Лабораторная работа №6

Настройка пропускной способности глобальной сети с помощью Token Bucket Filter

Тазаева Анастасия Анатольевна

Содержание

# 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 (рис. 6.3), состоящую из двух хостов и двух коммутаторов с назначенной по умолчанию mininet сетью 10.0.0.0/8.

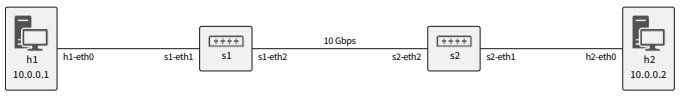


Рис. 1: Топология моделируемой сети с Token Bucket Filter

1. Проведите интерактивные эксперименты по ограничению пропускной способности сети с помощью TBF в эмулируемой глобальной сети.
2. Самостоятельно реализуйте воспроизводимые эксперимент по применению TBF для ограничения пропускной способности. Постройте соответствующие графики.

# 3 Выполнение лабораторной работы

## 3.1 Запуск лабораторной топологии

1. Запустила виртуальную среду с mininet. Из основной ОС подключилась к виртуальной машине (рис. 2). Исправила права запуска X-соединения (рис. 2)

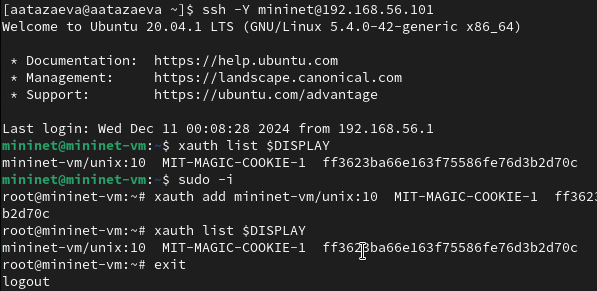


Рис. 2: Подключение к mininet. Исправление прав запуска X-соединения

1. Создала простейшую тополонию, состоящую из двух хостов и двух коммутаторов с назначенной по умолчанию mininet сетью 10.0.0.0/8.

sudo mn --topo=linear,2 -x

1. На хостах h1 и h2, а также на коммутаторе ввела команду ifconfig (рис. 3 - 5), чтобы отобразить информацию, относящуюся к их сетевым интерфейсам и назначенным им IP-адресам. В дальнейшем при работе с NETEM и командой tc будут использоваться интерфейсы h1-eth0 и h2-eth0 .

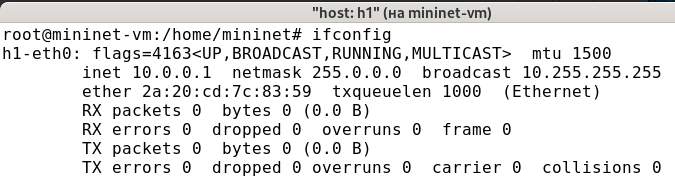


Рис. 3: ifconfig h1

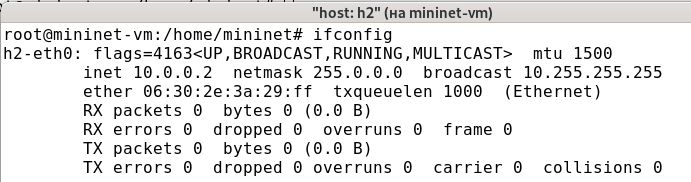


Рис. 4: ifconfig h2

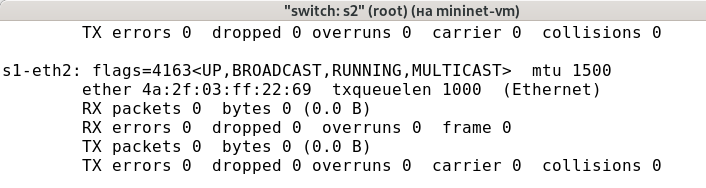


Рис. 5: ifconfig s1

1. Проверила подключение между хостами h1 и h2 с помощью команды ping с параметром -c 4 (рис. 6).

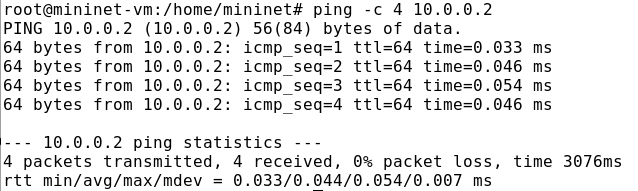


Рис. 6: Проверка подключения ping h1 -> h2

1. В терминале хоста h2 запустила iPerf3 в режиме сервера:

iperf3 -s

1. В терминале хоста h1 запустила iPerf3 в режиме клиента (рис. 7):

iperf3 -c 10.0.0.2

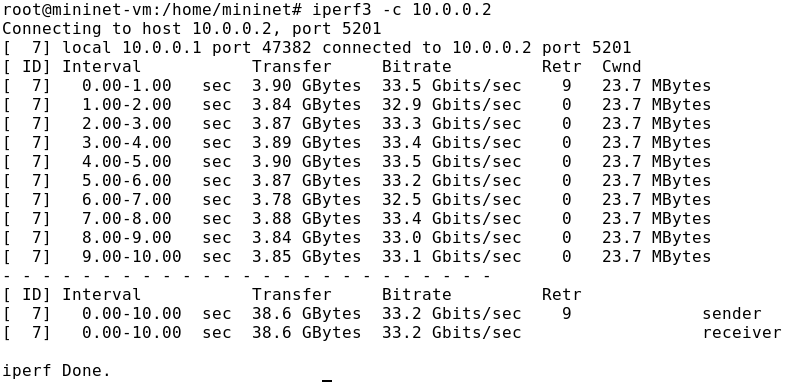


Рис. 7: Результат отработки iPerf3, когда отсутствуют ограничения скорости передачи данных

## 3.2 Интерактивные эксперименты

### 3.2.1 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. Изменила пропускную способность хоста h1 , установив пропускную способность на 10 Гбит/с на интерфейсе h1-eth0 и параметры TBF-фильтра: (рис. 8):

sudo tc qdisc add dev h1-eth0 root tbf rate 10gbit burst 5000000 limit 15000000

Рис. 8: Установка пропускной способности на инт.h1-eth0

Рис. 8: Установка пропускной способности на инт.h1-eth0

Здесь: - sudo : включить выполнение команды с более высокими привилегиями безопасности; - tc : вызвать управление трафиком Linux; - qdisc : изменить дисциплину очередей сетевого планировщика; - add (добавить): создать новое правило; - dev h1-eth0 root : интерфейс, на котором будет применяться правило; - tbf : использовать алгоритм Token Bucket Filter; - rate : указать скорость передачи (10 Гбит/с); - burst : количество байтов, которое может поместиться в корзину (5 000000); - limit : размер очереди в байтах (15 000 000).

1. 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:

egrep '^CONFIG\_HZ\_[0-9]+' /boot/config-`uname -r`

Для расчёта значения всплеска (burst) необходимо скорость передачи (10 Гбит/с или 10 Gbps = 10,000,000,000 bps) разделить на полученное таким образом значение HZ (на хосте h1 HZ = 250): Burst = 10,000,000,000/250 = 40,000,000 bits = 40,000,000/8 bytes = 5,000,000 bytes.

1. С помощью iPerf3 проверила, что значение пропускной способности изменилось (рис. 9)

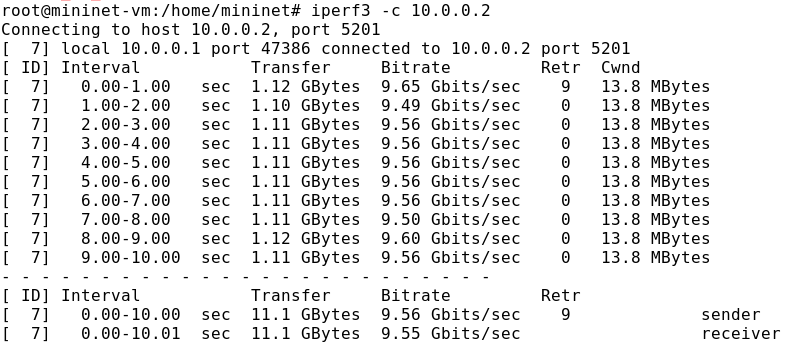


Рис. 9: Проверка значения пропускной способности с помощью iperf3

### 3.2.2 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. Применила 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. 10):

sudo tc qdisc add dev s1-eth2 root tbf rate 10gbit burst 5000000 limit 15000000

Рис. 10: Установка ограничения скорости на коммутаторах

Рис. 10: Установка ограничения скорости на коммутаторах

1. Проверила конфигурацию с помощью инструмента iperf3 для измерения пропускной способности(рис. 11).

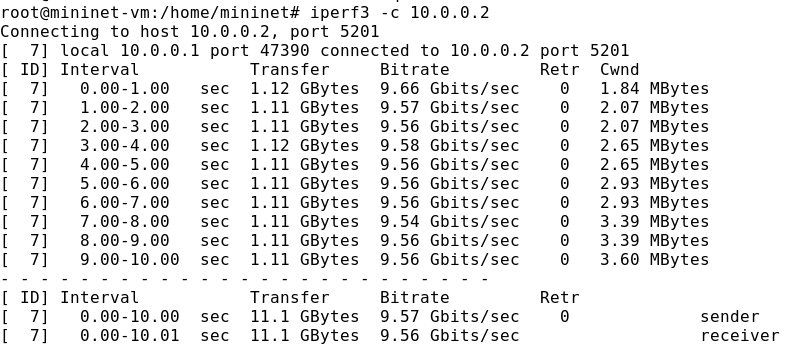


Рис. 11: Результат отработки iPerf3, с ограничением скорости на коммутаторах

1. Удалила модифицированную конфигурацию на коммутаторе s1 (рис. 12).

Рис. 12: Удаление конфигурации на коммутаторе

Рис. 12: Удаление конфигурации на коммутаторе

### 3.2.3 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. Объединила NETEM и TBF, введя на интерфейсе s1-eth2 коммутатора s1 задержку, джиттер, повреждение пакетов и указав скорость (рис. 13):

sudo tc qdisc add dev s1-eth2 root handle 1: netem delay 10ms

Рис. 13: Объединение NETEM и TBF. ч1

Рис. 13: Объединение NETEM и TBF. ч1

Здесь ключевое слово handle задаёт дескриптор подключения, имеющий смысл очерёдности подключения разных дисциплин qdisc .

1. Убедилась, что соединение от хоста h1 к хосту h2 имеет заданную задержку. Для этого запустила команду ping с параметром -c 4 с терминала хоста h1 (рис. 14).

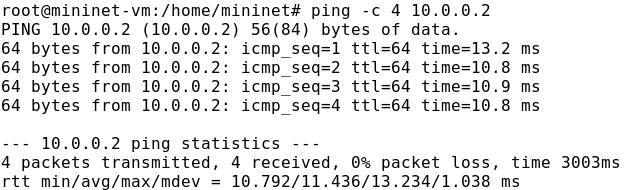


Рис. 14: Проверка задержки между h1 h2

1. Добавила второе правило на коммутаторе s1 , которое задаёт ограничение скорости с помощью tbf с параметрами rate =2gbit, burst =1,000,000, limit =2,000,000 (рис. 15):

sudo tc qdisc add dev s1-eth2 parent 1: handle 2: tbf rate 2gbit burst 1000000 limit 2000000

Рис. 15: Объединение NETEM и TBF. ч2

Рис. 15: Объединение NETEM и TBF. ч2

1. Проверила конфигурацию с помощью инструмента iperf3 для измерения пропускной способности (рис. 16).

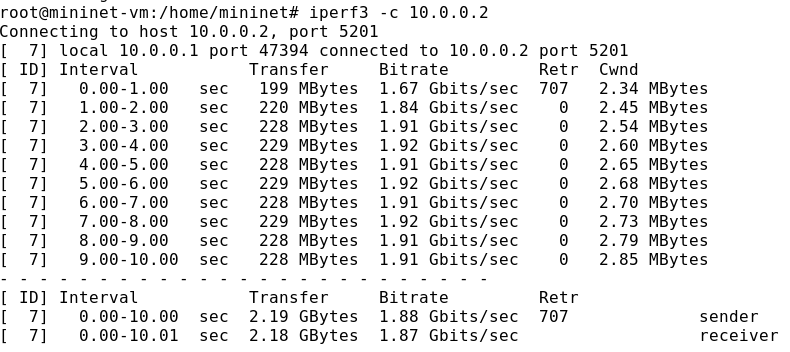


Рис. 16: Результат отработки iPerf3, с ограничением скорости на коммутаторах, введении задержки (netem+tbf)

## 3.3 Воспроизведение экспериментов

1. В виртуальной среде mininet в своём рабочем каталоге с проектами создала каталог tbf и перешла в него. Создала скрипт для эксперимента samost.py (рис. 17).

#!/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' )  
 net.start()  
  
 info('\*\*\*set loss\n' )  
 h1.cmdPrint( 'tc qdisc add dev h1-eth0 root tbf rate 10gbit burst 5000000 limit 15000000' )  
   
 time.sleep(10)  
  
 info('\*\*\*proverka propysknoi sposobnosti')  
 h2.cmdPrint('iperf3 -s &')  
 time.sleep(10)  
 h1.cmdPrint('iperf3 -c', h2.IP(), ' | grep "MBytes" | awk \'{print $7}\' > ping.dat')  
  
 info('\*\*\*stopping network' )  
 net.stop()  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 setLogLevel( 'info' )  
 emptyNet()

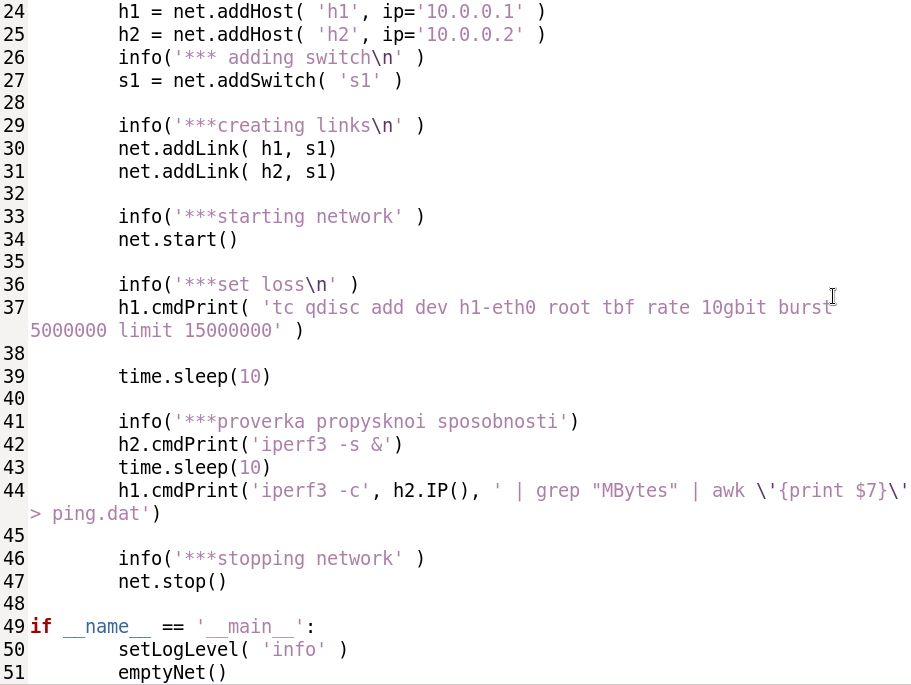


Рис. 17: samost.py

1. Создала Makefile для управления процессом проведения эксперимента (рис. 18).

all: ping.dat ping.png  
  
ping.dat:  
 sudo python samost.py  
 sudo chown mininet:mininet ping.dat  
  
ping.png: ping.dat  
 ./ping\_plot  
  
clean:  
 -rm -f \*.dat \*.png  
  
samost:   
 sudo python samost.py

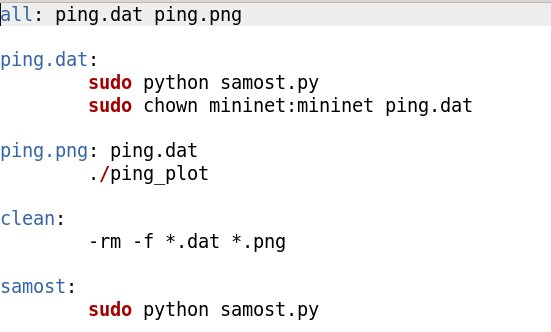


Рис. 18: Makefile

1. В файл ping.dat вывелась информация о скорости (bitrate) (рис. 19).



Рис. 19: ping.dat

1. По результатам ping.dat создала график (рис. 20).

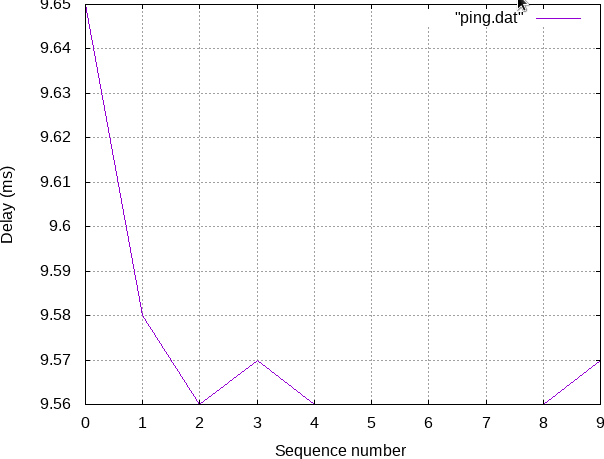


Рис. 20: ping.plot

# 4 Выводы

В ходе лабораторной работы я знакомилась 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.