

## Лабораторная работа № 6. Настройка пропускной способности глобальной сети с помощью Token Bucket Filter

### 6.1. 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.

### 6.2. Предварительные сведения

*Token Bucket Filter (TBF)* представляет собой алгоритм, используемый в сетях с коммутацией пакетов для ограничения пропускной способности и пиковой нагрузки трафика (рис. 6.1). Передача поступающих в очередь системы (queue) пакетов данных осуществляется при наличии в специальном буфере (bucket) необходимого числа разрешений на передачу (или токенов). Токены могут быть представлены в виде пакетов или числа байтов, поступающих в буфер (корзину) фиксированного размера с фиксированной скоростью.

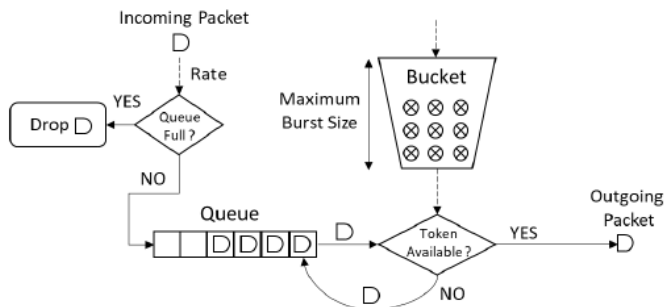


Рис. 6.1. Token Bucket Filter

Максимальная средняя скорость отправки потока данных из очереди системы зависит от скорости прибытия в специализированный буфер разрешений на передачу  $N$  единиц данных. Очередной пакет может быть отправлен только при получении числа разрешений, достаточного для передачи данных, объём которых больше или равен размеру пакета. Если разрешений на передачу достаточно, то необходимое число токенов удаляется из специализированного

буфера, а пакет данных отправляется. Если пакет поступит в очередь системы и не будет располагать необходимым количеством разрешений, то токены не удаляются из специализированного буфера, а сам пакет данных может быть отброшен, поставлен в очередь или передан, но помечен как несоответствующий условиям передачи.

Дисциплина TBF реализована в виде буфера (queue), постоянно заполняющегося токенами с заданной скоростью. Наиболее важным параметром буфера является его размер, определяющий количество хранимых токенов (рис. 6.2).

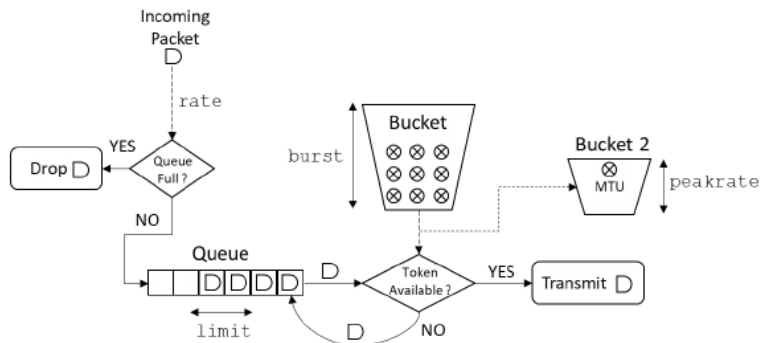


Рис. 6.2. Параметры и архитектура Token Bucket Filter

Если данные прибывают со скоростью, равной скорости входящих токенов, то каждый пакет имеет соответствующий токен и проходит очередь без задержки. Если данные прибывают со скоростью, меньшей скорости поступления токенов, то лишь часть существующих токенов будет уничтожаться, поэтому они станут накапливаться до размера специализированного буфера. Далее накопленные токены могут использоваться при всплесках, для передачи данных со скоростью, превышающей скорость пребывающих токенов.

Если данные прибывают быстрее, чем токены, то в буфере со временем не останется токенов, что заставит дисциплину приостановить передачу данных. Эта ситуация называется «превышением». Если пакеты продолжают поступать, то токены начинают уничтожаться. Данная ситуация позволяет административно ограничивать доступную полосу пропускания.

Накопленные токены позволяют пропускать короткие всплески, но при продолжительном превышении пакеты будут задерживаться, а в крайнем случае — уничтожаться.

Алгоритм TBF обладает свойством взрывоопасности (burstiness), когда корзина полностью заполнена (т. е. никакие пакеты не потребляют токены) и новые пакеты будут потреблять токены сразу, без ограничений. Всплеск определяется как количество токенов, которые могут поместиться в корзину, или как размер (ёмкость) корзины. Для обеспечения ограничения и контроля над всплесками при поступлении пакетов формируется ещё одна корзина, с размером, равным

максимально передаваемому элементу данных (Maximum Transmission Unit, MTU). Скорость обработки этого буфера намного превышает исходную (peak rate — пиковая скорость).

Основной синтаксис `tbfb`, используемый с `tc` в Linux:

```
1 tc qdisc [add | ...] dev [dev_id] root tbf limit [BYTES]
   ↪ burst [BYTES] rate [BPS] [mtu BYTES] [ peakrate BPS ] [
   ↪ latency TIME ]
```

Здесь:

- `tc`: инструмент управления трафиком Linux;
- `qdisc`: дисциплина очереди (`qdisc`), представляющая собой набор правил, определяющих порядок, в котором обслуживаются пакеты, поступающие с выходных данных IP-протокола;
- `[add | del | replace | change | show]`: операция над `qdisc`;
- `dev [dev_id]`: задаёт интерфейс;
- `tbf`: указывает, что используется алгоритм Token Bucket Filter;
- `limit [BYTES]`: размер очереди пакетов в байтах;
- `burst [BYTES]`: количество байтов, которое может поместиться в корзину;
- `rate [BPS]`: скорость передачи данных, определяемая частотой, с которой токены добавляются в корзину;
- `mtu [BYTES]`: максимальная единица передачи в байтах;
- `peakrate [BPS]` (пиковая скорость [бит/с]): максимальная скорость передачи пакета;
- `latency [TIME]`: максимальное время ожидания пакета в очереди.

### 6.3. 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.

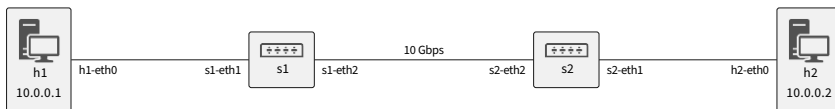


Рис. 6.3. Топология модулируемой сети с Token Bucket Filter

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.

## 6.4. Последовательность выполнения работы

### 6.4.1. Запуск лабораторной топологии

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

```
1  ssh -Y mininet@192.168.x.y
```

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:

```
1  mininet@mininet-vm:~$ xauth list $DISPLAY
2  mininet-vm/unix:10  MIT-MAGIC-COOKIE-1
   ↪  295acad8e35d17636924c5ab80e8462d
3
4  mininet@mininet-vm:~$ sudo -i
5  root@mininet-vm:~# xauth add mininet-vm/unix:10
   ↪  MIT-MAGIC-COOKIE-1  295acad8e35d17636924c5ab80e8462d
6  root@mininet-vm:~# logout
```

После выполнения этих действий графические приложения должны запускаться под пользователем mininet.

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

```
1  sudo mn --topo=linear,2 -x
```

После введения этой команды запустятся терминалы двух хостов, двух коммутаторов и контроллера.

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. В дальнейшем при работе с NETEM и командой `tc` будут использоваться интерфейсы h1-eth0, h2-eth0, s1-eth2.
6. Проверьте подключение между хостами h1 и h2 с помощью команды `ping` с параметром `-c 4`.
7. В терминале хоста h2 запустите `iPerf3` в режиме сервера:

```
1  iperf3 -s
```

8. В терминале хоста h1 запустите `iPerf3` в режиме клиента:

---

<sup>1</sup>Значение для *MIT-MAGIC-COOKIE* приведено условно.

```
1 iperf3 -c 10.0.0.2
```

9. После завершения работы iPerf3 на хосте h1 остановите iPerf3 на хосте h2, нажав **Ctrl**+**C**. В отчёте зафиксируйте результат отработки iPerf3 на данном этапе проведения эксперимента, когда отсутствуют ограничения скорости передачи данных.

## 6.4.2. Интерактивные эксперименты

### 6.4.2.1. 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. Измените пропускную способность хоста h1, установив пропускную способность на 10 Гбит/с на интерфейсе h1-eth0 и параметры TBF-фильтра:

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

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 (5 000 000);
  - `limit`: размер очереди в байтах (15 000 000).
2. 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:

```
1 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.

3. С помощью iPerf3 проверьте, что значение пропускной способности изменилось:

- В терминале хоста h2 запустите iPerf3 в режиме сервера:

```
1  iperf3 -s
```

- В терминале хоста h2 запустите iPerf3 в режиме клиента:

```
1  iperf3 -c 10.0.0.2
```

- После завершения работы iPerf3 на хосте h1 остановите iPerf3 на хосте h2, нажав [Ctrl] + [C]. В отчёте зафиксируйте результат отработки iPerf3 на данном этапе проведения эксперимента.

4. Удалите модифицированную конфигурацию на хосте h1:

```
1  sudo tc qdisc del dev h1-eth0 root
```

#### 6.4.2.2. 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. 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:

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

2. Проверьте конфигурацию с помощью инструмента iPerf3 для измерения пропускной способности:

- В терминале хоста h2 запустите iPerf3 в режиме сервера:

```
1  iperf3 -s
```

- В терминале хоста h2 запустите iPerf3 в режиме клиента:

```
1  iperf3 -c 10.0.0.2
```

- После завершения работы iPerf3 на хосте h1 остановите iPerf3 на хосте h2, нажав [Ctrl] + [C]. В отчёте зафиксируйте результат отработки iPerf3 на данном этапе проведения эксперимента.

3. Удалите модифицированную конфигурацию на коммутаторе s1:

```
1 sudo tc qdisc del dev s1-eth2 root
```

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

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

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

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`.
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`:

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

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

```
1 iperf3 -s
```

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

```
1 iperf3 -c 10.0.0.2
```

- После завершения работы `iPerf3` на хосте `h1` остановите `iPerf3` на хосте `h2`, нажав `Ctrl` + `C`. В отчёте зафиксируйте результат отработки `iPerf3` на данном этапе проведения эксперимента.
5. Удалите модифицированную конфигурацию на коммутаторе `s1`:

```
1 sudo tc qdisc del dev s1-eth2 root
```

### **6.4.3. Воспроизводимые эксперименты**

Самостоятельно реализуйте воспроизводимые эксперименты по использованию TBF для ограничения пропускной способности. Постройте соответствующие графики.

### **6.5. Содержание отчёта**

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулировка задания работы.
3. Описание результатов выполнения задания:
  - скриншоты (снимки экрана), фиксирующие выполнение работы;
  - подробное описание настроек служб в соответствии с заданием;
  - полные тексты конфигурационных файлов настраиваемых в работе служб;
  - результаты проверки корректности настроек служб в соответствии с заданием (подтверждённые скриншотами).
4. Выводы, согласованные с заданием работы.



Учебное издание

**Королькова Анна Владиславовна  
Кулябов Дмитрий Сергеевич**

## **Моделирование сетей передачи данных**

Редактор *И. Л. Панкратова*  
Технический редактор *Н. А. Ясько*  
Компьютерная вёрстка *А. В. Королькова, Д. С. Кулябов*

Подписано в печать . 2023 г. Формат 60×84/16.  
Бумага офсетная. Печать офсетная. Гарнитура Таймс.  
Усл. печ. л. . Тираж 500 экз. Заказ № .

---

Российский университет дружбы народов  
115419, ГСП-1, г. Москва, ул. Орджоникидзе, д. 3

---

Типография РУДН  
115419, ГСП-1, г. Москва, ул. Орджоникидзе, д. 3, тел. 952-04-41