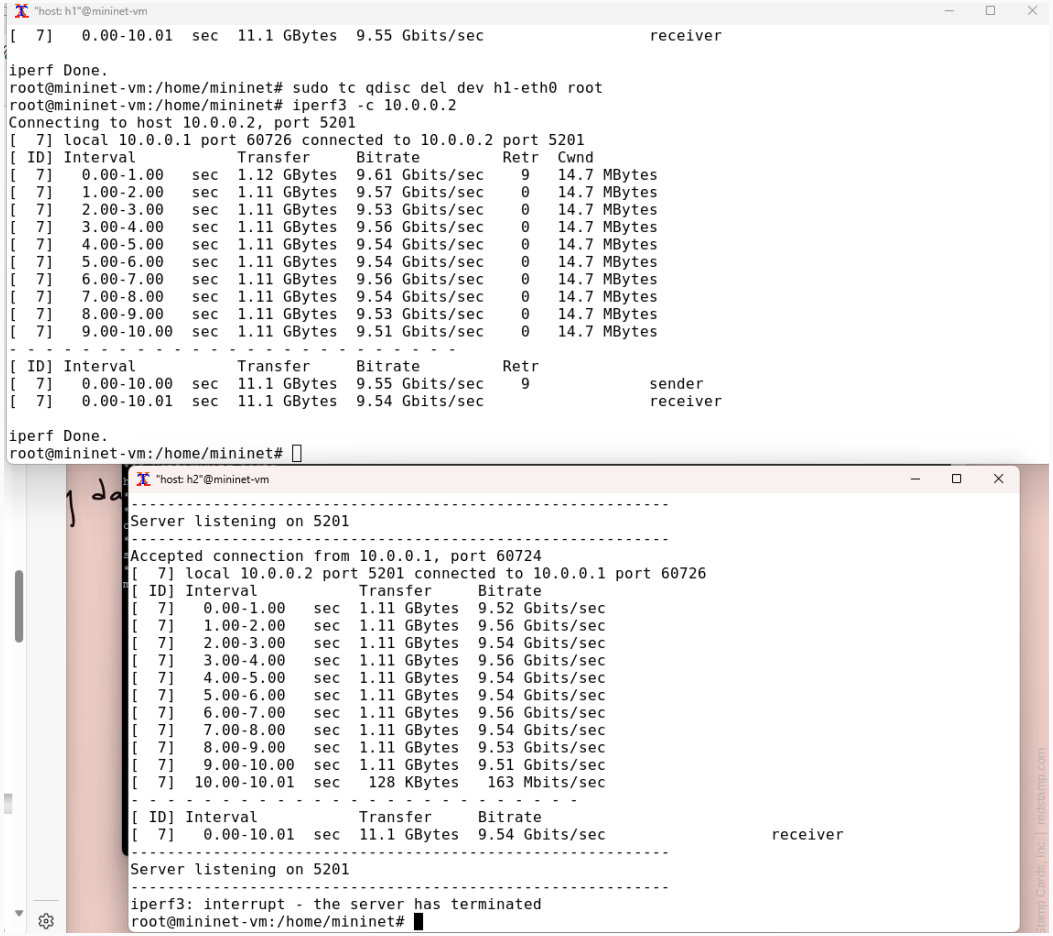


Рис. 5: Запуск iperf3 для проверки

# Объединение NETEM и TBF

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

Рис. 6: Добавление второе правило на коммутаторе s1

# Объединение NETEM и TBF

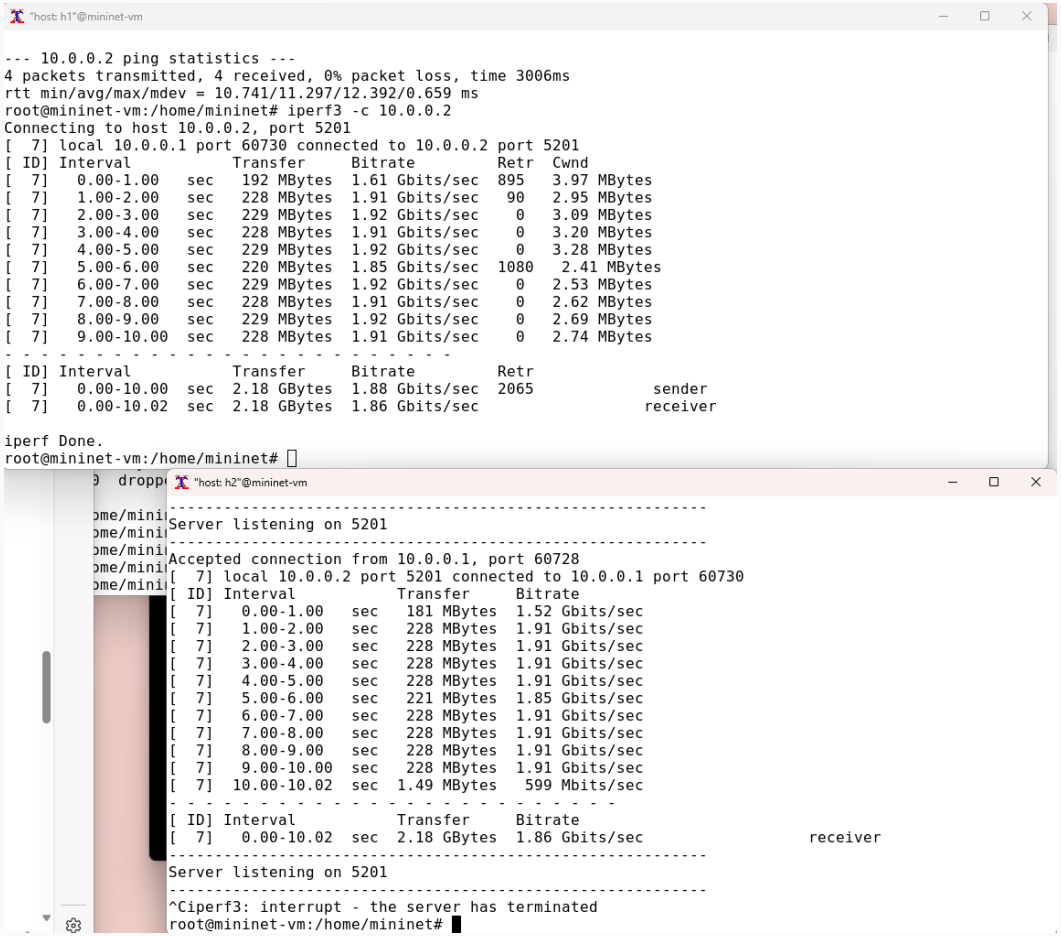


Рис. 7: Запуск iperf3 для проверки

# Воспроизводимые эксперименты

```
  GNU nano 4.8                                    /home/mininet/work/lab_tbf/lab_tbf.py
#!/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.link import TCLink
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( h2, s2 )
        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' )
        s1.cmdPrint( 'tc qdisc add dev s1-eth2 parent 1: handle 2: tbf rate 2gbit burst 1000000 limit 2000000' )

        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. 8: Скрипт lab_tbf.py

# Воспроизводимые эксперименты

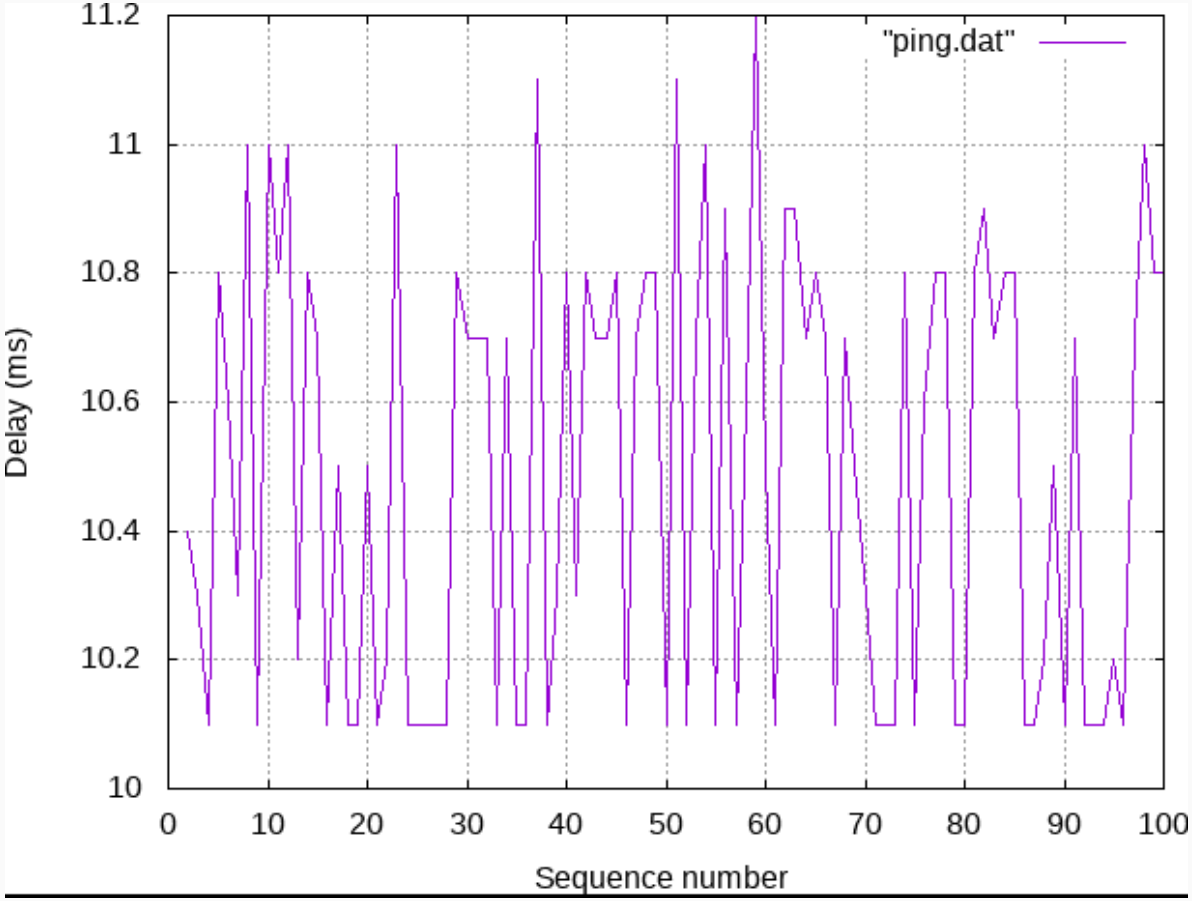


Рис. 9: График

# Выводы по проделанной работе

# Вывод

Я познакомилась 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.