

The image shows a terminal window with two side-by-side windows displaying network speed test results. The left window is titled 'root@centos7-ip-196:~' and shows a test connected to 10.20.1.23 port 5201. The right window is titled 'root@ubuntu18-ip-195: ~' and shows a test connected to 10.20.1.23 port 5202. Both windows show a table of results with columns for Interval, Transfer, and Bandwidth. In the left window, the bandwidth values are around 84.5 Mbits/sec, and a red box highlights the first few rows. In the right window, the bandwidth values are around 94.9 Mbits/sec, and a red box highlights the first few rows. Below the terminal windows, a browser window shows the Yandex Internet Speed Test page. The page displays the user's IP address, IPv4 and IPv6 addresses, and the browser used (Edge 85.0.564). The speed test results show an incoming connection speed of 42.14 Mbit/s (5.27 MByte/s) and an outgoing connection speed of 29.22 Mbit/s (3.65 MByte/s). The page also has buttons for 'Измерить ещё раз' and 'Поделиться'.

ID	Interval	Transfer	Bandwidth
[4]	0.00-3.00 sec	30.1 MBytes	84.1 Mbits/sec
[4]	3.00-6.00 sec	30.4 MBytes	84.9 Mbits/sec
[4]	6.00-9.00 sec	30.2 MBytes	84.5 Mbits/sec
[4]	9.00-12.00 sec	16.7 MBytes	46.8 Mbits/sec
[4]	12.00-15.00 sec	12.6 MBytes	35.2 Mbits/sec
[4]	15.00-18.00 sec	22.6 MBytes	63.3 Mbits/sec
[4]	18.00-21.00 sec	22.3 MBytes	62.3 Mbits/sec
[4]	21.00-24.00 sec	25.6 MBytes	71.5 Mbits/sec
[4]	24.00-27.00 sec	28.6 MBytes	79.8 Mbits/sec
[4]	27.00-30.00 sec	28.8 MBytes	80.6 Mbits/sec
[4]	30.00-33.00 sec	28.9 MBytes	80.7 Mbits/sec
[4]	33.00-36.00 sec	29.4 MBytes	82.2 Mbits/sec
[4]	36.00-39.00 sec	30.4 MBytes	84.9 Mbits/sec
[4]	39.00-42.00 sec	30.2 MBytes	84.5 Mbits/sec
[4]	42.00-45.00 sec	32.2 MBytes	90.1 Mbits/sec
[4]	45.00-48.00 sec	33.9 MBytes	94.9 Mbits/sec
[4]	48.00-50.00 sec	22.6 MBytes	94.9 Mbits/sec

ID	Interval	Transfer	Bandwidth	Retr	sender	receiver
[4]	0.00-50.00 sec	456 MBytes	76.5 Mbits/sec	7230		
[4]	0.00-50.00 sec	456 MBytes	76.4 Mbits/sec		sender	receiver

ID	Interval	Transfer	Bandwidth	Retr	sender	receiver
[4]	0.00-3.00 sec	35.1 MBytes	98.2 Mbits/sec			
[4]	3.00-6.00 sec	33.9 MBytes	94.9 Mbits/sec			
[4]	6.00-9.00 sec	12.6 MBytes	35.2 Mbits/sec			
[4]	9.00-12.00 sec	3.58 MBytes	10.0 Mbits/sec			
[4]	12.00-15.00 sec	3.71 MBytes	10.4 Mbits/sec			
[4]	15.00-18.00 sec	4.05 MBytes	11.3 Mbits/sec			
[4]	18.00-21.00 sec	4.08 MBytes	11.4 Mbits/sec			
[4]	21.00-24.00 sec	4.26 MBytes	11.9 Mbits/sec			
[4]	24.00-27.00 sec	4.70 MBytes	13.1 Mbits/sec			
[4]	27.00-30.00 sec	4.94 MBytes	13.8 Mbits/sec			
[4]	30.00-33.00 sec	5.24 MBytes	14.6 Mbits/sec			
[4]	33.00-36.00 sec	5.02 MBytes	14.0 Mbits/sec			
[4]	36.00-39.00 sec	5.08 MBytes	14.2 Mbits/sec			
[4]	39.00-42.00 sec	4.66 MBytes	13.0 Mbits/sec			
[4]	42.00-45.00 sec	3.67 MBytes	10.3 Mbits/sec			
[4]	45.00-48.00 sec	3.58 MBytes	10.0 Mbits/sec			
[4]	48.00-50.00 sec	2.48 MBytes	10.4 Mbits/sec			

ID	Interval	Transfer	Bandwidth	Retr	sender	receiver
[4]	0.00-50.00 sec	141 MBytes	23.7 Mbits/sec	2093		
[4]	0.00-50.00 sec	141 MBytes	23.7 Mbits/sec		sender	receiver

<https://serveradmin.ru/audio/mikrotik-nastrojka-qos.ogg>

Хочу сегодня рассмотреть полезный и востребованный функционал популярных роутеров известной латвийской компании. Я расскажу, как настроить ограничение трафика и приоритизацию, используя функции QOS в виде очередей (queues) в Mikrotik. Как обычно все покажу на конкретных примерах, чтобы смогли без проблем их повторить и подстроить под себя.

Если у вас есть желание научиться работать с роутерами микротик и стать специалистом в этой области, рекомендую по программе, основанной на информации из официального курса **MikroTik Certified Network Associate**. Курс стоящий, все подробности читайте по ссылке.



Содержание

Введение

Возможности QOS в Микротике

Типы очередей (queue type) в Mikrotik

Simple queue vs Queue Tree

Настройка Simple queue в Mikrotik

Режим Burst

Ограничение скорости по IP

Приоритет трафика по IP

Приоритет HTTP трафика для максимально быстрого серфинга

Приоритет SIP трафика для VOIP

Ограничение скорости на интерфейсе

Заключение

Помогла статья? Подписывайся на telegram канал автора

Данная статья является частью единого цикла статьей про Mikrotik.

Введение

Для начала поясню, о чем вообще пойдет речь. **Quality of Service (QoS)** - технология представления различным классам трафика различных приоритетов в обслуживании. Условная классификация трафика может проводиться по разным признакам:

- IP адрес
- MAC адрес
- Протокол
- TCP порт
- Маркировка и т.д.

Чаще всего под QOS подразумевают настройку приоритезации трафика (задержка при передачи пакета) и ограничение максимальной скорости канала. Как следствие этих возможностей, можно настраивать временное превышение заданного ограничения, гарантию заданной полосы пропускания, равномерное распределение ширины канала между пользователями и т.д. Реже управляют потерями пакетов, но лично я с этим не сталкивался и не настраивал.

Приведу простой и наглядный пример использования QOS. С его помощью вы можете настроить максимальный приоритет для трафика по протоколу SIP. Далее разделить поровну между пользователями пропускную способность интернет канала. Так же сильно ограничить в скорости соединения пользователей, качающих файлы через торренты.

Сейчас существует распространенное заблуждение, что так как скорости интернета стали очень высокими, а трафик в основном безлимитный, QOS стал не нужен. Очевидно, что это не так, потому что с ростом скоростей выросло и потребление трафика в виде просмотра потокового видео и загрузки файлов через торренты.

Далее расскажу, как настроить все эти ограничения и приоритеты в Mikrotik. За основу возьму информацию из официальной wiki по теме Queue. А так же данные, которые получил на обучении по Микротик, которое я проходил.

В интернете есть много информации по теме настройки микротиков и в частности qos на этих устройствах. Я сам много читал различных статей. В своем материале я не претендую на истину и best practice. Я просто постараюсь взять наиболее важные моменты и описать их как можно подробнее и максимально понятно, чтобы вы смогли повторить. Я хоть и не плохо разбираюсь в микротиках, но изучать данную тему на основе доступных материалов мне было не просто. Потратил много времени и сил, поэтому собрав свой опыт, хочу по возможности упростить вам эту задачу.

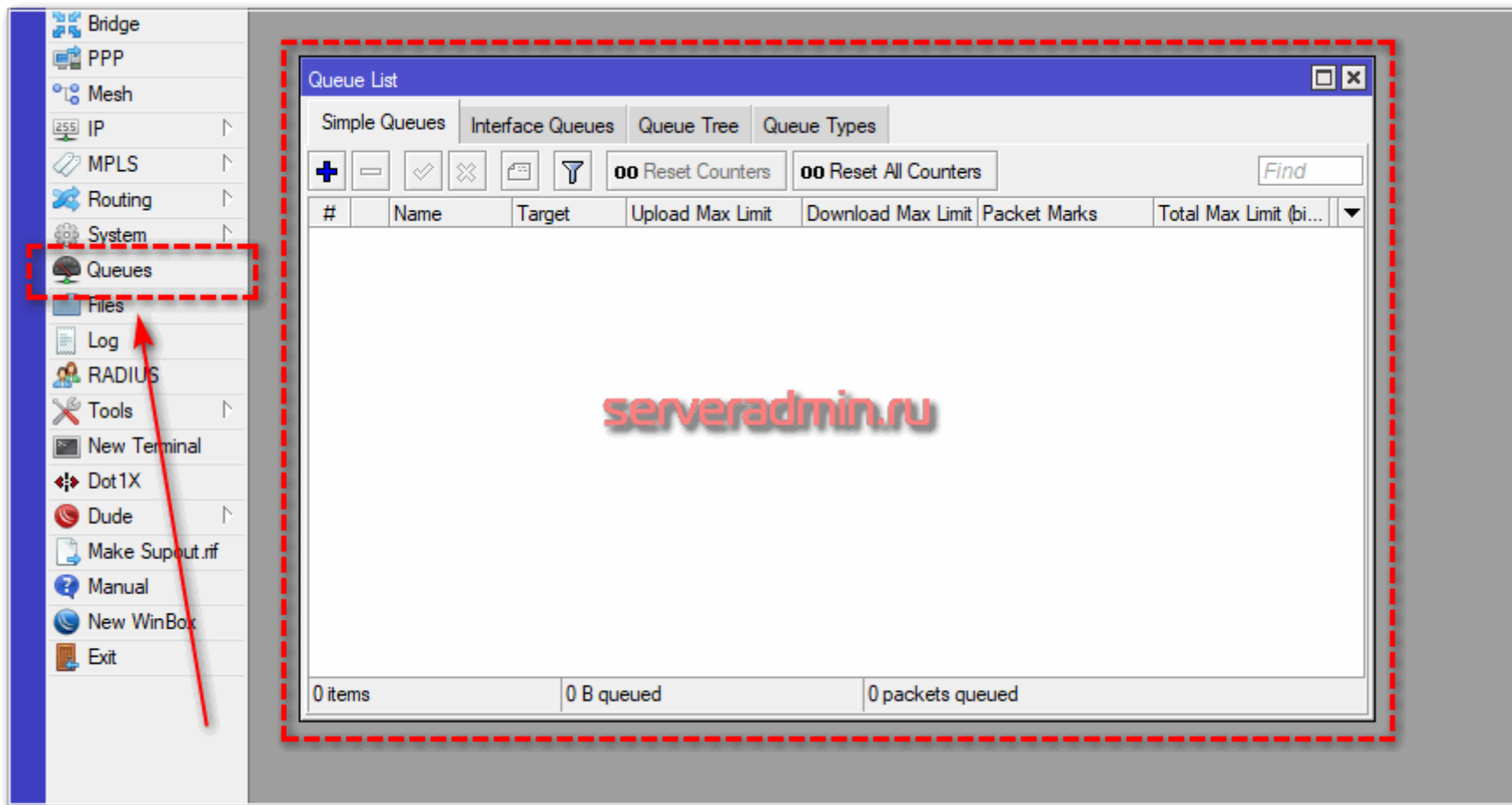
Если вы новичок в микротиках, или еще не сделали базовую настройку, рекомендую мою статью на эту тему - настройка mikrotik.

Возможности QOS в Микротике

Для начала кратко рассмотрим, какие основные возможности QOS есть в Микротике.

- Ограничение скорости и приоритезация на основании ip, mac, порта и интерфейса (например wifi), подсети, протокола и др.
- Возможности кратковременного увеличения скорости.
- Настройка различных лимитов по времени суток.
- Распределение ширины канала между пользователями равномерно или по каким-то правилам.
- И многое другое.

Вся настройка qos производится в отдельном разделе интерфейса управления под названием **Queues** (очереди).



В Микротике существуют несколько типов очередей, о которых мы поговорим далее.

Типы очередей (queue type) в Mikrotik

Операционная система в Mikrotik - RouterOS поддерживает следующие типы очередей:

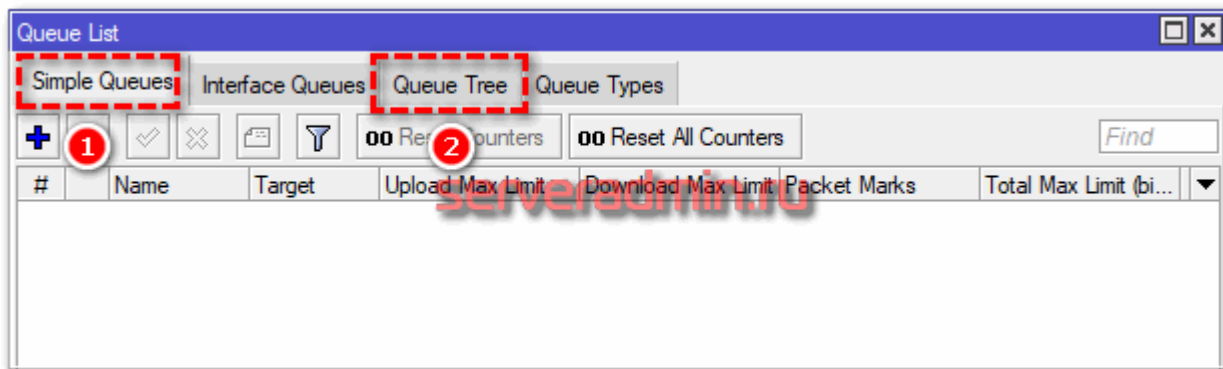
1. **FIFO** - BFIFO, PFIFO, MQ PFIFO. Алгоритм FIFO (first in — first out, первый пришёл - первый ушёл) подразумевает, что данные, попадающие в очередь первыми, будут первыми переданы на дальнейшую обработку. Тот, кто пришел позже, ждет окончания обработки поступивших ранее. Подобный алгоритм применяется для ethernet интерфейсов.
2. **RED** (random early drop) - случайное превентивное отбрасывание. Управление потоком данных осуществляется на основе заданных порогов. При выходе за эти пороги, пакеты могут отбрасываться. Данный алгоритм способен выравнить пропускную способность и сглаживать резкие скачки нагрузки.
3. **SFQ** (Stochastic fairness queuing) - стохастическая честная очередь. Условно можно считать этот алгоритм наиболее честным в том плане, что всем входящим подключениям предоставляется равная возможность по передаче данных. Пакеты распределяются в одну из очередей и обрабатываются по алгоритму round-robin.
4. **PCQ** (Per Connection Queuing) - распределение очередей по соединениям. Это частный случай предыдущего алгоритма sfq. В нем добавляется возможность задавать различные условия для очередей, в которые попадают пакеты. Этот алгоритм является основным, когда надо поровну разделить канал между клиентами. Так же с его помощью можно организовывать динамический шейпинг.

На практике, алгоритмы RED и SFQ используются редко. Для настройки приоритизации трафика и ограничения скорости канала используют очередь PCQ. У нее много параметров для гибкого конфигурирования, примеры которого я покажу далее.

Simple queue vs Queue Tree

В Mikrotik присутствуют 2 типа очередей:

- Simple Queues - простой тип очередей.
- Queue Tree - очереди из иерархических деревьев с правилами.



Отличий у этих очередей много. Показываю наиболее значимые в виде таблицы. Автор этой таблицы - Дмитрий Скоромнов. По крайней мере я ее увидел у него.

Разница между Simple queue и Queue Tree

Опция	Simple Queue	Queue Tree
Порядок правил	играет роль	не играет роль
Маркировка трафика	возможна	обязательная
Время действия правила	возможно	не возможно
График загрузки	есть	нет

В целом, настройка Simple Queues более простая и очевидная, в то время как Queue Tree позволяют строить более сложные конфигурации QOS с обязательным использованием маркировки в Mangle. В случае же с Simple Queues в простых ситуациях достаточно будет информации об адресах источника и получателя. Маркировать ничего не придется.

Что же выбрать для настройки QOS - Simple queue или Queue Tree? Я советую начать с простых очередей. Если ваша задача не решается с их помощью, переходите на иерархические деревья правил с маркировкой. И еще нужно понимать, что Simple queue это частный случай Queue Tree, поэтому при использовании обоих типов очередей нужно внимательно смотреть на то, чтобы правила не пересекались в разных очередях.

Настройка Simple queue в Mikrotik

Давайте потихоньку переходить к практике. Но перед этим необходимо проверить важную деталь. Для того, чтобы в Mikrotik работали очереди, необходимо обязательно отключить fasttrack. Для начала настроим равномерное распределение трафика между клиентами. Их у меня будет два:

- 192.168.88.195 - ubuntu18.
- 192.168.88.196 - centos7.
- 10.20.1.23 - сервер в "интернете", к которому будем подключаться и проверять скорость соединения.

Адрес Mikrotika - 192.168.88.1. Измерять скорость буду с помощью программы **iperf**. Канал в интернет - 100 Мбит/с. Для начала посмотрим, как будет распределяться канал между клиентами без настройки qos. Для этого просто запускаю iperf на обоих клиентах с разницей в 10 секунд.

```

root@centos7:~# iperf3 -c 10.20.1.23 -p 5201 -i 5 -t 30
Connecting to host 10.20.1.23, port 5201
[ 4] local 192.168.88.199 port 37190 connected to 10.20.1.23 port 5201
[ ID] Interval      Transfer    Bandwidth  Retr  Cwnd
[ 4]  0.00-5.00    sec  15.4 MBytes  25.9 Mbits/sec  0   70.7 KBytes
[ 4]  5.00-10.00   sec  16.4 MBytes  27.5 Mbits/sec  0   73.5 KBytes
[ 4] 10.00-15.00   sec  15.7 MBytes  26.3 Mbits/sec  0   73.5 KBytes
[ 4] 15.00-20.00   sec  22.7 MBytes  38.1 Mbits/sec  0   188 KBytes
[ 4] 20.00-25.00   sec  57.7 MBytes  96.7 Mbits/sec  0   269 KBytes
[ 4] 25.00-30.00   sec  57.6 MBytes  96.6 Mbits/sec  0   269 KBytes
-----
[ ID] Interval      Transfer    Bandwidth  Retr
[ 4]  0.00-30.00   sec  185 MBytes  51.9 Mbits/sec  0      sender
[ 4]  0.00-30.00   sec  184 MBytes  51.5 Mbits/sec                receiver

iperf Done.
[root@centos7 ~]#

root@ubuntu18:~# iperf3 -c 10.20.1.23 -p 5202 -t 30 -i 5
Connecting to host 10.20.1.23, port 5202
[ 4] local 192.168.88.198 port 50180 connected to 10.20.1.23 port 5202
[ ID] Interval      Transfer    Bandwidth  Retr  Cwnd
[ 4]  0.00-5.00    sec  58.5 MBytes  98.2 Mbits/sec  0   188 KBytes
[ 4]  5.00-10.00   sec  57.4 MBytes  96.3 Mbits/sec  0   197 KBytes
[ 4] 10.00-15.00   sec  44.3 MBytes  74.3 Mbits/sec  0   197 KBytes
[ 4] 15.00-20.00   sec  41.3 MBytes  69.2 Mbits/sec  0   197 KBytes
[ 4] 20.00-25.00   sec  41.4 MBytes  69.4 Mbits/sec  0   197 KBytes
[ 4] 25.00-30.00   sec  41.1 MBytes  69.0 Mbits/sec  0   308 KBytes
-----
[ ID] Interval      Transfer    Bandwidth  Retr
[ 4]  0.00-30.00   sec  284 MBytes  79.4 Mbits/sec  0      sender
[ 4]  0.00-30.00   sec  282 MBytes  78.9 Mbits/sec                receiver

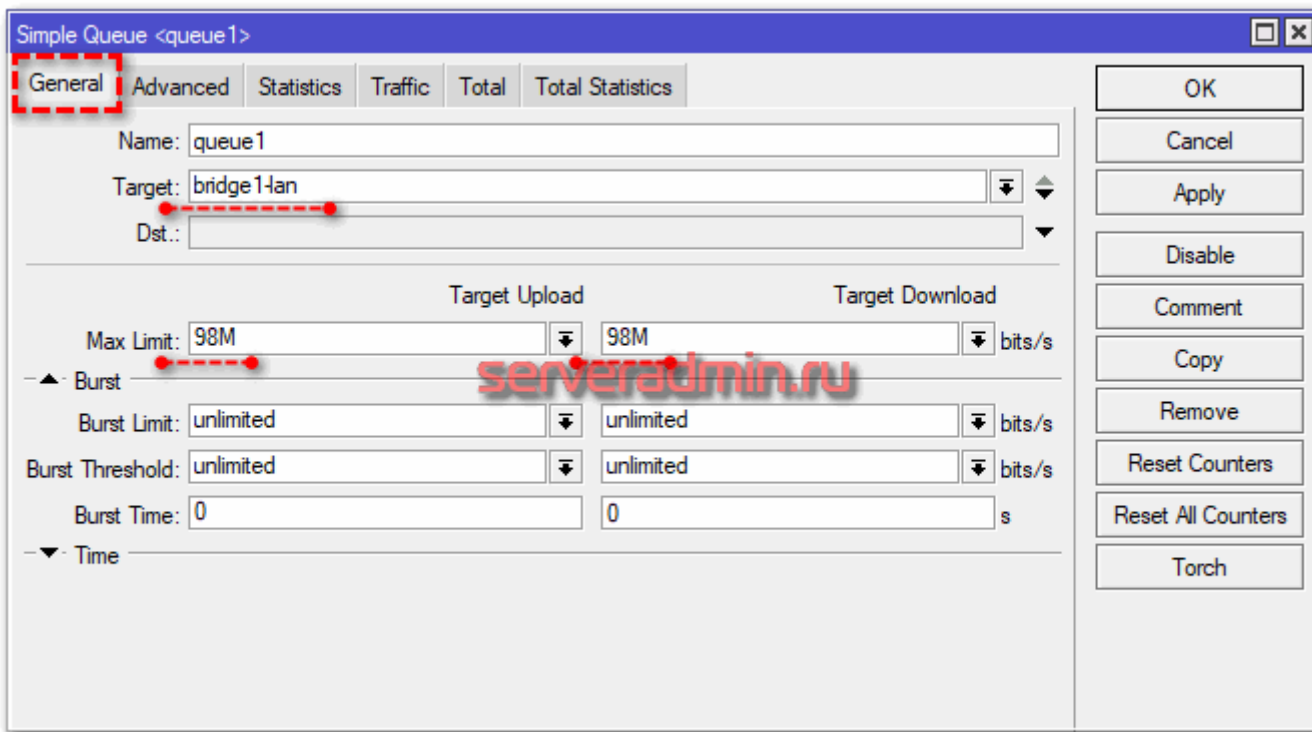
iperf Done.
root@ubuntu18:~#

```

Я сначала запустил проверку скорости на ubuntu18, а через 10 секунд на centos7. Видно, что ubuntu сначала качала на полной скорости 100 Мбит/с, а потом, при запуске параллельного теста на centos, снизила скорость на треть. По факту получилось, что кто раньше запустил тест, тот и забрал себе больше канала. Средняя скорость загрузки разных клиентов отличается - 51.9 Mb/s у того, кто начал позже, против 79.4 Mb/s. Если бы тут работал

какой-нибудь торрент клиент, он бы забрал большую часть канала себе.

А теперь сделаем так, чтобы скорость между клиентами распределялась равномерно. Для этого идем в очереди и создаем **simple queue**.



Simple Queue <queue1>

General Advanced Statistics Traffic Total Total Statistics

Name: queue 1

Target: bridge 14an

Dst.:

Target Upload Target Download

Max Limit: 98M 98M bits/s

Burst Limit: unlimited unlimited bits/s

Burst Threshold: unlimited unlimited bits/s

Burst Time: 0 0 s

Time

OK

Cancel

Apply

Disable

Comment

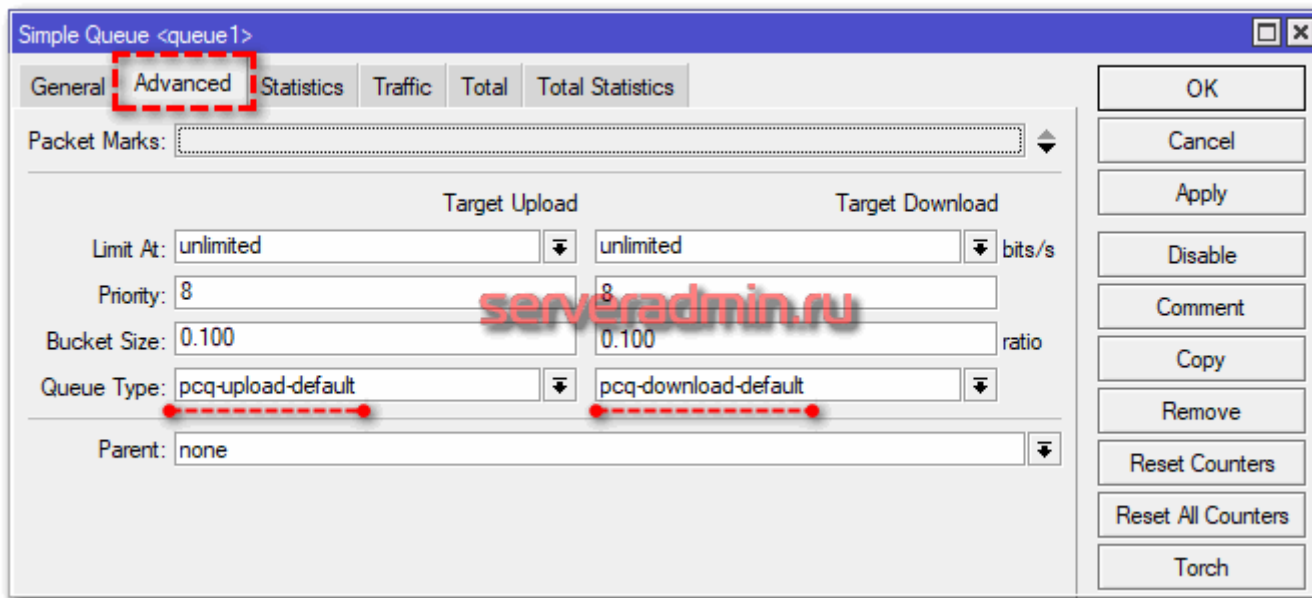
Copy

Remove

Reset Counters

Reset All Counters

Torch



Я сделал минимально необходимые настройки. Все остальное оставил в дефолте. Это самый простой путь справедливого распределения канала между клиентами. Теперь повторим то же самое тестирование.

```

root@centos7:~# iperf3 -c 10.20.1.23 -p 5201 -i 5 -t 30
Connecting to host 10.20.1.23, port 5201
[ 4] local 192.168.88.196 port 51476 connected to 10.20.1.23 port 5201
[ ID] Interval      Transfer    Bandwidth   Retr  Cwnd
[ 4]  0.00-5.00    sec  28.5 MBytes  47.8 Mbits/sec  39  50.9 KBytes
[ 4]  5.00-10.00   sec  28.3 MBytes  47.4 Mbits/sec  36  43.8 KBytes
[ 4] 10.00-15.00   sec  28.3 MBytes  47.4 Mbits/sec  34  48.1 KBytes
[ 4] 15.00-20.00   sec  31.3 MBytes  52.5 Mbits/sec  44  41.0 KBytes
[ 4] 20.00-25.00   sec  56.6 MBytes  95.0 Mbits/sec 103  41.0 KBytes
[ 4] 25.00-30.00   sec  56.5 MBytes  94.9 Mbits/sec  98  39.6 KBytes
-----
[ ID] Interval      Transfer    Bandwidth   Retr
[ 4]  0.00-30.00   sec  230 MBytes  64.2 Mbits/sec  354
[ 4]  0.00-30.00   sec  229 MBytes  64.1 Mbits/sec
iperf Done.
root@centos7:~#

root@ubuntu18:~# iperf3 -c 10.20.1.23 -p 5202 -i 5 -t 30
Connecting to host 10.20.1.23, port 5202
[ 4] local 192.168.88.195 port 56636 connected to 10.20.1.23 port 5202
[ ID] Interval      Transfer    Bandwidth   Retr  Cwnd
[ 4]  0.00-5.00    sec  58.3 MBytes  97.9 Mbits/sec 140  39.6 KBytes
[ 4]  5.00-10.00   sec  56.5 MBytes  94.9 Mbits/sec  92  46.7 KBytes
[ 4] 10.00-15.00   sec  31.3 MBytes  52.5 Mbits/sec  42  39.6 KBytes
[ 4] 15.00-20.00   sec  28.3 MBytes  47.5 Mbits/sec  35  35.4 KBytes
[ 4] 20.00-25.00   sec  28.3 MBytes  47.5 Mbits/sec  36  39.6 KBytes
[ 4] 25.00-30.00   sec  28.2 MBytes  47.3 Mbits/sec  32  38.2 KBytes
-----
[ ID] Interval      Transfer    Bandwidth   Retr
[ 4]  0.00-30.00   sec  231 MBytes  64.6 Mbits/sec  377
[ 4]  0.00-30.00   sec  230 MBytes  64.4 Mbits/sec
iperf Done.
root@ubuntu18:~#

```

Работает четкое равномерное деление скорости интернет канала. В настройке **Target** у меня указан bridge, в который объединены все интерфейсы локальной сети. Там можно указать адрес или подсеть, например 192.168.88.0/24. Все зависит от того, что конкретно вы хотите сделать и между кем и кем поделить поровну интернет. Если у вас выход из этой локальной сети есть не только в интернет, но и в другие подсети, то укажите в **Dst** свой wan интерфейс, иначе скорость будет резаться всегда для всех адресов назначения.

Разберу основные параметры, которые есть в simple queue. Напомню, что все это можно посмотреть в документации.

Target	Как я уже сказал, источник, к которому будет применено queue правило. Можно указать интерфейс или ip адрес.
Dst (destination)	Интерфейс или адрес, куда будет направлен поток с ограничением по queue. Обычно какой-то внешний адрес.
Max Limit	Максимально разрешенная скорость upload/download для данной очереди.
Burst Limit	Максимально разрешенная скорость upload/download для включенного режима Burst.
Burst Threshold	Граница средней скорости, превышение которой выключает режим burst.

Burst Time	Интервал времени, в течении которого выполняется оценка средней скорости передачи данных (average-rate), которая используется для управления режимом burst.
Packet Marks	Здесь выбираются промаркированные пакеты, если используется маркировка.
Limit At	Скорость, которая выделяется очереди гарантированно.
Priority	Приоритет для потоков данной очереди. 1 - максимальный приоритет, 8 - самый низкий.
Queue Type	Выбор типа очереди, перечисленной в Queue Types.
Parent	Очередь, являющейся родителем по отношению к текущей.

Теперь некоторые комментарии к перечисленным параметрам.

1. Трафик с более высоким приоритетом будет иметь преимущество в достижении скорости, указанной в max-limit. Отсюда следствие - очень важно корректно указывать и следить за max-limit. Обычно его указывают на 5% ниже реальной скорости канала. Сумма параметров max-limit дочерних очередей может превышать лимит родительской очереди, но не может быть меньше limit-at. Приоритет работает только для дочерних очередей, не родительских. Очереди с наивысшим приоритетом будут иметь максимальный шанс на достижение указанного в них max-limit.
2. С помощью Dst удобно управлять очередями в случае использования нескольких каналов wan. Достаточно указывать их интерфейсы в правилах очередей.
3. Приоритизировать можно только исходящий трафик! Не все это понимают и пытаются настраивать приоритеты для входящего.
4. Параметр limit at используется, чтобы гарантировать очереди заданную скорость, если это возможно. При этом он имеет преимущество перед приоритетом. То есть с его помощью можно гарантировать какую-то полосу пропускания трафику с низким приоритетом. Если данный канал не используется полностью, то он доступен другим очередям.

В целом, настройка simple queue в Микротике не представляет какой-то особой сложности и интуитивно понятна. Исключение только параметр **Burst**. Рассмотрим его отдельно.

Режим Burst

В переводе с английского burst - вспышка, взрыв. В общем случае режим Burst используется для кратковременного увеличения полосы пропускания. С его помощью можно по определенным правилам **превышать значение max-limit** своего правила. При этом не могут быть превышены значения max-limit вышестоящего правила или rscq-rate типа очереди. С помощью настроек burst превышение скорости может быть как очень редким, так и частым.

Ранее я уже упоминал параметры очереди, которые относятся к режиму burst:

- burst-limit

- burst-time
- burst-threshold

С их помощью происходит настройка активации этого режима. Я попробую своими словами рассказать, как работает Burst. Не обещаю, что получится понятно всем, но я постараюсь донести информацию. С помощью параметра burst-limit задается максимальный порог скорости при работе burst. Этот порог должен быть больше max-limit, иначе теряется его смысл. С помощью параметра burst-time задается время, в течении которого учитывается текущая средняя скорость (average-rate). **Важно!!! Это не время работы Burst**, как я видел в некоторых описаниях в интернете. Значение average-rate рассчитывается каждую 1/16 интервала burst-time. Если у вас указано 10 секунд, средняя скорость будет рассчитываться каждую 0,625 секунды. Учтите, что на все это расходуются ресурсы процессора.

Дальше с помощью burst-threshold вы указываете порог average-rate, после превышения которого режим burst отключается. Таким образом, когда вы начинаете что-то качать, получаете максимальную скорость, указанную в burst-limit. Через некоторое время ваша средняя скорость возрастет до значения burst-threshold и режим burst будет отключен. Скорость будет равна значению max-limit. После того, как вы закончите закачку, режим burst не включится обратно сразу. Должно пройти немного времени, чтобы average-rate за интервал burst-time стал ниже burst-threshold и тогда burst включится обратно.

Рассмотрим условный пример. Это не точный расчет, я просто покажу принцип и примерные последствия, просчитанные на глазок по ходу написания статьи. Допустим, у вас указаны следующие параметры simple queue в вашем mikrotik:

- max-limit = 20M;
- burst-threshold = 15M;
- burst-time = 16s;
- burst-limit = 50M.

Вы начали качать большой файл. Первое время загрузка равна 50M (burst-limit), примерно на 5-й секунде средняя скорость за интервал в 16 (burst-time) секунд будет выше 15M (burst-threshold) (5 секунд на скорости 50M и 11 секунд 0M, пока вы еще ничего не качали, $50M \cdot 5\text{сек} + 0M \cdot 10\text{сек} / 16 = 15,625M$, что больше 15M) и режим burst отключается. Вы продолжаете качать со скоростью 20M (max-limit), пока не загрузите весь файл. После окончания загрузки должно пройти примерно 4 секунды, чтобы у вас обратно включился burst ($(20M \cdot 12\text{сек} + 0M \cdot 4\text{сек}) / 16 = 15M$). За это время у вас средняя скорость на интервале последних 16 секунд станет ниже 15M (burst-threshold). Это при условии, что вы не будете больше продолжать что-то скачивать.

Надеюсь, понятно объяснил :) Я в свое время очень крепко надо всем этим корпел, пока пытался разобраться. Здесь по хорошему надо график и табличку значений сделать, но мне не хочется этим заниматься. Ниже будет еще один конкретный пример, так что если не поняли, как работает burst, посмотрите его.

Ограничение скорости по IP

Продолжим изучать функционал qos в mikrotik на основе simple queue. Добавим к нашей предыдущей настройке ограничение максимальной скорости доступа в интернет для конкретного ip. Для этого создаем зависимую queue для ip 192.168.88.197 со следующими параметрами.

Simple Queue <queue2>

General Advanced Statistics Traffic Total Total Statistics

Name: queue2

Target: 192.168.88.197

Dst.:

Target Upload Target Download

Max Limit: 10M 10M bits/s

Burst

Burst Limit: unlimited unlimited bits/s

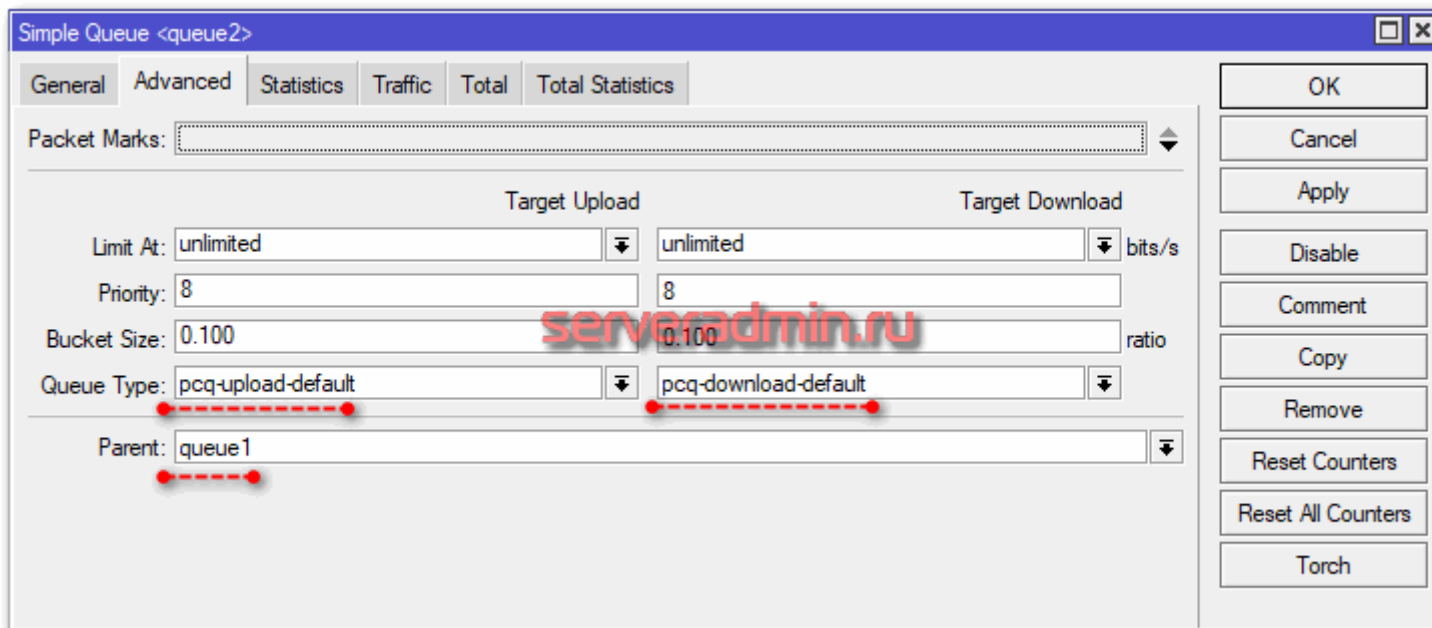
Burst Threshold: unlimited unlimited bits/s

Burst Time: 0 0 s

Time

enabled

OK
Cancel
Apply
Disable
Comment
Copy
Remove
Reset Counters
Reset All Counters
Torch

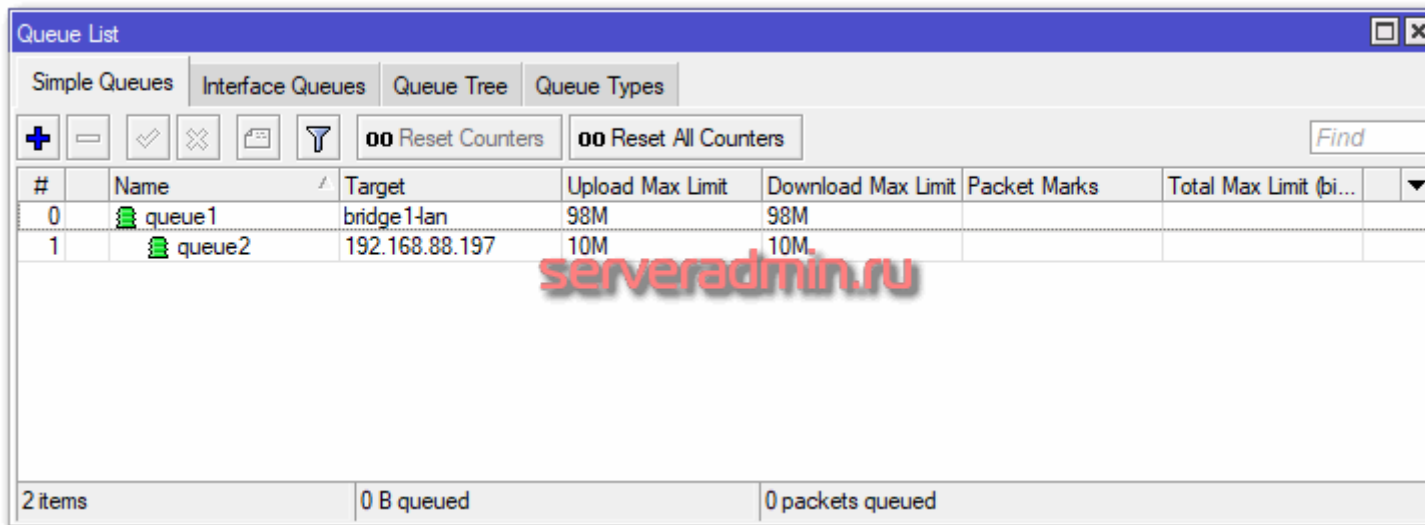


The screenshot shows the 'Simple Queue <queue2>' configuration window in Mikrotik WinBox. The window has several tabs: 'General', 'Advanced', 'Statistics', 'Traffic', 'Total', and 'Total Statistics'. The 'General' tab is active. The configuration fields are as follows:

- Packet Marks: (empty)
- Limit At: Target Upload: unlimited; Target Download: unlimited bits/s
- Priority: 8
- Bucket Size: 0.100; ratio: 0.100
- Queue Type: pcq-upload-default; pcq-download-default
- Parent: queue 1

Red dashed lines with dots at the ends are drawn under the 'Queue Type' and 'Parent' fields. A large red watermark 'serveradmin.ru' is overlaid on the center of the window. On the right side, there is a vertical stack of buttons: OK, Cancel, Apply, Disable, Comment, Copy, Remove, Reset Counters, Reset All Counters, and Torch.

Должно получиться вот так, если отсортировать список по #.



The screenshot shows the 'Queue List' window in Mikrotik WinBox. It features a toolbar with icons for adding, deleting, and filtering queues, along with buttons for 'Reset Counters' and 'Reset All Counters'. A search field labeled 'Find' is also present. The main area contains a table with the following data:

#	Name	Target	Upload Max Limit	Download Max Limit	Packet Marks	Total Max Limit (bi...
0	queue 1	bridge 1-lan	98M	98M		
1	queue 2	192.168.88.197	10M	10M		

At the bottom of the window, it displays '2 items', '0 B queued', and '0 packets queued'. A watermark 'serveradmin.ru' is visible in the center of the table.

Проверяем скорость доступа в интернет с помощью iperf.

```
Выбрать Командная строка
C:\Users\zerox\Downloads\iperf>iperf3 -c 10.20.1.23 -p 5203 -i 5 -t 30
Connecting to host 10.20.1.23, port 5203
 4] local 192.168.88.197 port 49833 connected to 10.20.1.23 port 5203
ID] Interval      Transfer      Bandwidth
 4]  0.00-5.02    sec  5.62 MBytes  9.41 Mbits/sec
 4]  5.02-10.02   sec  5.75 MBytes  9.65 Mbits/sec
 4] 10.02-15.02   sec  5.75 MBytes  9.65 Mbits/sec
 4] 15.02-20.02   sec  5.75 MBytes  9.65 Mbits/sec
 4] 20.02-25.02   sec  5.75 MBytes  9.65 Mbits/sec
 4] 25.02-30.02   sec  5.75 MBytes  9.65 Mbits/sec
-----
ID] Interval      Transfer      Bandwidth
 4]  0.00-30.02   sec  34.4 MBytes  9.61 Mbits/sec
 4]  0.00-30.02   sec  34.2 MBytes  9.56 Mbits/sec
iperf Done.
C:\Users\zerox\Downloads\iperf>
```

Как мы видим, ограничение скорости по ip работает. При этом скорость режется сразу, так как режим burst не настроен и не активирован. Теперь добавим к этому правилу настройку Burst и посмотрим, как она в реальности будет влиять на полосу пропускания. Для этого добавляем несколько параметров Burst. Я специально указал достаточно низкий burst-time, чтобы наглядно было видно результат работы burst.

Simple Queue <queue2>

General | Advanced | Statistics | Traffic | Total | Total Statistics

Name: queue2

Target: 192.168.88.197

Dst.:

Max Limit: 10M

Target Upload: 10M

Target Download: 10M bits/s

Burst

Burst Limit: 20M 20M bits/s

Burst Threshold: 15M 15M bits/s

Burst Time: 10 10 s

Time

OK
Cancel
Apply
Disable
Comment
Copy
Remove
Reset Counters
Reset All Counters
Torch

Смотрим, что получилось.

```
C:\Users\zerox\Downloads\iperf>iperf3 -c 10.20.1.23 -p 5203 -i 2 -t 30
Connecting to host 10.20.1.23, port 5203
[ 4] local 192.168.88.197 port 49840 connected to 10.20.1.23 port 5203
[ ID] Interval          Transfer           Bandwidth
[ 4]  0.00-2.02      sec  4.12 MBytes      17.2 Mbits/sec
[ 4]  2.02-4.02      sec  4.62 MBytes      19.4 Mbits/sec
[ 4]  4.02-6.00      sec  4.62 MBytes      19.5 Mbits/sec
[ 4]  6.00-8.02      sec  4.62 MBytes      19.3 Mbits/sec
[ 4]  8.02-10.02     sec  2.38 MBytes      9.96 Mbits/sec
[ 4] 10.02-12.02     sec  2.25 MBytes      9.44 Mbits/sec
[ 4] 12.02-14.01     sec  2.75 MBytes      11.5 Mbits/sec
[ 4] 14.01-16.02     sec  4.50 MBytes      18.9 Mbits/sec
[ 4] 16.02-18.02     sec  4.62 MBytes      19.4 Mbits/sec
[ 4] 18.02-20.02     sec  3.00 MBytes      12.6 Mbits/sec
[ 4] 20.02-22.02     sec  2.38 MBytes      9.96 Mbits/sec
[ 4] 22.02-24.02     sec  2.25 MBytes      9.44 Mbits/sec
[ 4] 24.02-26.02     sec  4.38 MBytes      18.4 Mbits/sec
[ 4] 26.02-28.02     sec  4.50 MBytes      18.9 Mbits/sec
[ 4] 28.02-30.02     sec  3.75 MBytes      15.7 Mbits/sec
-----
[ ID] Interval          Transfer           Bandwidth
[ 4]  0.00-30.02     sec  54.8 MBytes      15.3 Mbits/sec      sender
[ 4]  0.00-30.02     sec  54.6 MBytes      15.3 Mbits/sec      receiver

iperf Done.
```

Загрузка начинается на максимальной скорости. Потом отключается burst и скорость падает до max-limit. Через некоторое время падает average-rate и burst включается снова. И так несколько раз в течении всей загрузки канала.

Приоритет трафика по IP

Продолжим настройку qos на основе очередей, создав несколько правил, одно из которых будет давать приоритет трафику от конкретного IP. Для этого создаем 3 simple queue - одна основная и две зависимые. Как я уже говорил ранее, приоритеты работают только в дочерних очередях, не родительских.

Simple Queue <wan1>

General Advanced Statistics Traffic Total Total Statistics

Name: wan1

Target: 0.0.0.0/0

Dst.: ether1-wan1

Max Limit: 98M 98M bits/s

Burst Limit: unlimited unlimited bits/s

Burst Threshold: unlimited unlimited bits/s

Burst Time: 0 0 s

OK

Cancel

Apply

Disable

Comment

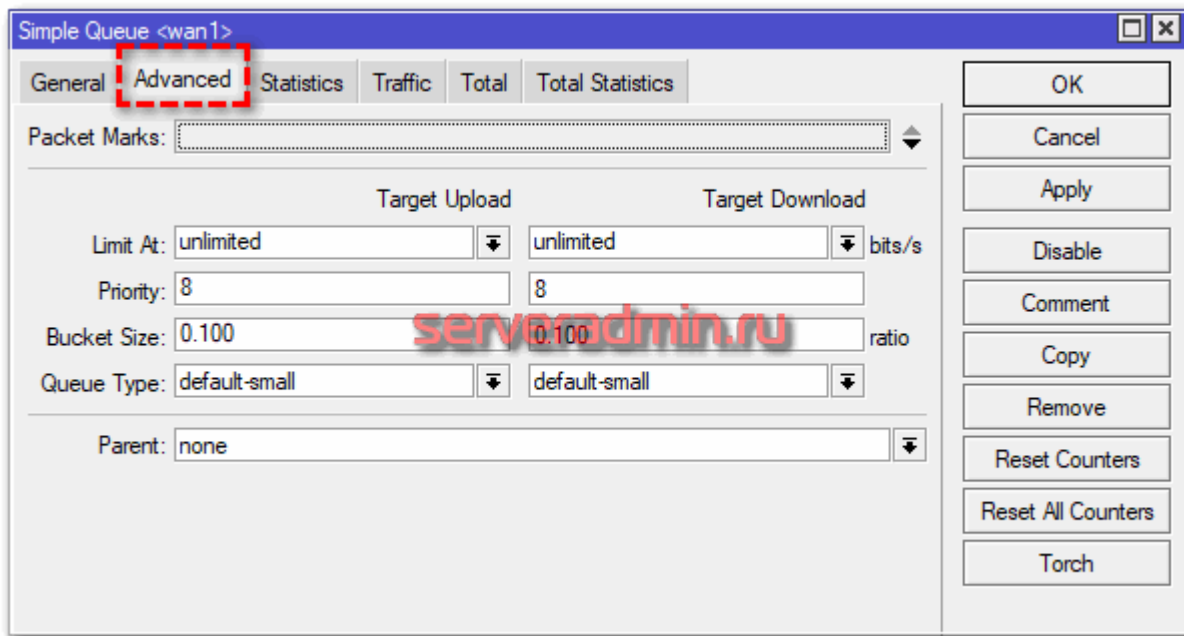
Copy

Remove

Reset Counters

Reset All Counters

Torch



Общая родительская очередь. Дальше идут потомки.

Simple Queue <IP-196-hi-Priority>

General | Advanced | Statistics | Traffic | Total | Total Statistics

Name: IP-196-hi-Priority

Target: 192.168.88.196

Dst.:

Max Limit: 98M

Target Upload: 98M

Target Download: 98M bits/s

Burst Limit: unlimited

Burst Threshold: unlimited

Burst Time: 0 s

Time

OK

Cancel

Apply

Disable

Comment

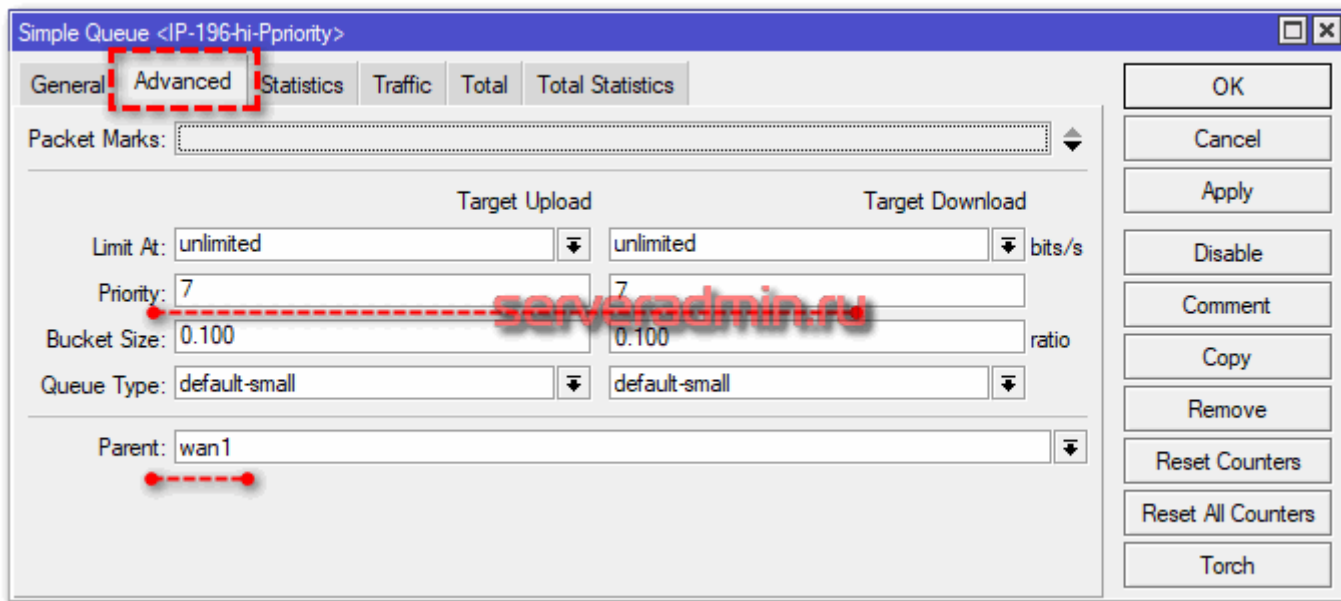
Copy

Remove

Reset Counters

Reset All Counters

Torch



Указываем ip адрес источника, для которого увеличиваем приоритет, и выбираем родителя. Остальное все то же самое, что в первом правиле.

Simple Queue <all>

General | Advanced | Statistics | Traffic | Total | Total Statistics

Name: all

Target: 192.168.88.0/24

Dst.:

Max Limit: 98M

Target Upload: 98M

Target Download: 98M bits/s

Burst Limit: unlimited

Burst Threshold: unlimited

Burst Time: 0

Time

OK

Cancel

Apply

Disable

Comment

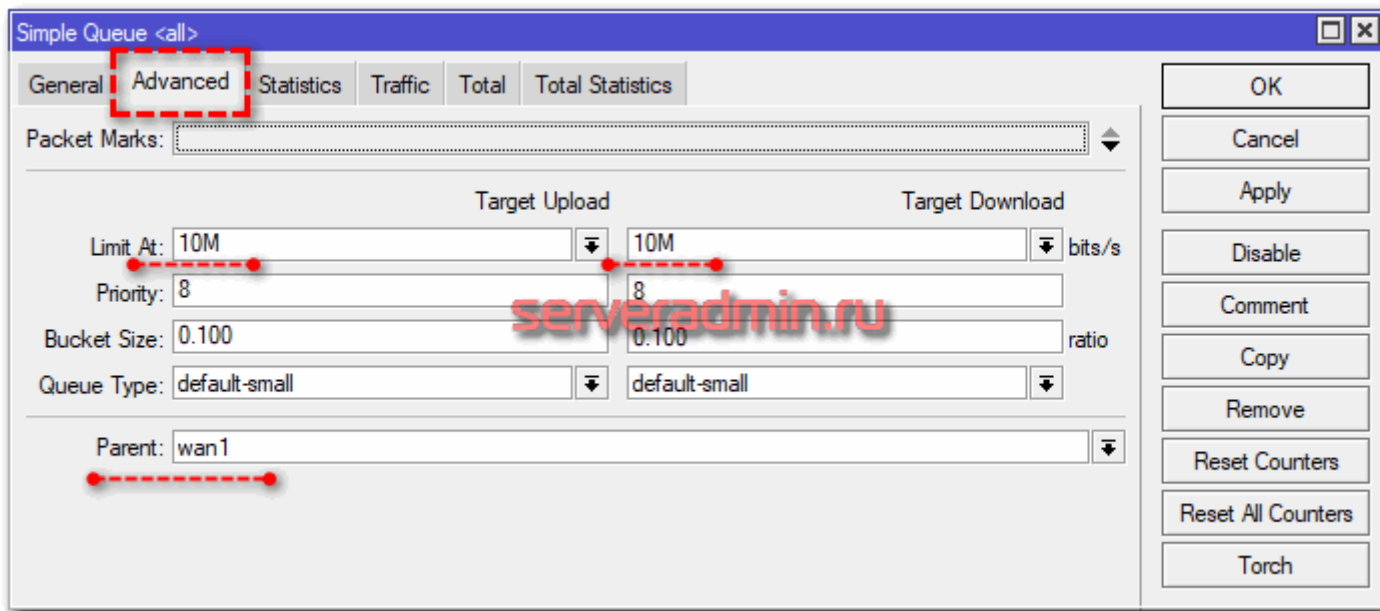
Copy

Remove

Reset Counters

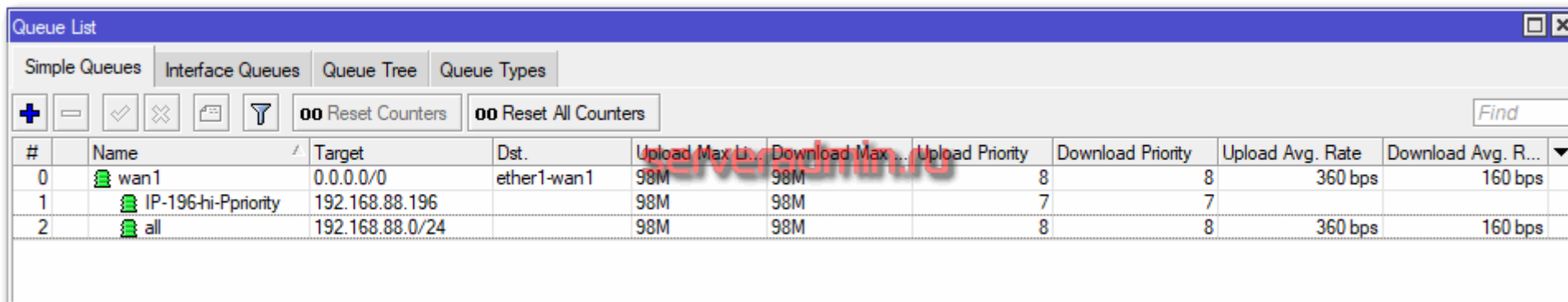
Reset All Counters

Torch



Здесь мы дополнительно указываем **limit-at**, чтобы для этого потока оставалось немного пропускной способности даже в то время, когда весь канал будет занят более приоритетным трафиком.

Должен получиться такой набор иерархически упорядоченных правил.



#	Name	Target	Dst.	Upload Max Li...	Download Max...	Upload Priority	Download Priority	Upload Avg. Rate	Download Avg. R...
0	wan1	0.0.0.0/0	ether1-wan1	98M	98M	8	8	360 bps	160 bps
1	IP-196-hi-Ppriority	192.168.88.196		98M	98M	7	7		
2	all	192.168.88.0/24		98M	98M	8	8	360 bps	160 bps

```
/queue simple  
add dst=ether1-wan1 max-limit=98M/98M name=wan1 target=""  
add max-limit=98M/98M name=IP-196-hi-Ppriority parent=wan1 priority=7/7 target=192.168.88.196/32  
add limit-at=10M/10M max-limit=98M/98M name=all parent=wan1 target=192.168.88.0/24
```

Важно следить за нумерацией правил. Пакеты попадают в правила по порядку. Как только подходящее правило найдено, пакет идет по нему. Поэтому правила с более узкими сегментами надо всегда поднимать выше общих правил, захватывающих весь остальной трафик.

Теперь запускаем тестирование приоритета трафика по ip. Сначала я запустил проверку скорости на машине с ip адресом, для которого приоритет не настроен. Загрузка началась на максимальной скорости для этой очереди. Через 6 секунд я стартовал такой же тест на машине с ip - 192.168.88.196, для которой мы увеличили приоритет с 8 до 7. Напомню, что меньше значение приоритета, тем он выше по факту. Нет необходимости ставить числа сильно ниже дефолта. Достаточно разницы на одну единицу, чтобы один трафик получил приоритет над другим.

```

root@centos7-ip-196:~# iperf3 -c 10.20.1.23 -p 5201 -i 3 -t 20
Connecting to host 10.20.1.23, port 5201
[ 4] local 192.168.88.196 port 43724 connected to 10.20.1.23 port 5201
[ ID] Interval      Transfer    Bandwidth  Retr  Cwnd
[ 4] 0.00-3.00 sec  30.5 MBytes 85.3 Mbits/sec  255 15.6 KBytes
[ 4] 3.00-6.00 sec  30.3 MBytes 84.7 Mbits/sec  330 14.1 KBytes
[ 4] 6.00-9.00 sec  30.4 MBytes 84.9 Mbits/sec  349 15.6 KBytes
[ 4] 9.00-12.00 sec 30.3 MBytes 84.6 Mbits/sec  336 12.7 KBytes
[ 4] 12.00-15.00 sec 30.5 MBytes 85.2 Mbits/sec  329 15.6 KBytes
[ 4] 15.00-18.00 sec 30.4 MBytes 84.9 Mbits/sec  338 15.6 KBytes
[ 4] 18.00-20.00 sec 20.3 MBytes 85.0 Mbits/sec  227 15.6 KBytes
-----
[ ID] Interval      Transfer    Bandwidth  Retr
[ 4] 0.00-20.00 sec 202 MBytes 84.9 Mbits/sec 2264
[ 4] 0.00-20.00 sec 202 MBytes 84.9 Mbits/sec
iperf Done.
root@centos7-ip-196 ~#

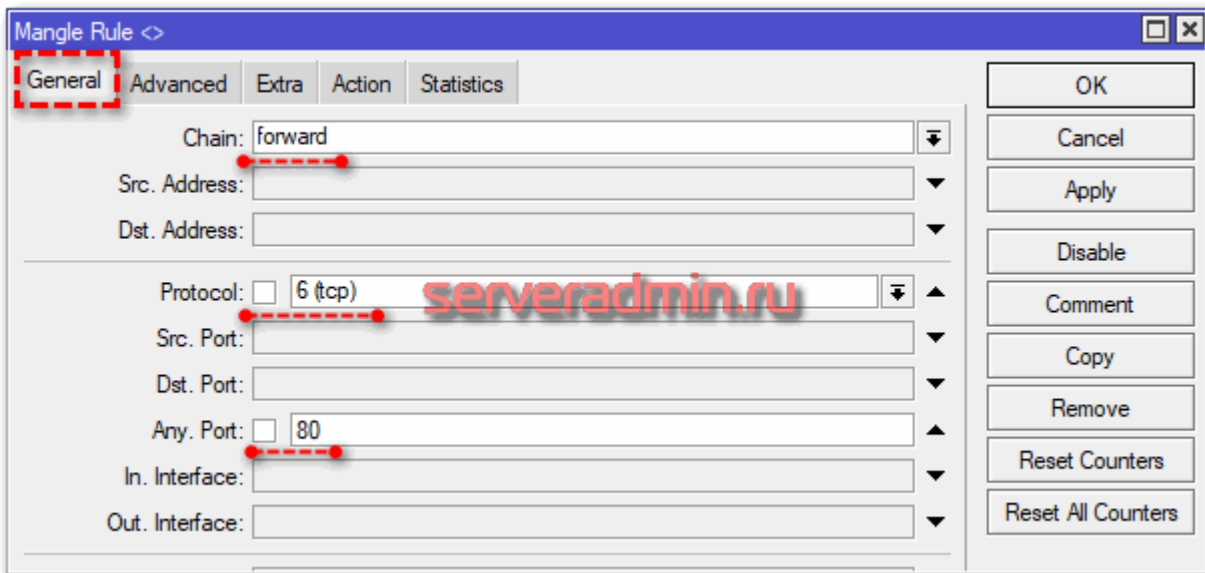
root@ubuntu18-ip-195:~# iperf3 -c 10.20.1.23 -p 5202 -i 3 -t 35
Connecting to host 10.20.1.23, port 5202
[ 4] local 192.168.88.195 port 37948 connected to 10.20.1.23 port 5202
[ ID] Interval      Transfer    Bandwidth  Retr  Cwnd
[ 4] 0.00-3.00 sec  35.8 MBytes 100 Mbits/sec  733 21.2 KBytes
[ 4] 3.00-6.00 sec  33.9 MBytes 94.9 Mbits/sec  588 15.6 KBytes
[ 4] 6.00-9.00 sec  16.0 MBytes 44.8 Mbits/sec  251 17.0 KBytes
[ 4] 9.00-12.00 sec  3.73 MBytes 10.4 Mbits/sec  45 12.7 KBytes
[ 4] 12.00-15.00 sec  3.54 MBytes 9.90 Mbits/sec  44 12.7 KBytes
[ 4] 15.00-18.00 sec  3.67 MBytes 10.3 Mbits/sec  42 14.1 KBytes
[ 4] 18.00-21.00 sec  3.54 MBytes 9.90 Mbits/sec  41 11.3 KBytes
[ 4] 21.00-24.00 sec  3.54 MBytes 9.90 Mbits/sec  43 12.7 KBytes
[ 4] 24.00-27.00 sec  3.48 MBytes 9.73 Mbits/sec  38 12.7 KBytes
[ 4] 27.00-30.00 sec 31.6 MBytes 88.4 Mbits/sec  526 14.1 KBytes
[ 4] 30.00-33.00 sec 33.9 MBytes 94.9 Mbits/sec  562 14.1 KBytes
[ 4] 33.00-35.00 sec 22.7 MBytes 95.1 Mbits/sec  381 15.6 KBytes
-----
[ ID] Interval      Transfer    Bandwidth  Retr
[ 4] 0.00-35.00 sec 195 MBytes 46.8 Mbits/sec 3294
[ 4] 0.00-35.00 sec 195 MBytes 46.7 Mbits/sec
iperf Done.
root@ubuntu18-ip-195:~#
  
```

В итоге мы видим, что сначала была максимальная загрузка трафика на первой машине, потом приоритет был отдан второй, а у первой осталась скорость из значения limit-at. После того, как более приоритетный трафик кончился, вся доступная полоса пропускания опять досталась первому ip.

Приоритет HTTP трафика для максимально быстрого серфинга

Давайте теперь к нашим правилам добавим еще одно - приоритет http трафика. Благодаря этому, серфинг интернета будет всегда комфортным, даже если весь канал занят чем-нибудь другим. Причем мы сделаем так, чтобы http трафик не смог занять весь канал, так как с максимальным приоритетом это будет сделать не сложно. Особенно, если качать что-то объемное. Для этого мы его ограничим через max-limit и кратковременно ускорим через burst, чтобы серфить было комфортно.

Начнем настройку приоритета для http трафика. Для этого его надо промаркировать. Идем в IP -> Firewall -> Mangle и добавляем 2 правила маркировки.



Mangle Rule <>

General Advanced Extra Action Statistics

Chain: forward

Src. Address:

Dst. Address:

Protocol: 6 (tcp) serveradmin.ru

Src. Port:

Dst. Port:

Any. Port: 80

In. Interface:

Out. Interface:

OK

Cancel

Apply

Disable

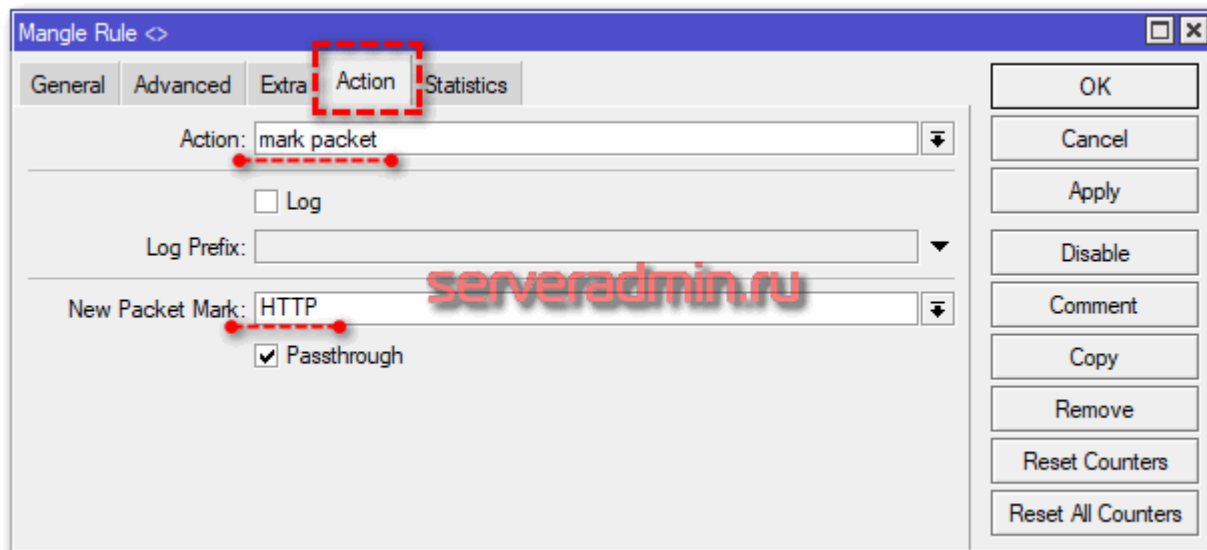
Comment

Copy

Remove

Reset Counters

Reset All Counters



То же самое делаем для https трафика, просто указав порт 443. Вот оба правила.

```
/ip firewall mangle
add action=mark-packet chain=forward comment=HTTP new-packet-mark=HTTP passthrough=yes port=80 protocol=tcp
add action=mark-packet chain=forward comment=HTTPS new-packet-mark=HTTP passthrough=yes port=443 protocol=tcp
```

Теперь идем добавлять новую очередь с более высоким приоритетом для веб трафика.

The screenshot shows the Mikrotik Simple Queue configuration window for a queue named 'http_hi_Priority'. The 'General' tab is selected. The configuration includes the following fields:

- Name: http_hi_Priority
- Target: 0.0.0.0/0
- Max Limit: 20M (Target Upload) / 20M (Target Download) bits/s
- Burst Limit: 50M (Target Upload) / 50M (Target Download) bits/s
- Burst Threshold: 15M (Target Upload) / 15M (Target Download) bits/s
- Burst Time: 16 s

The 'Burst Limit', 'Burst Threshold', and 'Burst Time' fields for both Target Upload and Target Download are highlighted with a red dashed box. The 'General' tab is also highlighted with a red dashed box. On the right side of the window, there are buttons for OK, Cancel, Apply, Disable, Comment, Copy, Remove, Reset Counters, Reset All Counters, and Torch.

Simple Queue <http_hi_Priority>

General **Advanced** Statistics Traffic Total Total Statistics

Packet Marks: HTTP

Limit At: 10M 10M bits/s

Priority: 6 6

Bucket Size: 0.100 0.100 ratio

Queue Type: default-small default-small

Parent: wan1

OK
Cancel
Apply
Disable
Comment
Copy
Remove
Reset Counters
Reset All Counters
Torch

```
add burst-limit=50M/50M burst-threshold=15M/15M burst-time=16s/16s limit-at=10M/10M max-limit=20M/20M  
name=http_hi_Priority packet-marks=HTTP parent=wan1 priority=6/6 target=""
```

В итоге имеем следующий набор правил:

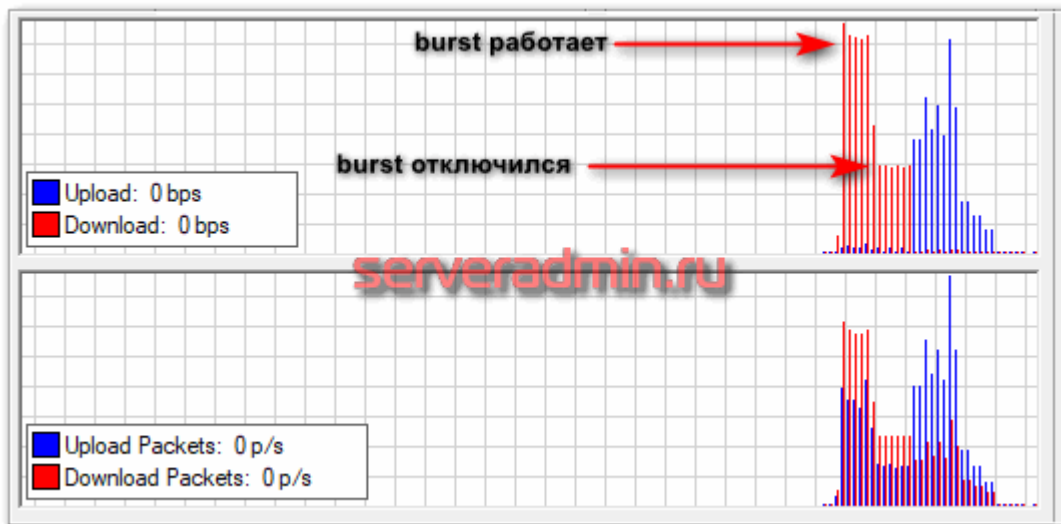
#	Name	Target	Dst	Upload Max Li...	Download Max ...	Upload Priority	Download Priority	Up...
0	wan1	0.0.0.0/0	ether1-wan1	98M	98M	8	8	
1	http_hi_Priority	0.0.0.0/0		20M	20M	6	6	
3	all	192.168.88.0/24		98M	98M	8	8	
2	IP-196-hi-Ppriority	192.168.88.196		98M	98M	7	7	

Не забываем про порядок правил в простых очередях. Он имеет значение, в отличие от queue tree.

Теперь проверяем, насколько заметно и корректно в итоге работает приоритизация http трафика. Я, как обычно, запущу iperf на двух других машинах в сети и параллельно с этим запущу веб серфинг на еще одном компьютере. Вот какая картина будет в списке очередей.

#	Name	Target	Dst	Upload Max Li...	Download Max ...	Uploa...
0	wan1	0.0.0.0/0	ether1-wan1	98M	98M	
1	http_hi_Priority	0.0.0.0/0		20M	20M	
3	all	192.168.88.0/24		98M	98M	
2	IP-196-hi-Ppriority	192.168.88.196		98M	98M	

Если посмотреть статистику очереди с приоритетом http трафика, то там будет такая картинка.



Напоминаю, что при этом канал в интернет забит полностью другим трафиком, но из-за более высокого приоритета, www трафик бежит с заданными лимитами на ширину канала.

The image shows a screenshot of a desktop environment with three windows:

- Terminal 1 (Left):** Shows a netperf test on a Centos7 host. The output table shows bandwidth values fluctuating between approximately 62.3 and 94.9 Mbits/sec. Red boxes highlight the first few rows, and Russian text annotations state: "http трафика нет, максимальный приоритет 7 по ip здесь" and "работает тест на загрузку http трафика".
- Terminal 2 (Right):** Shows a netperf test on an Ubuntu18 host. The output table shows bandwidth values starting at 96.2 Mbits/sec and then dropping to around 10 Mbits/sec. Red boxes highlight the first two rows, and Russian text annotations state: "весь канал был занят только этим трафиком с приоритетом 8".
- Browser Window:** Shows the Yandex speed test page (Яндекс Интернетометр) with the URL <https://yandex.ru/internet/>. The page displays user data and internet speed test results:

ДАННЫЕ О ПОЛЬЗОВАТЕЛЕ		СКОРОСТЬ ИНТЕРНЕТА	
IPv4-адрес	[blurred]	Входящее соединение	42.14 Мбит/с = 5.27 МБайт/с
IPv6-адрес	-	Исходящее соединение	29.22 Мбит/с = 3.65 МБайт/с
Браузер	Edge 85.0.564 (WebKit 537.36)	<input type="button" value="Измерить ещё раз"/> <input type="button" value="Поделиться"/>	

Я заметил небольшую особенность, причину которой не понял. Во время теста на ip 195, что на картинке справа, четко действует лимит в 10М, оставляя указанную полосу, даже если канал забивает более приоритетный трафик с сервера 196 слева. Но когда я запускают тест http трафика, он работает на максимальной скорости, но при этом у ip 195 полоса увеличивается примерно на 3-4М и становится в районе 13-14М, что выше указанного limit-at для этой очереди. Я сначала думал, что это просто погрешность какая-то, но эффект стабильно воспроизводится. Как только заканчивается http тест, скорость четко опускается до лимита в 10М. С чем это связано, я не понял.

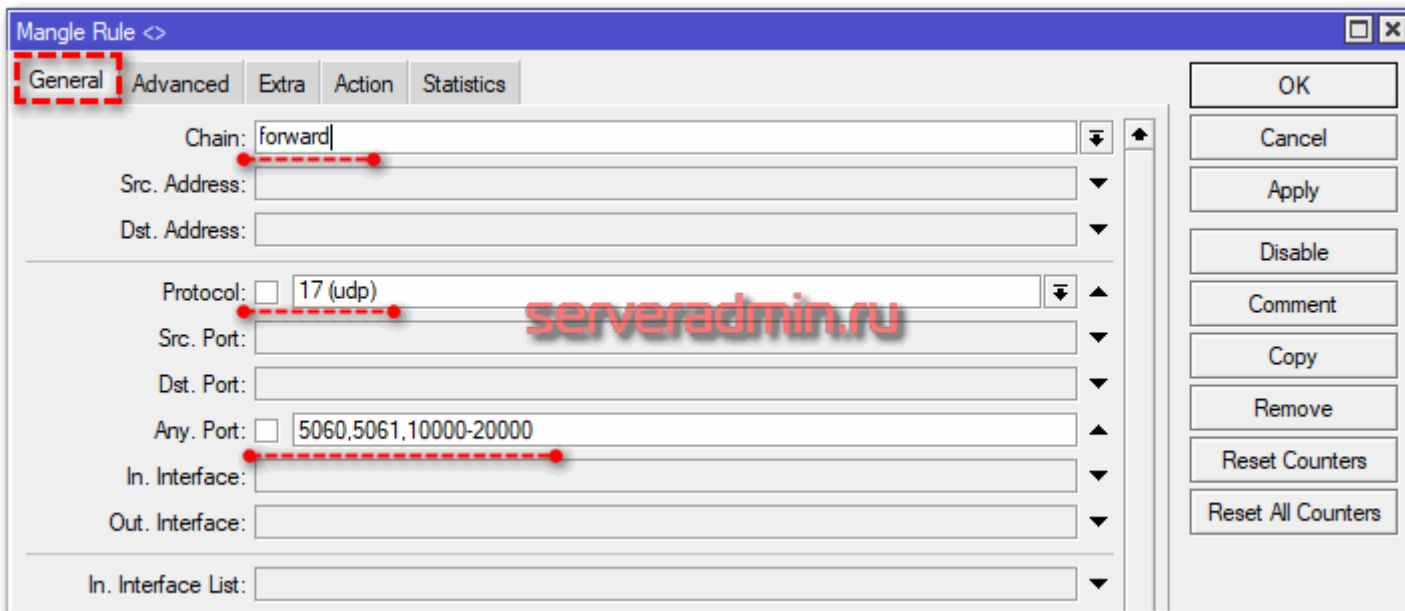
Надеюсь, общая идея понятна. По такому принципу можно маркировать любой трафик и встраивать в дерево очередей на основе маркировки.

Приоритет SIP трафика для VOIP

Настроить приоритет SIP трафика можно как минимум двумя принципиально разными способами:

1. Самый простой случай, когда вам нужно настроить приоритет SIP трафика с вашего сервера VOIP в интернет. Для этого можно воспользоваться предыдущим примером с настройкой приоритета по ip, явно создав правило с высоким приоритетом для ip адреса сервера телефонии. Этого будет достаточно, если у вас в сети свой сервер, а все клиенты звонят через него.
2. Если же у вас в сети много sip клиентов, которые напрямую подключаются к voip серверу через интернет и выделить их в отдельную подсеть не представляется возможным, то придется действовать по-другому. Нужно маркировать весь sip трафик и отдавать приоритет на основе этой маркировки. Такой случай разобран выше на примере http трафика.

Так что в зависимости от своей ситуации, подбирайте реализацию приоритета конкретно под ваш случай. Я покажу пример второго способа с маркировкой. Снова идем в **IP -> Firewall -> Mangle** и добавляем правило маркировки для sip трафика.



Mangle Rule <>

General Advanced Extra Action Statistics

Chain: forward

Src. Address:

Dst. Address:

Protocol: 17 (udp)

Src. Port:

Dst. Port:

Any. Port: 5060,5061,10000-20000

In. Interface:

Out. Interface:

In. Interface List:

OK

Cancel

Apply

Disable

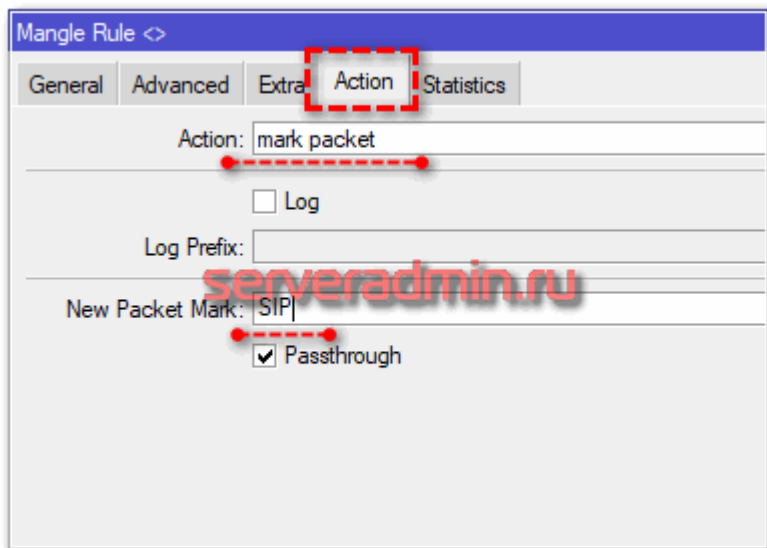
Comment

Copy

Remove

Reset Counters

Reset All Counters



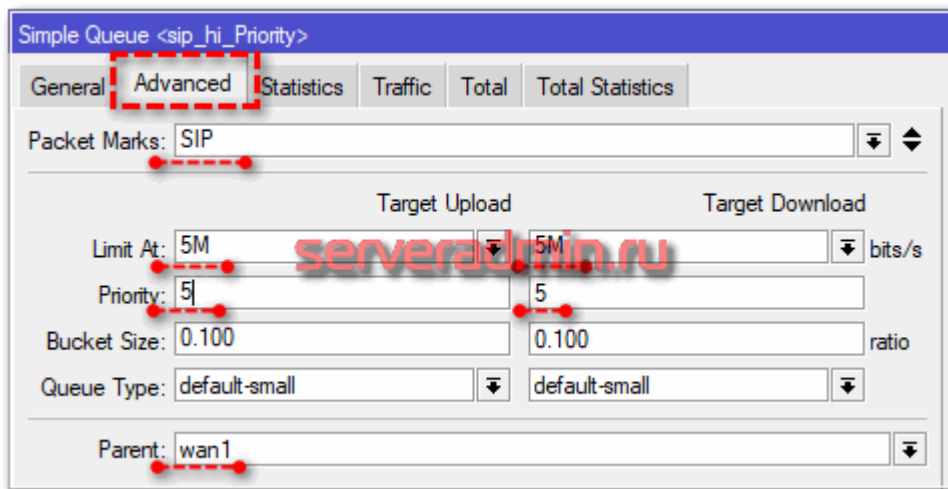
```
add action=mark-packet chain=forward comment=SIP new-packet-mark=SIP passthrough=yes port=5060,5061,10000-20000 protocol=udp
```

И как обычно, добавляем еще одну simple queue с более высоким приоритетом промаркированного sip трафика.

The screenshot shows the 'Simple Queue' configuration window for a queue named 'sip_hi_Priority'. The 'General' tab is selected and highlighted with a red dashed box. The configuration includes:

- Name: sip_hi_Priority
- Target: 0.0.0.0/0
- Dst.: (empty)
- Max Limit: 20M (Target Upload) and 20M (Target Download) bits/s
- Burst: (expanded) Burst Limit: unlimited (Target Upload) and unlimited (Target Download) bits/s
- Burst Threshold: unlimited (Target Upload) and unlimited (Target Download) bits/s
- Burst Time: 0 s (Target Upload) and 0 s (Target Download)
- Time: (collapsed)

On the right side of the window, there are several control buttons: OK, Cancel, Apply, Disable, Comment, Copy, Remove, Reset Counters, Reset All Counters, and Torch. A watermark 'serveradmin.ru' is visible across the center of the window.



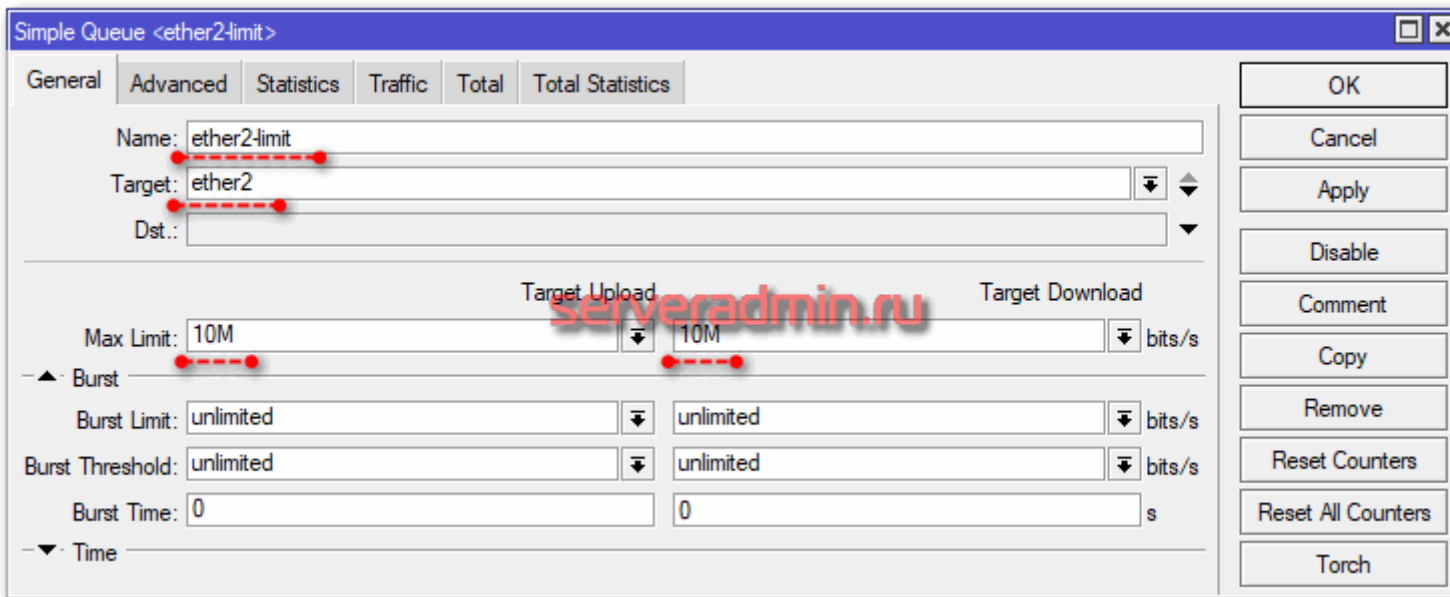
Думаю, что для sip трафика burst не нужен. Я не могу себе представить ситуацию, когда возможен какой-то резкий скачек трафика у отдельно взятого абонента. Не забывайте про порядок правил очередей. С более высоким приоритетом правила ставим выше.

На практике показать работу приоритета для sip трафика затруднительно. Я не знаю, как практически это сделать. В итоге проверил так. Взял утилиту **sipp**, которая есть для linux. Далее запустил ее и нагенерировал sip трафик. Убедился, что он попал в правила маркировки, а так же в указанную очередь. Если счетчики пакетов там растут, значит все сделали правильно и приоритезация будет работать. Ведь работу самих очередей мы уже проверили ранее.

Ограничение скорости на интерфейсе

В заключении статьи про qos в mikrotik рассмотрим пример с ограничением скорости на конкретном интерфейсе. Если у вас все получилось с очередями ранее, думаю, проблем с лимитированием скорости интерфейса не возникнет. Для этого достаточно создать отдельную очередь в simple queue, у которой в параметре Target указать нужный интерфейс. Причем это может быть как физический интерфейс, так и бридж, либо какое-то виртуальное соединение (pptp, l2tp и т.д.).

Итак, включаем ограничение скорости физического интерфейса микротика. Идем в **Queues** и добавляем правило.

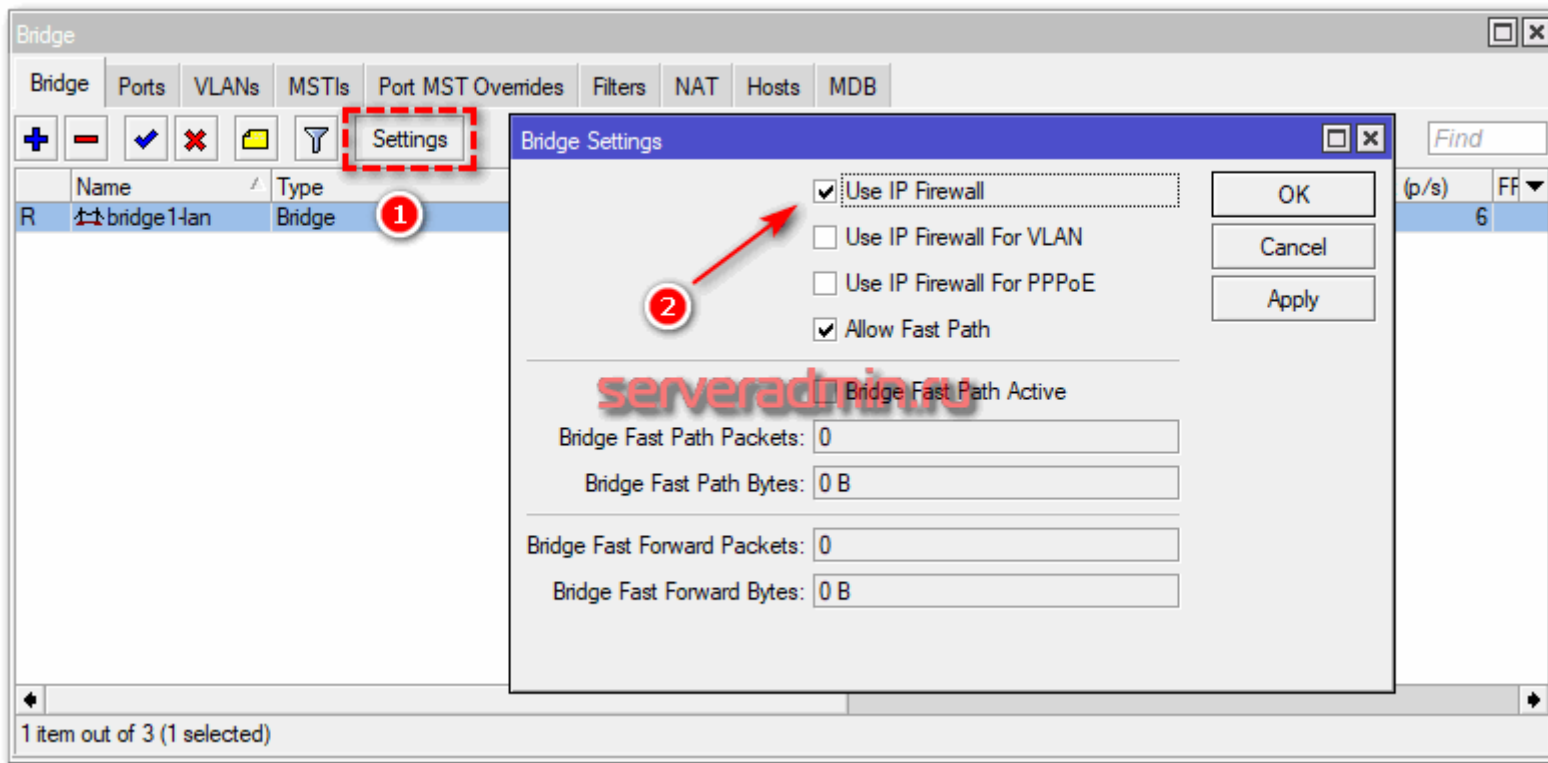


The screenshot shows the 'Simple Queue' configuration window for a queue named 'ether2-limit'. The window has several tabs: 'General', 'Advanced', 'Statistics', 'Traffic', 'Total', and 'Total Statistics'. The 'General' tab is active. The configuration fields are as follows:

- Name: ether2-limit
- Target: ether2
- Dst.: (empty)
- Max Limit: 10M
- Target Upload: 10M
- Target Download: 10M
- Burst Limit: unlimited
- Burst Threshold: unlimited
- Burst Time: 0

On the right side of the window, there are several buttons: OK, Cancel, Apply, Disable, Comment, Copy, Remove, Reset Counters, Reset All Counters, and Torch. A watermark 'serveradmin.ru' is visible in the center of the window.

Важно понимать один нюанс. Если у вас порт находится в bridge, то данное ограничение работать не будет. Необходимо в качестве Target указывать весь бридж. Если же вам это не подходит и хочется ограничить скорость интерфейса в бридже, но не убирать его оттуда, то надо маркировать пакеты. Для этого в Mangle надо делать маркировку по признаку In / Out Bridge Port. Чтобы она заработала, в свойствах бриджа необходимо включить Use Ip Firewall.



Ну и дальше в queue указывать маркировку пакетов из бриджа, как мы уже делали ранее. Я на практике это не проверял, так как не смог смоделировать на своем тестовом стенде. У меня все абоненты подключены к одному сетевому интерфейсу микротика.

Заключение

Не понравилась статья и хочешь научить меня администрировать? Пожалуйста, я люблю учиться. Комментарии в твоём распоряжении. Расскажи, как сделать правильно!

На этом у меня по настройке qos в Mikrotik все. Постарался рассмотреть основные моменты на простых примерах и разложить все по полочкам, чтобы на основе этого можно было настраивать более сложные конфигурации. Лично я никогда не использовал Queue Tree, так как все вопросы по приоритизации трафика удавалось решить возможностями Simple Queues.

Важное замечание по написанному. Я не являюсь большим специалистом по сетям и по микротикам в частности. Мои статьи это мой небольшой опыт и мои старания по изучению документации и другого пользовательского опыта. Я могу в чем-то заблуждаться, где-то ошибаться и советовать плохие практики. Если видите это, то говорите, как сделать лучше, правильнее, удобнее. В следующей редакции статьи я обязательно это учту. Я регулярно обновляю статьи в том числе и по мотивам замечаний в комментариях.

То же самое касается вопросов и дополнений. Если вам кажется, что я не рассмотрел какую-то важную настройку или конфигурацию по теме qos в микротиках, скажите об этом. В следующий раз я постараюсь рассмотреть и этот вопрос. Например, моя статья про настройку firewall в mikrotik претерпела уже 3 редакции.

Другие мои статьи на тему Mikrotиков:

- Бесшовный wifi роуминг с помощью CAPsMAN.
- Доступ к Mikrotik по двум и более внешним IP.
- Защита Mikrotik.
- Блокировка сайта Mikrotikom.
- My "holy war" against masquerade.
- Мониторинг Mikrotika.
- Настройка vpn сервера.
- Централизованный сбор логов с Mikrotиков.

И многие другие в разделе, полностью посвящённому Mikrotik.

Онлайн курсы по Mikrotik

Если у вас есть желание научиться работать с роутерами микротик и стать специалистом в этой области, рекомендую пройти курсы по программе, основанной на информации из официального курса **MikroTik Certified Network Associate**. Помимо официальной программы, в курсах будут лабораторные работы, в которых вы на практике сможете проверить и закрепить полученные знания. Все подробности на

сайте . Стоимость обучения весьма демократична, хорошая возможность получить новые знания в актуальной на сегодняшний день предметной области. Особенности курсов:

- Знания, ориентированные на практику;
- Реальные ситуации и задачи;
- Лучшее из международных программ.

Помогла статья? Подписывайся на telegram канал автора

Анонсы всех статей, плюс много другой полезной и интересной информации, которая не попадает на сайт.