

## Qos компонентти

QoS складається з таких ключових компонентів:

**Класифікація** — це процес розрізнення одного типу трафіку від іншого на основі ACL, точки коду диференційованих послуг (DSCP), класу обслуговування (CoS) та інших факторів.

**Маркування та мутація** — маркування використовується в трафіку для передачі певної інформації до пристрою, що знаходиться нижче за потоком, у мережі або для передачі інформації від одного інтерфейсу комутатора до іншого. Коли трафік позначено, до нього можна застосувати операції QoS. Це можна зробити безпосередньо за допомогою команди set або за допомогою карти таблиці, яка приймає вхідні значення та перетворює їх безпосередньо на значення на виході.

Формування та контролювання.

**Формування** — це процес встановлення максимальної швидкості трафіку з одночасним регулюванням швидкості трафіку таким чином, щоб вихідні пристрої не піддавалися перевантаженням. Формування в найбільш поширеній формі використовується для обмеження трафіку, що надсилається з фізичного або логічного інтерфейсу. Контроль використовується для встановлення максимальної ставки для класу трафіку. Якщо швидкість перевищена, то конкретна дія виконується, як тільки відбувається подія.

**Постановка в чергу** — Постанова в чергу використовується для запобігання заторів. Трафік надсилається в певні черги для обслуговування та планування на основі розподілу пропускної здатності. Потім трафік планується або надсилається через порт. Пропускна здатність. Розподіл пропускної здатності визначає доступну пропускну здатність для трафіку, який підлягає політикам QoS.

**Довіра** — Довіра дозволяє трафіку проходити через комутатор, а значення DSCP, пріоритет або CoS, що надходять із кінцевих точок, зберігаються за відсутності будь-якої явної конфігурації політики.

## Qos термінологія

Наступні терміни використовуються як взаємозамінні в цьому посібнику з налаштування QoS:

- Вгору (напрямок до перемикача) такий самий, як вхід.
- Вниз за течією (напрямок від перемикача) такий самий, як вихід.

## Qos огляд

Налаштувавши якість обслуговування (QoS), ви можете надати перевагу певним типам трафіку за рахунок інших типів трафіку. Без QoS комутатор пропонує найкраще обслуговування для кожного пакета, незалежно від вмісту або розміру

пакета. Комутатор надсилає пакети без жодних гарантій надійності, меж затримки чи пропускної здатності.

Нижче наведені особливі функції, які надає QoS:

- Низька затримка
- Гарантія пропускної здатності
- Можливості буферизації та видалення дисциплін
- Дорожня поліція
- Дозволяє змінювати атрибут кадру або заголовка пакета
- Родичні послуги
- Модульний інтерфейс командного рядка QoS
- Огляд бездротового QoS
- QoS і IPv6 для бездротового зв'язку
- Підтримувані функції дротового та бездротового доступу
- Підтримувані функції QoS на бездротових цілях
- Правила порту
- Політика радіо
- Політики SSID
- Політика клієнта
- Ієрархічний QoS
- Ієрархічний бездротовий QoS

### **Модульний інтерфейс командного рядка QoS**

З перемикачем функції QoS вмикаються через модульний інтерфейс командного рядка QoS (MQC).

**MQC** — це структура інтерфейсу командного рядка (CLI), яка дозволяє створювати політики трафіку та приєднувати ці політики до інтерфейсів. Політика трафіку містить клас трафіку та одну або декілька функцій QoS. Клас трафіку використовується для класифікації трафіку, а функції QoS у політиці трафіку визначають, як обробляти класифікований трафік. Однією з головних цілей MQC є надання незалежного від платформи інтерфейсу для налаштування QoS на платформах Cisco.

### **Встановлення Qos на Debian**

1. Встановлення необхідних пакетів.

```
root@debian:~# apt-get install iptables iptables-persistent
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
iptables is already the newest version (1.8.9-2).
iptables set to manually installed.
iptables-persistent is already the newest version (1.0.20).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
(arg: 4)
```

## 2. Створення файлу налаштування /etc/network/if-up.d/qos.

```
GNU nano 7.2 /etc/network/if-up.d/qos
IFACE="eth0"

tc qdisc del dev $IFACE root 2>/dev/null

tc qdisc add dev $IFACE root handle 1: htb default 10

tc class add dev $IFACE parent 1: classid 1:1 htb rate 100mbit ceil 100mbit

tc class add dev $IFACE parent 1:1 classid 1:10 htb rate 50mbit ceil 100mbit prio 1
tc filter add dev $IFACE parent 1: protocol ip prio 1 u32 match ip dport 80 0xffff flowid 1:10

tc qdisc add dev $IFACE parent 1:10 handle 10: sfq perturb 10

exit 0
```

## 3. Робимо цей файл виконавчим.

```
root@debian:~# chmod +x /etc/network/if-up.d/qos
root@debian:~#
```

## 4. Перезавантажуємо сервіс.

```
root@debian:~# systemctl restart networking_
```

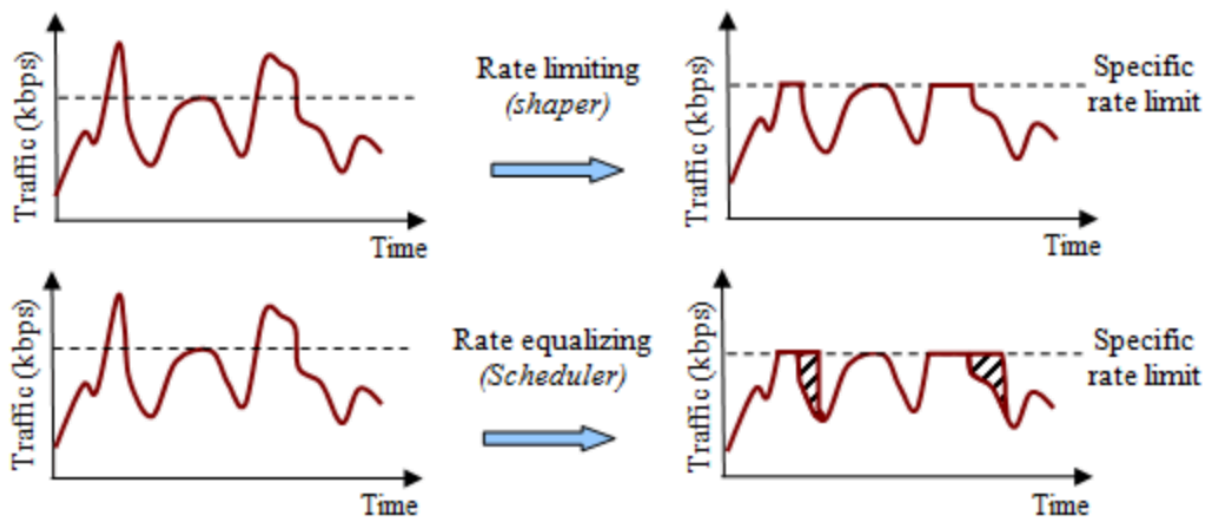
Ці налаштування створюють QoS-політику, яка встановлює високий пріоритет для HTTP-трафіку (порт 80) зі швидкістю 50 Mbit/s та максимальною пропускнуою здатністю 100 Mbit/s. Ви можете змінити ці значення відповідно до своїх потреб.

### Microtic Qos

Обмеження швидкості передачі може бути виконано двома способами:

1. Відкидаються всі пакети, що перевищують ліміт швидкості передачі (Шейпер).

2. Затримка перевищили задане обмеження швидкості передачі пакетів у черзі і відправлення їх пізніше, щойно утворюється така можливість, тобто. вирівнювання швидкості передачі (шедулер).



Як видно на ілюстрації, шейпер ріже все, що не влізло, а шедулер просто пригальмовує. Відповідно саме шедулер нам і потрібен.

Тепер необхідно поділити трафік на класи та задати кожному класу свій пріоритет. Перший клас обслуговується насамперед, останній – в останню.

Найпростіший варіант такого рішення, який найчастіше використовується — просто пустити пріоритетом VoIP-трафік, а решту за залишковим принципом, але я зроблю трохи складніше.

Сьогодні, завдяки загальному шифруванню трафіку, ми не можемо так запросто взяти і за допомогою L7-regexp відловити трафік youtube, наприклад, або Skype. Тому, використовуючи такі скрипти, уважно поставтеся до питання визначення трафіку. Це, на мою думку, єдина складність у цьому питанні.

Тепер розмітимо трафік згідно з планом вище. У коді я використовую `interfacebandwidth`, тобто. ширину каналу. У мене він симетричний і дорівнює 100м. Якщо у вас відрізняється ширина каналу, то необхідно змінити значення `interfacebandwidth` на необхідне. Якщо канал асинхронний, то скрипт буде складніше за потребу окремо маркувати пакети для вхідного та вихідного трафіку.

Це нескладно, але значно збільшить скрипт, погіршивши його читальність і загалом виходить за рамки статті.

В address-list я демонструю можливість масової вставки адрес з FQDN. Зрозуміло, можна просто прописати потрібні IP вручну.

```
-----  
#Set bandwidth of the interface  
:local interfaceBandwidth 100M  
  
# address-lists  
:for i from=1 to=10 do={/ip firewall address-list add list=WoT  
address=("login.p"."$i"."worldoftanks.net")}  
#  
/ip firewall mangle  
# prio_1  
    add chain=prerouting action=mark-packet new-packet-mark=prio_1  
protocol=icmp  
    add chain=prerouting action=mark-packet new-packet-mark=prio_1  
protocol=tcp port=53  
    add chain=prerouting action=mark-packet new-packet-mark=prio_1  
protocol=udp port=53  
    add chain=prerouting action=mark-packet new-packet-mark=prio_1  
protocol=tcp tcp-flags=ack packet-size=0-123  
# prio_2  
    add chain=prerouting action=mark-packet new-packet-mark=prio_2 dscp=40  
    add chain=prerouting action=mark-packet new-packet-mark=prio_2 dscp=46  
    add chain=prerouting action=mark-packet new-packet-mark=prio_2  
protocol=udp port=5060,5061,10000-20000 src-address=192.168.100.110  
    add chain=prerouting action=mark-packet new-packet-mark=prio_2  
protocol=udp port=5060,5061,10000-20000 dst-address=192.168.100.110  
# prio_3  
    add chain=prerouting action=mark-packet new-packet-mark=prio_3  
protocol=tcp port=22  
    add chain=prerouting action=mark-packet new-packet-mark=prio_3 src-  
address-list=WoT  
    add chain=prerouting action=mark-packet new-packet-mark=prio_3 dst-  
address-list=WoT  
# prio_4  
    add chain=prerouting action=mark-packet new-packet-mark=prio_4  
protocol=tcp port=3389  
    add chain=prerouting action=mark-packet new-packet-mark=prio_4  
protocol=tcp port=80,443  
Аккуратно уложим размеченный трафик в очередь:
```

```

<source>queue tree add max-limit=$interfaceBandwidth name=QoS_global
parent=global priority=1
:for indexA from=1 to=4 do={
    /queue tree add \
        name=( "prio_" . "$indexA" ) \
        parent=QoS_global \
        priority=($indexA) \
        queue=ethernet-default \
        packet-mark=("prio_" . $indexA) \
        comment=("Priority " . $indexA . " traffic")
}
/queue tree add name="prio_5" parent=QoS_global priority=5 \
    queue=ethernet-default packet-mark=no-mark comment="Priority 5 traffic"
-----

```

І останнє, якщо Mikrotik підтримує WMM, було б логічно розмітити трафік і для нього. Робиться це тим самим mangle-ом за допомогою команди set\_priority. Згідно з wiki Mikrotik'a, таблиця пріоритетів WMM виглядає досить химерно:

- 1,2 - background
- 0,3 - best effort
- 4,5 — відео
- 6,7 - voice.

Розмітимо пріоритети, використовуючи самі правила, що й маркування пакетів:

```

-----
/ip firewall mangle
# prio_1
    add chain=prerouting action=set-priority new-priority=7 protocol=icmp
    add chain=prerouting action=set-priority new-priority=7 protocol=tcp port=53
    add chain=prerouting action=set-priority new-priority=7 protocol=udp port=53
    add chain=prerouting action=set-priority new-priority=7 protocol=tcp tcp-
flags=ack packet-size=0-123
# prio_2
    add chain=prerouting action=set-priority new-priority=6 dscp=40
    add chain=prerouting action=set-priority new-priority=6 dscp=46
    add chain=prerouting action=set-priority new-priority=6 protocol=udp
port=5060,5061,10000-20000 src-address=192.168.100.110
    add chain=prerouting action=set-priority new-priority=6 protocol=udp
port=5060,5061,10000-20000 dst-address=192.168.100.110
# prio_3
    add chain=prerouting action=set-priority new-priority=5 protocol=tcp port=22
    add chain=prerouting action=mark-packet new-packet-mark=prio_3 src-
address-list=(smth)

```

```
add chain=prerouting action=mark-packet new-packet-mark=prio_3 dst-  
address-list=(smth)  
# prio_4  
add chain=prerouting action=set-priority new-priority=3 protocol=tcp port=3389  
-----
```

### **Висновок**

Налаштування QoS (Quality of Service) на Debian є важливим кроком для керування пропускнуою здатністю мережі та забезпечення пріоритетів для певних типів трафіку. Використання QoS дозволяє покращити продуктивність та ефективність мережі, особливо в умовах великої завантаженості або коли потрібно надавати пріоритет деяким додаткам або послугам.