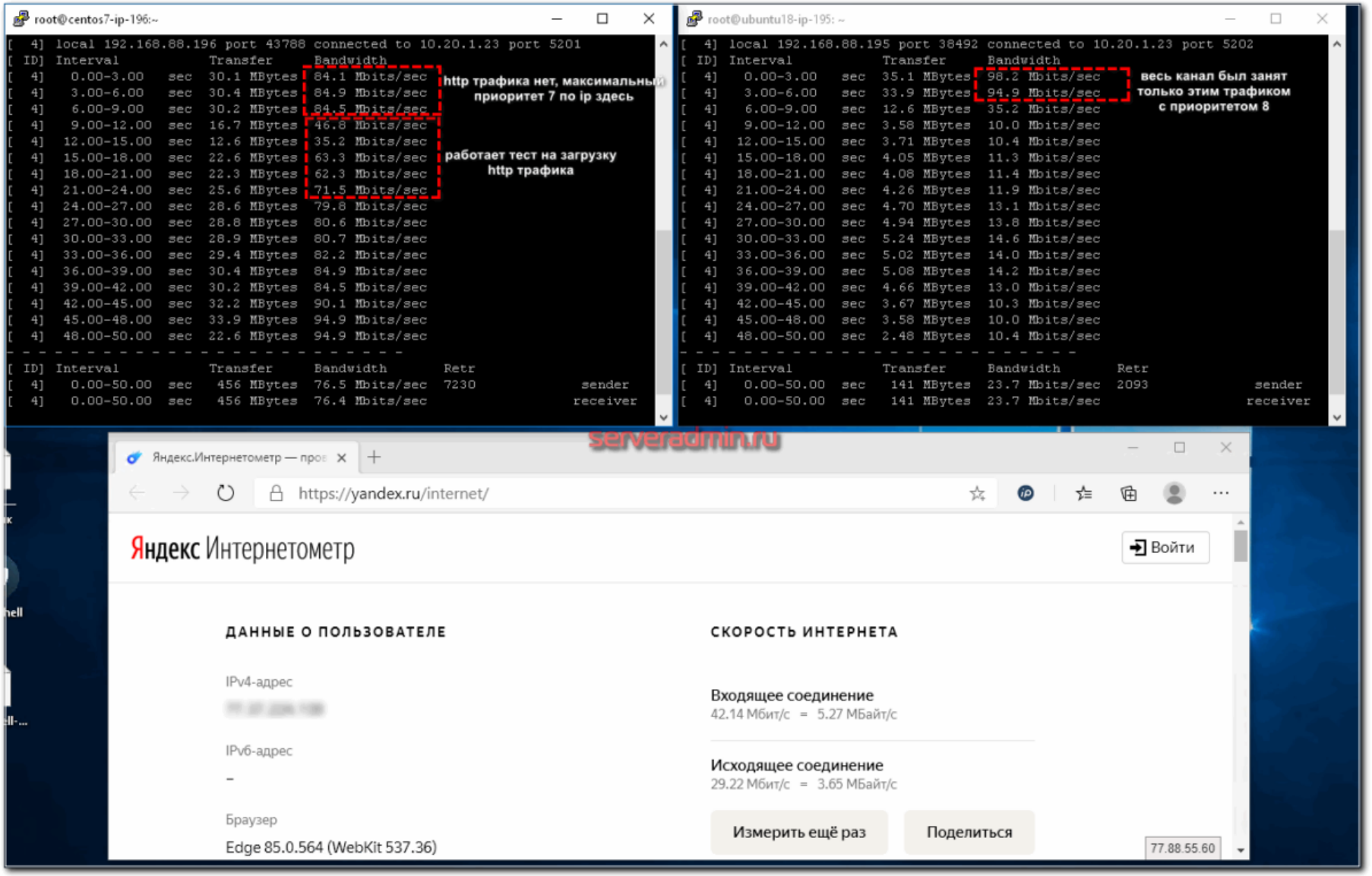

root@centos7-ip-196:~
[4] local 192.168.88.196 port 43788 connected to 10.20.1.23 port 5201
[ID] Interval Transfer Bandwidth
[4] 0.00-3.00 sec 30.1 MBytes 84.1 Mbits/sec
[4] 3.00-6.00 sec 30.4 MBytes 84.9 Mbits/sec
[4] 6.00-9.00 sec 30.2 MBytes 84.5 Mbits/sec
[4] 9.00-12.00 sec 16.7 MBytes 46.8 Mbits/sec
[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
[4] 0.00-50.00 sec 456 MBytes 76.5 Mbits/sec 7230 sender
[4] 0.00-50.00 sec 456 MBytes 76.4 Mbits/sec receiver
http трафика нет, максимальный приоритет 7 по ip здесь
работает тест на загрузку http трафика
root@ubuntu18-ip-195: ~
[4] local 192.168.88.195 port 38492 connected to 10.20.1.23 port 5202
[ID] Interval Transfer Bandwidth
[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
[4] 0.00-50.00 sec 141 MBytes 23.7 Mbits/sec 2093 sender
[4] 0.00-50.00 sec 141 MBytes 23.7 Mbits/sec receiver
весь канал был занят только этим трафиком с приоритетом 8
serveradmin.ru
Яндекс.Интернетометр — про
https://yandex.ru/internet/
Яндекс Интернетометр
Войти
ДАННЫЕ О ПОЛЬЗОВАТЕЛЕ
IPv4-адрес
IPv6-адрес
–
Браузер
Edge 85.0.564 (WebKit 537.36)
СКОРОСТЬ ИНТЕРНЕТА
Входящее соединение
42.14 Мбит/с = 5.27 МБайт/с
Исходящее соединение
29.22 Мбит/с = 3.65 МБайт/с
Измерить ещё раз
Поделиться
77.88.55.60

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

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 траффик в основном безлимитный, QOS стал не нужен. Очевидно, что это не так, потому что с ростом скоростей выросло и потребление траффика в виде просмотра потокового видео и закачки 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 Микротику, которое я проходил.

В интернете есть много информации по теме настройки миркотиков 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.

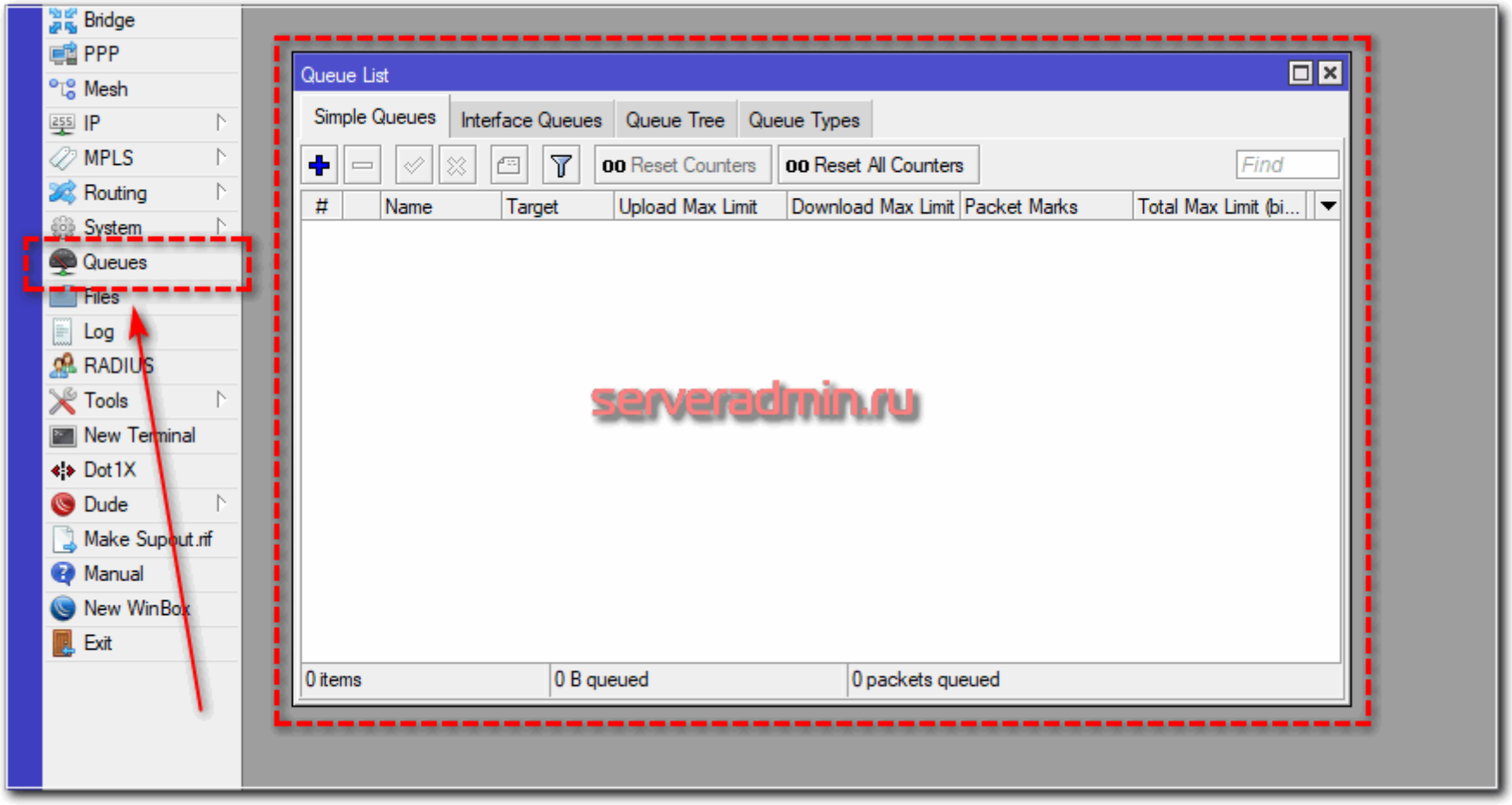
Если вы новичок в микротиках, или еще не сделали базовую настройку, рекомендую мою статью на эту тему - настройка mikrotik.

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:

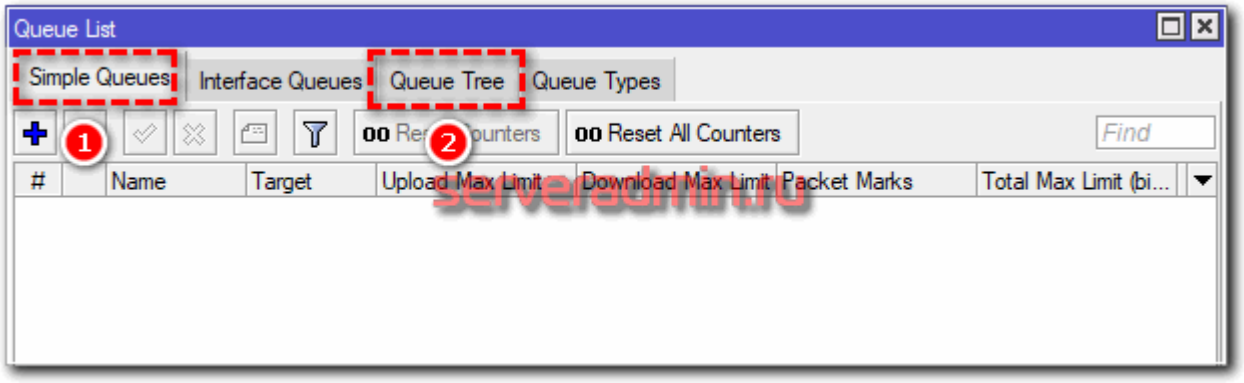
1. **FIFO** - BFIFO, PFIFO, MQ PFIFO. Алгоритм FIFO (first in — first out, первый пришёл - первый ушёл) подразумевает, что данные, попадающие в очередь первыми, будут первыми переданы на дальнейшую обработку. Тот, кто пришел позже, ждет окончания обработки поступивших ранее. Подобный алгоритм применяется 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 выравнивать 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 приоритезации 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.

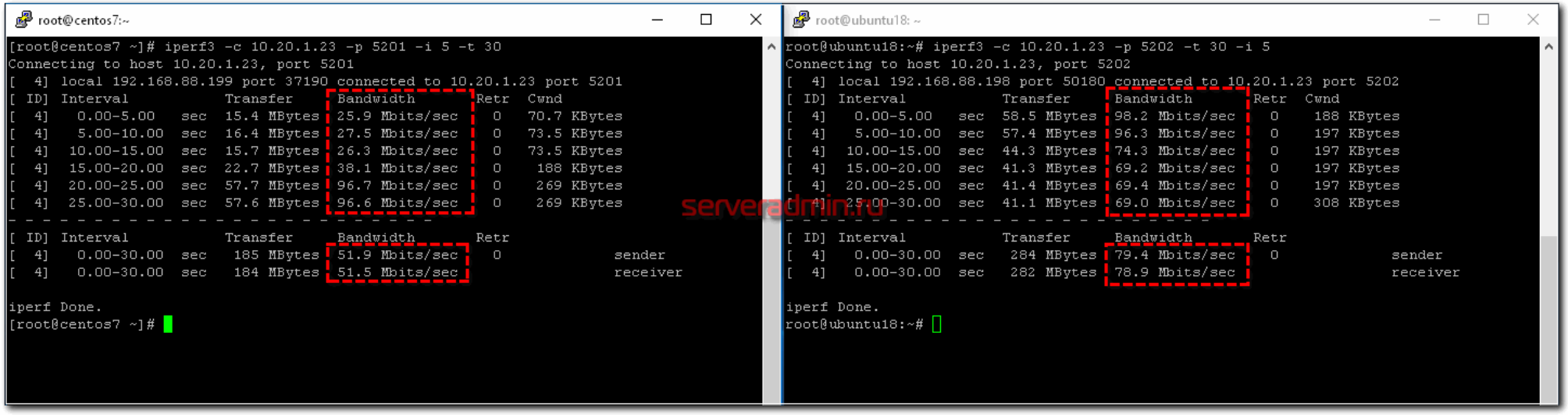
Что же выбрать для настройки QOS - Simple queue или Queue Tree? Я советую начать с простых очередей. Если ваша задача не решается с их помощью, переходите на иерархические деревья правил с маркировкой. И еще нужно понимать, что Simple queue это частный случай Queue Tree, поэтому при использовании обоих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

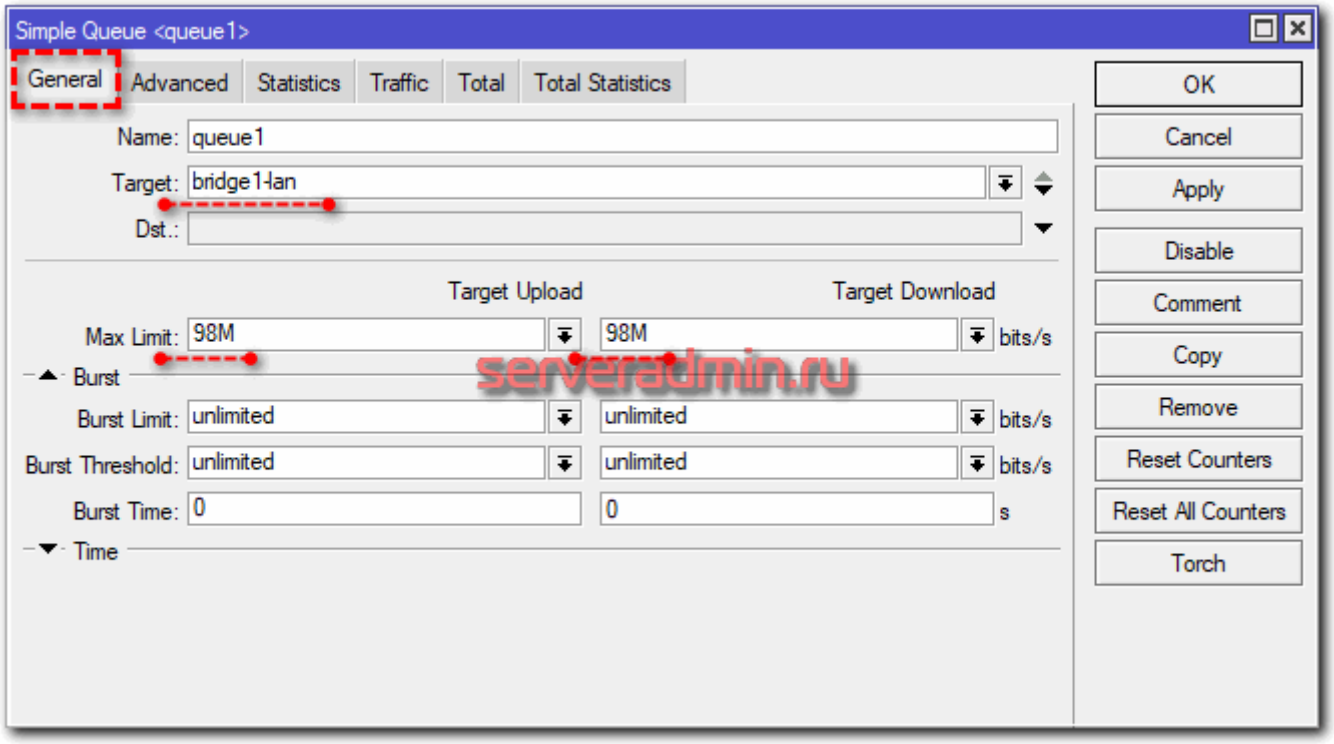
Адрес Микротика - 192.168.88.1. Измерять скорость буду с помощью программы **iperf**. Канал в интернет - 100 Мбит/с. Для начала посмотрим, как будет распределяться канал между клиентами без настройки qos. Для этого просто запускаю iperf на обоих клиентах с разницей в 10 секунд.

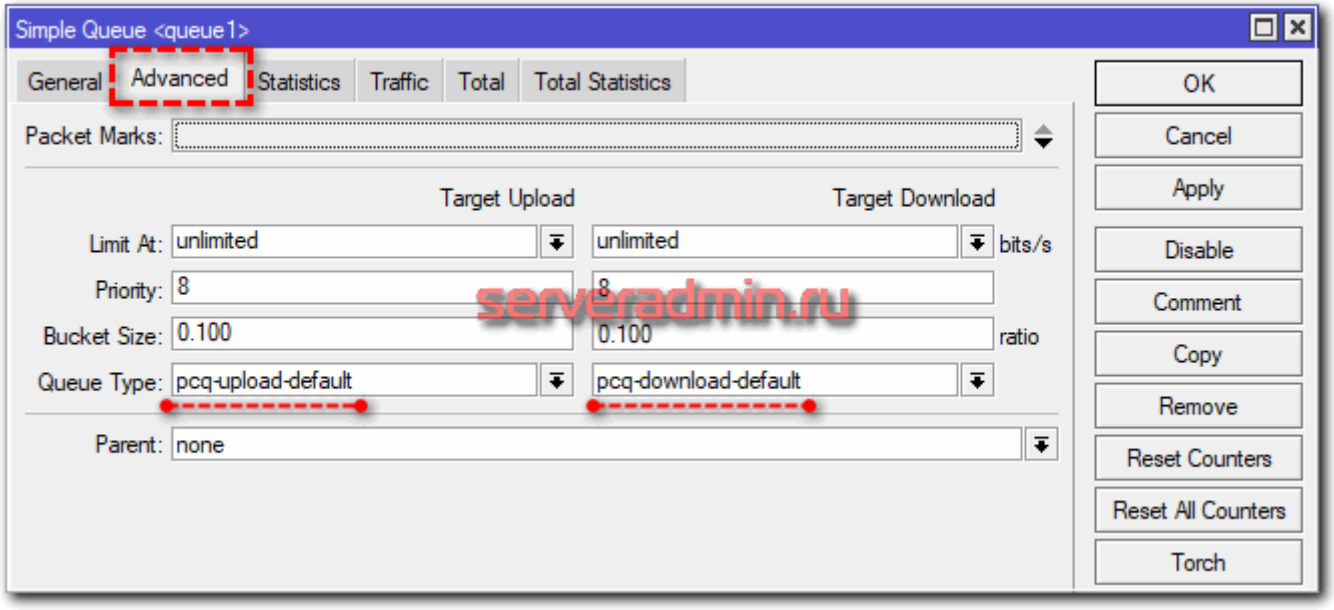


Я сначала запустил проверку скорости на ubuntu18, а через 10 секунд на centos7. Видно, что ubuntu сначала качала на полной скорости 100 Мбит/с, а потом, при запуске параллельного теста на centos, снизила скорость на треть. По факту получилось, что кто раньше запустил тест, тот и забрал себе больше канала.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

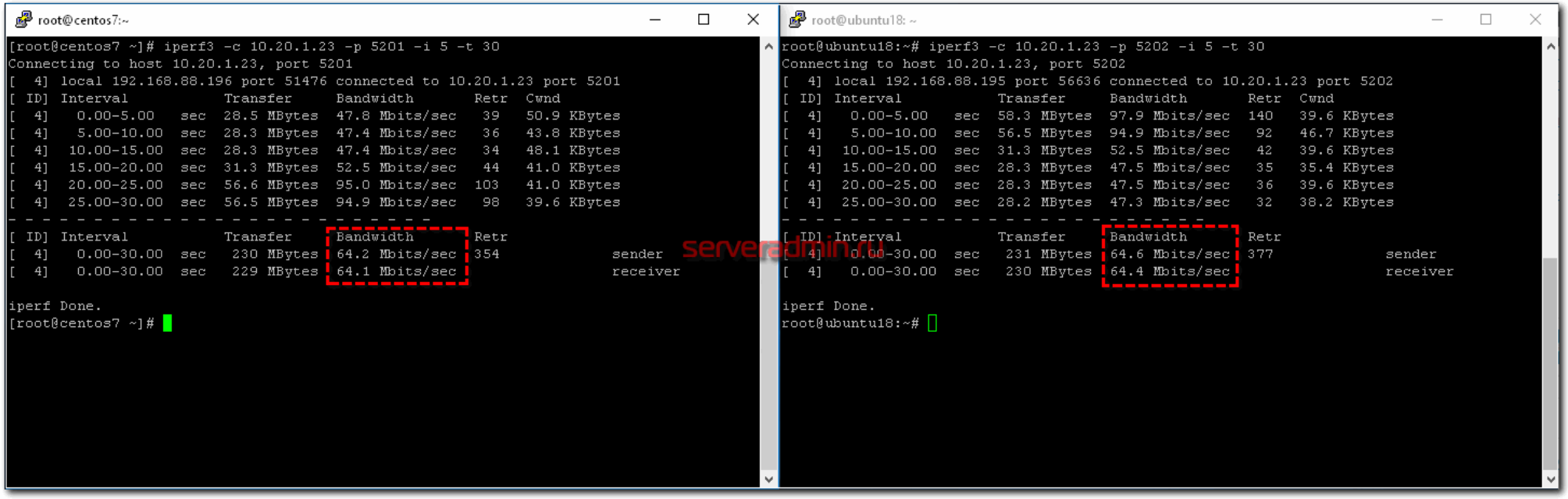
какой-нибудь торрент клиент, он бы забрал большую часть канала себе.

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





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.



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 pcq-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:

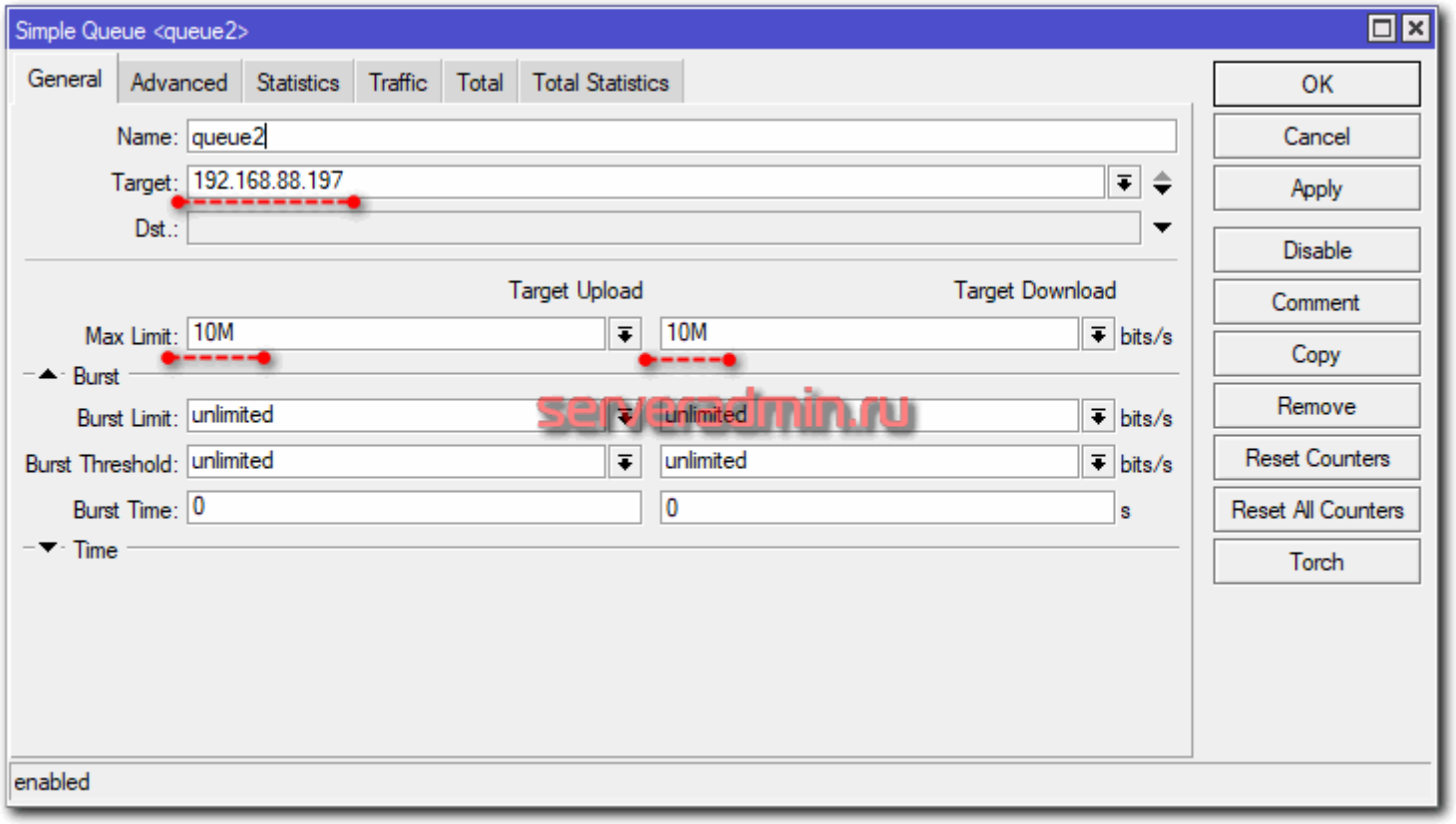
- max-limit = 20M;
- burst-threshold = 15M;
- burst-time = 16s;
- burst-limit = 50M.

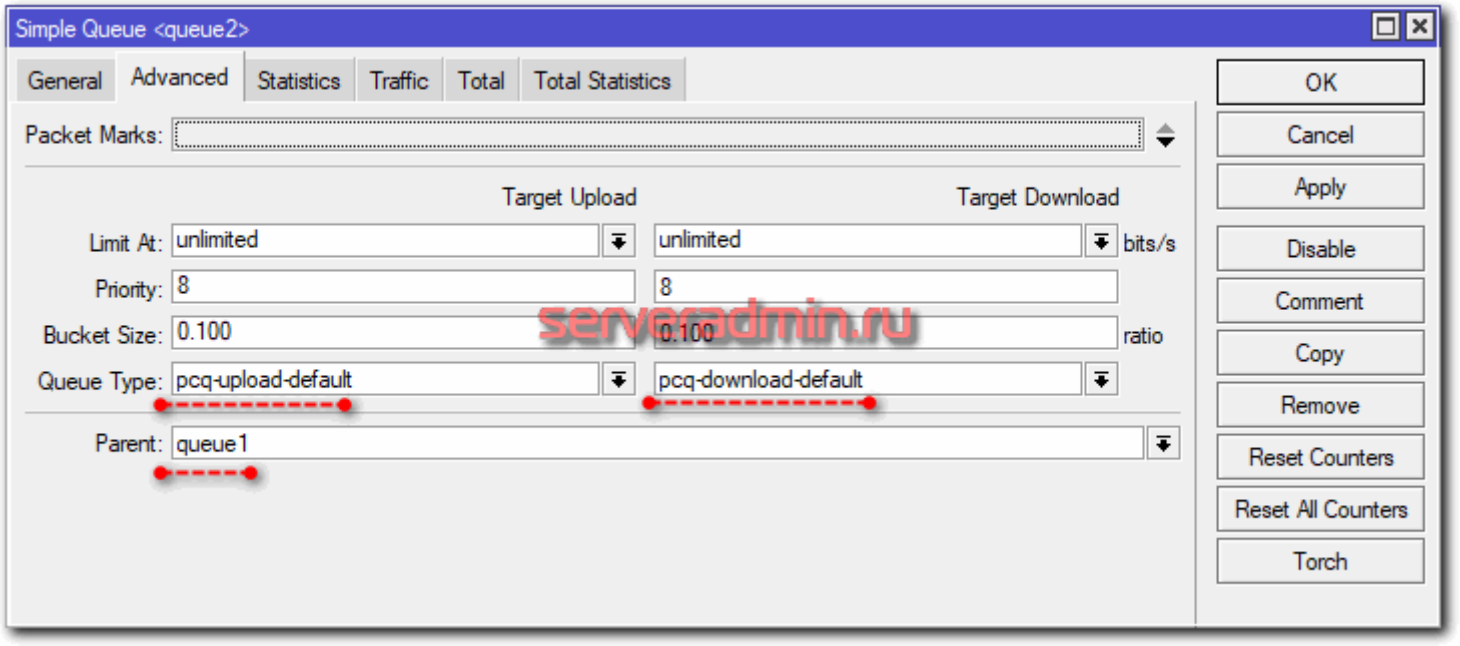
Вы начали качать большой файл. Первое время загрузка равна 50M (burst-limit), примерно на 5-й секунде средняя скорость за интервал в 16 (burst-time) секунд будет выше 15M (burst-threshold) (5 секунд на скорости 50M и 11 секунд 0M, пока вы еще ничего не качали, 50M*5сек + 0M*10сек)/16=15,625M, что больше 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*12сек + 0M*4сек)/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.

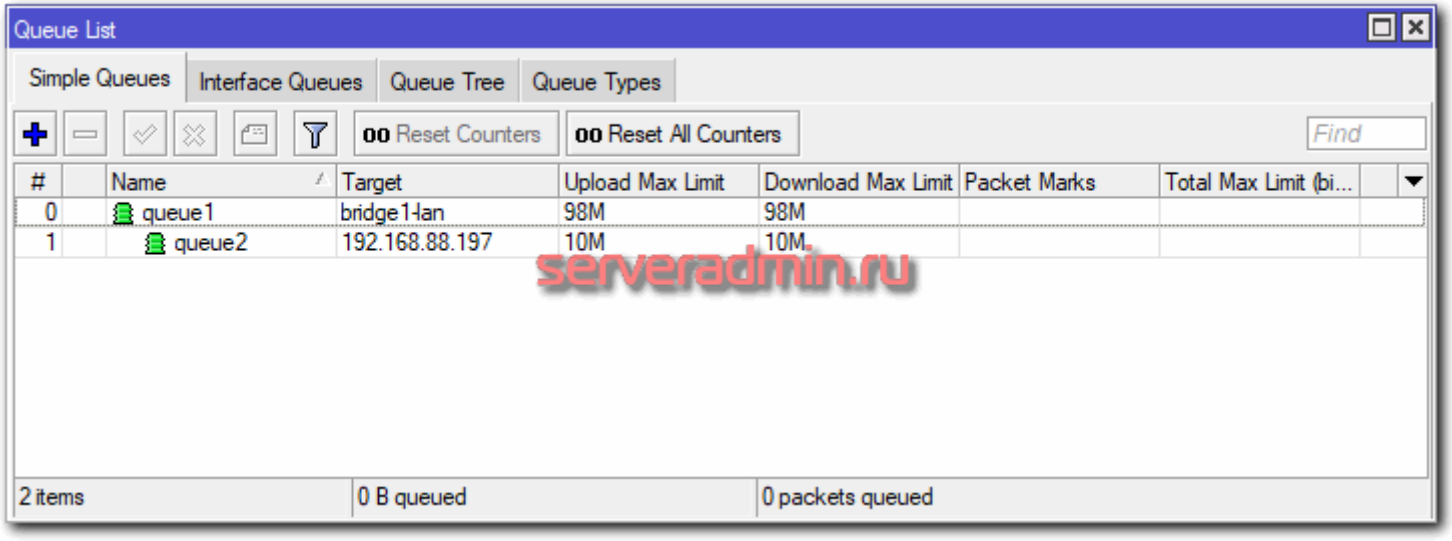
Ограничение скорости по IP

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.

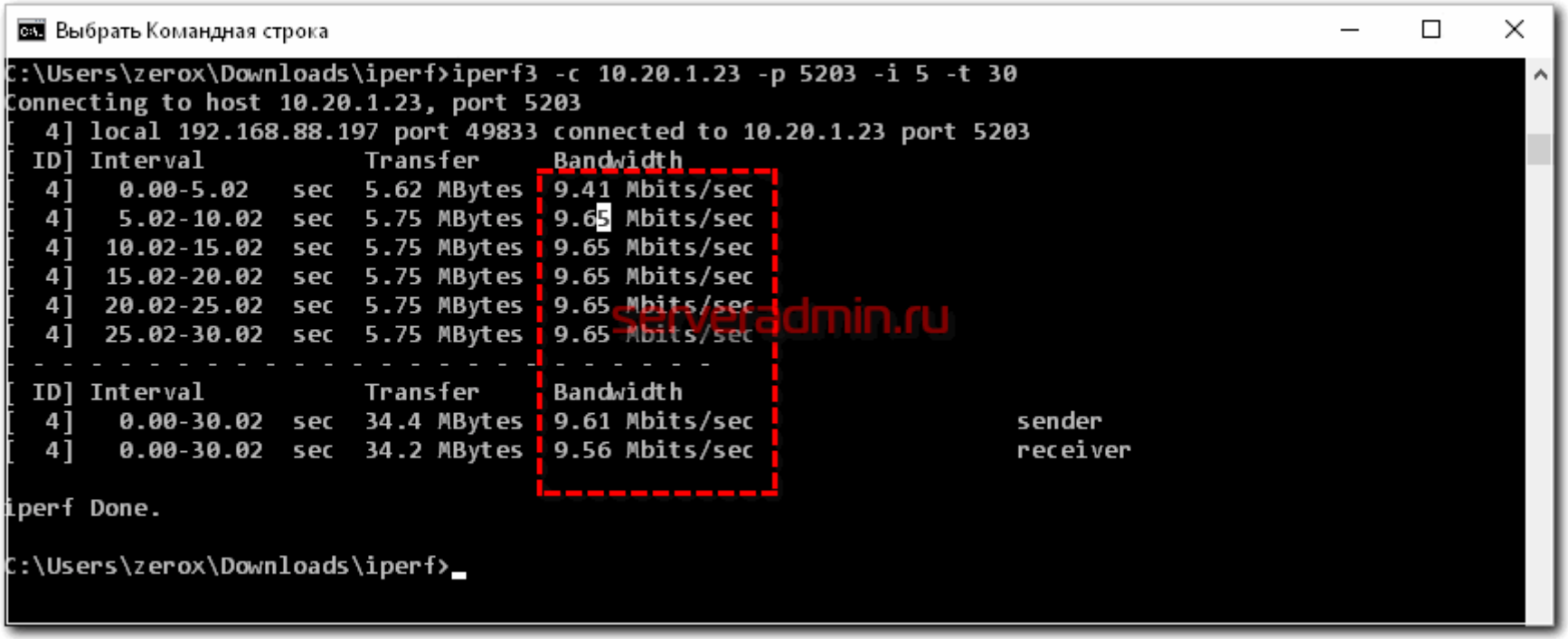




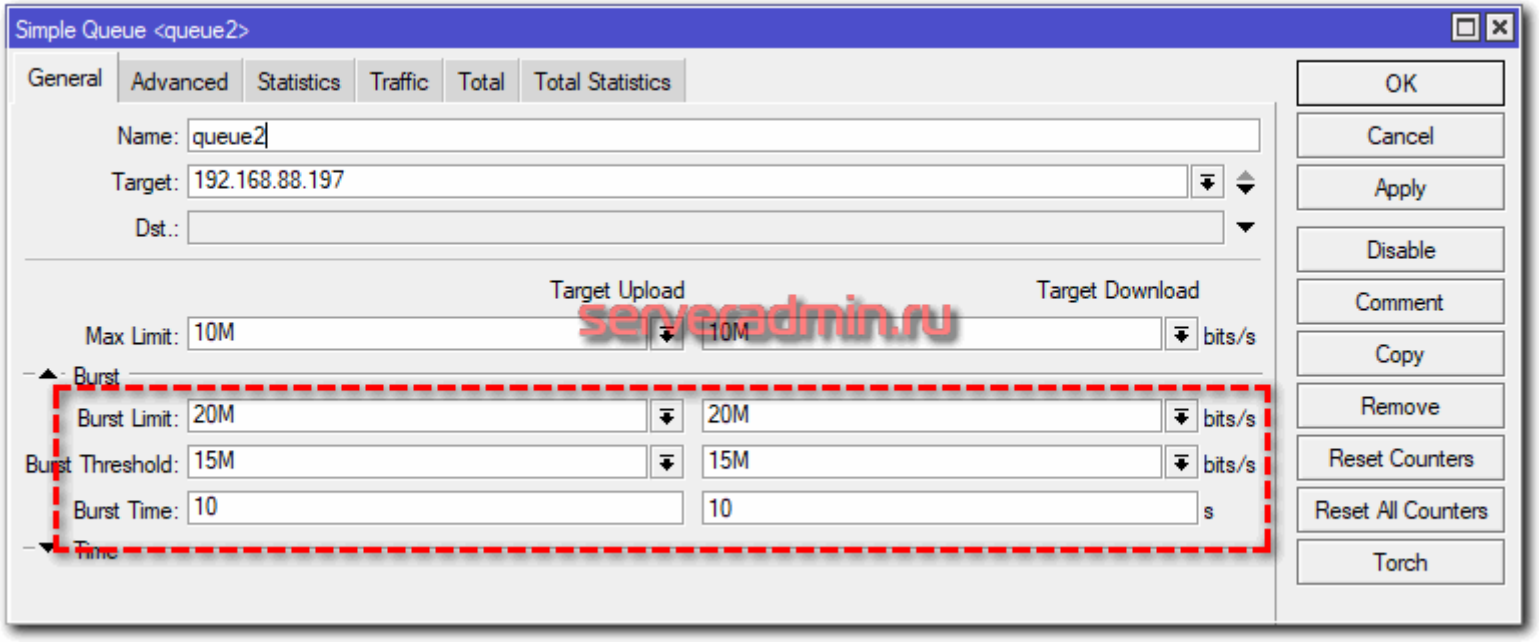
Должно получиться вот так, если отсортировать список по #.



Проверяем скорость доступа в интернет с помощью 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.



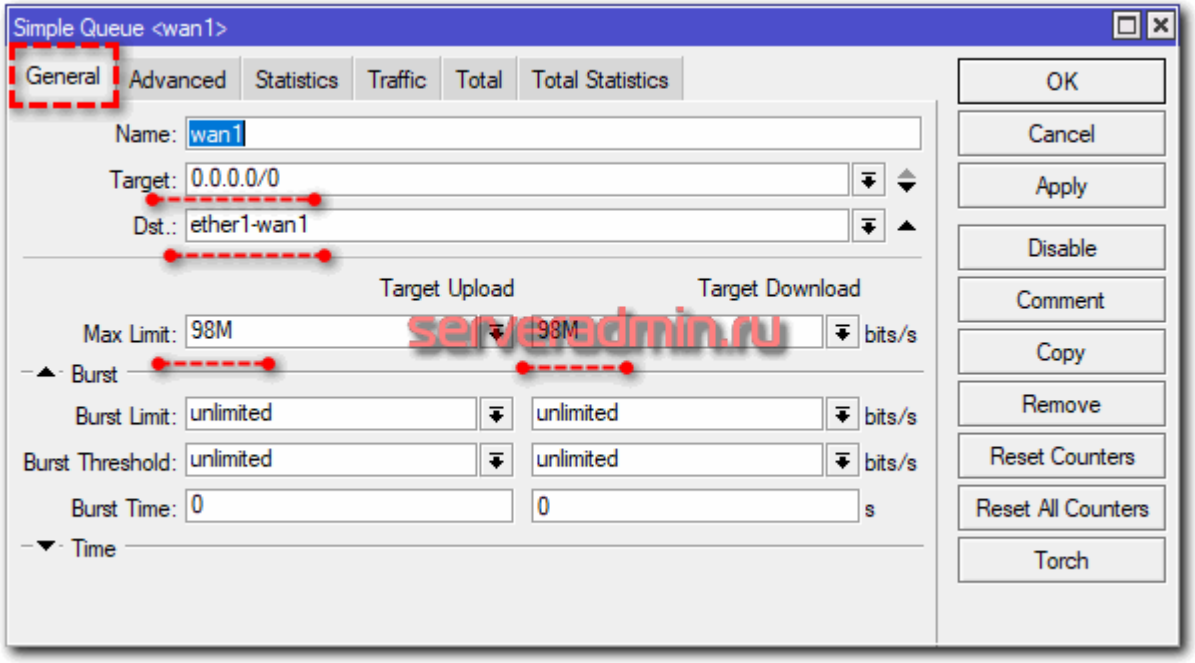
Смотрим, что получилось.

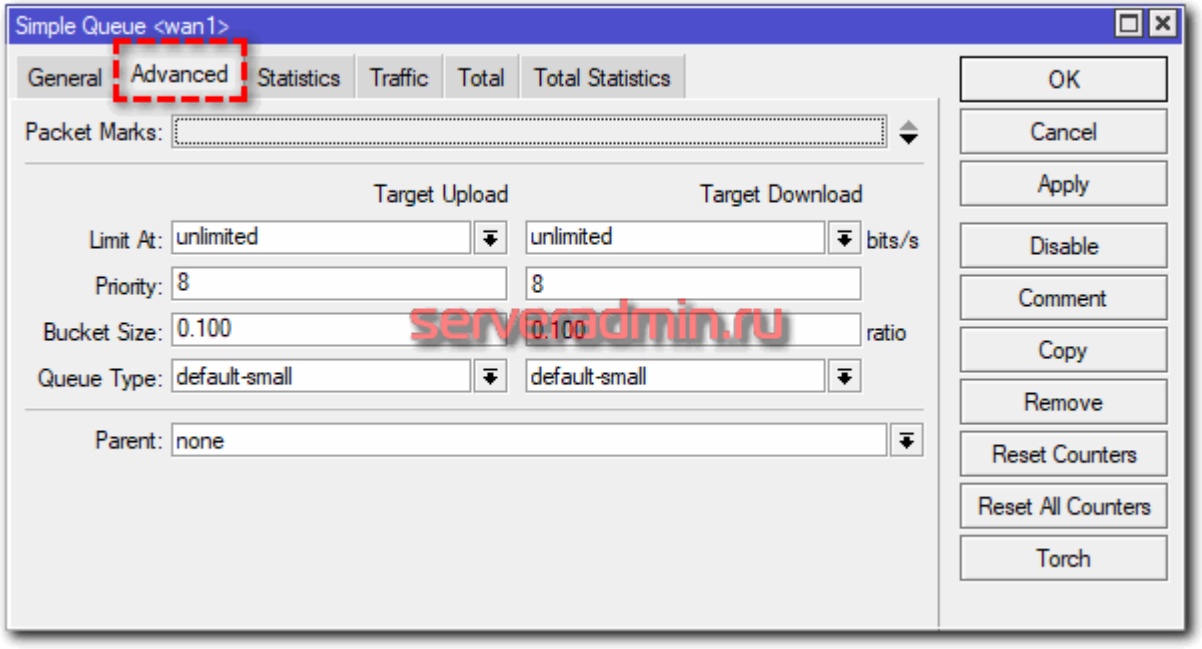
```
C:\Users\zerox\Downloads\iperf>iperf3 -c 10.20.1.23 -p 5203 -i 2 -t 30
Connecting to host 10.20.1.23, port 5203
[  4] local 192.168.88.197 port 49840 connected to 10.20.1.23 port 5203
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-2.02   sec  4.12 MBytes  17.2 Mbits/sec
[  4]   2.02-4.02   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.

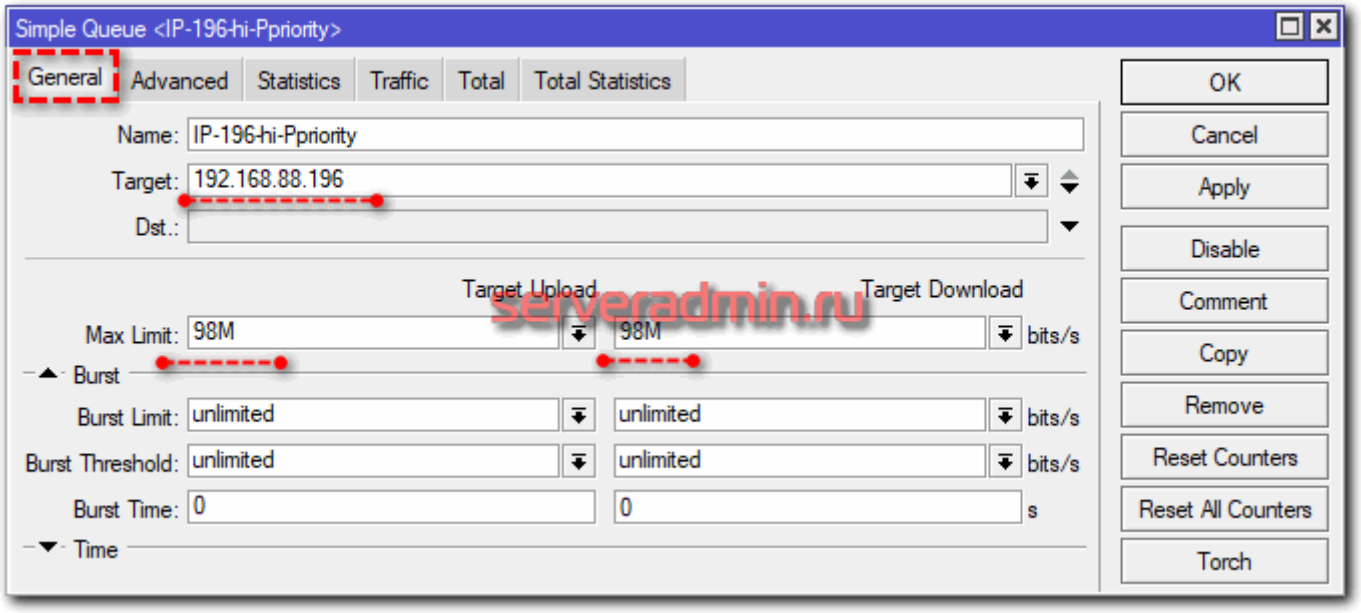
Приоритет трафика по IP

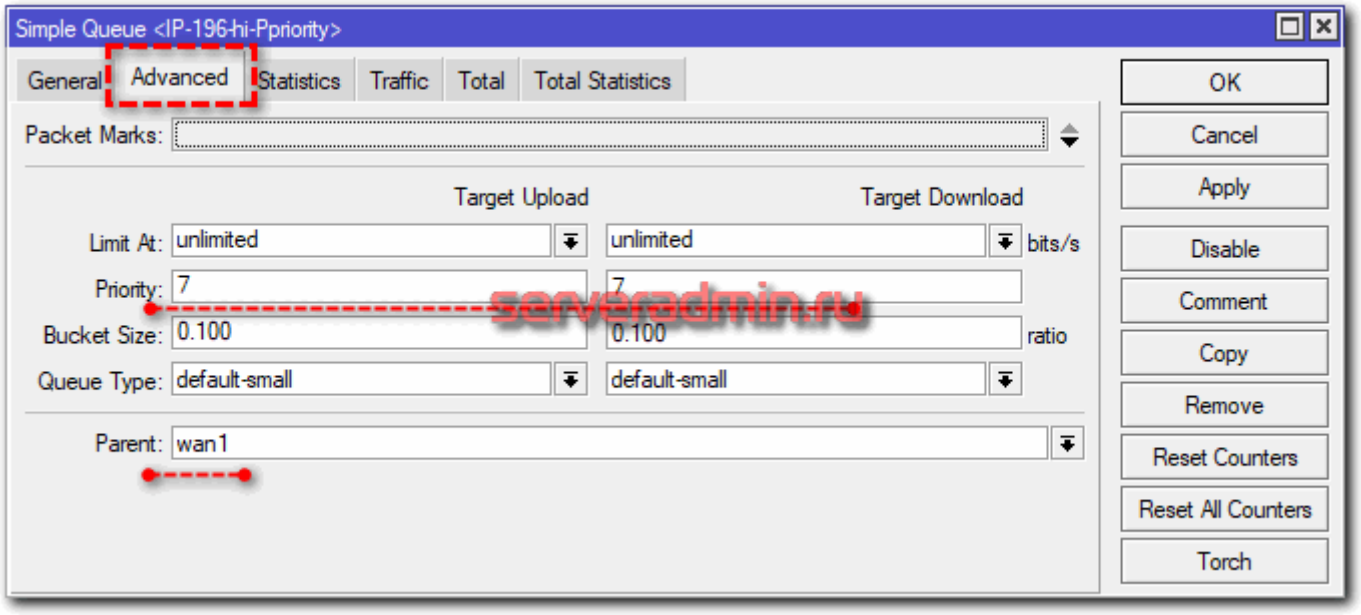
Продолжим настройку qos на основе очередей, создав несколько правил, одно из которых будет давать приоритет трафику от конкретного IP. Для этого создаем 3 simple queue - одна основная и две зависимые. Как я уже говорил ранее, приоритеты работают только в дочерних очередях, не родительских.


Simple Queue <wan1>
General
Advanced
Statistics
Traffic
Total
Total Statistics
Name: wan1
Target: 0.0.0.0/0
Dst.: ether1-wan1
Target Upload
Target Download
Max Limit: 98M
98M
bits/s
Burst
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
serveradmin.ru

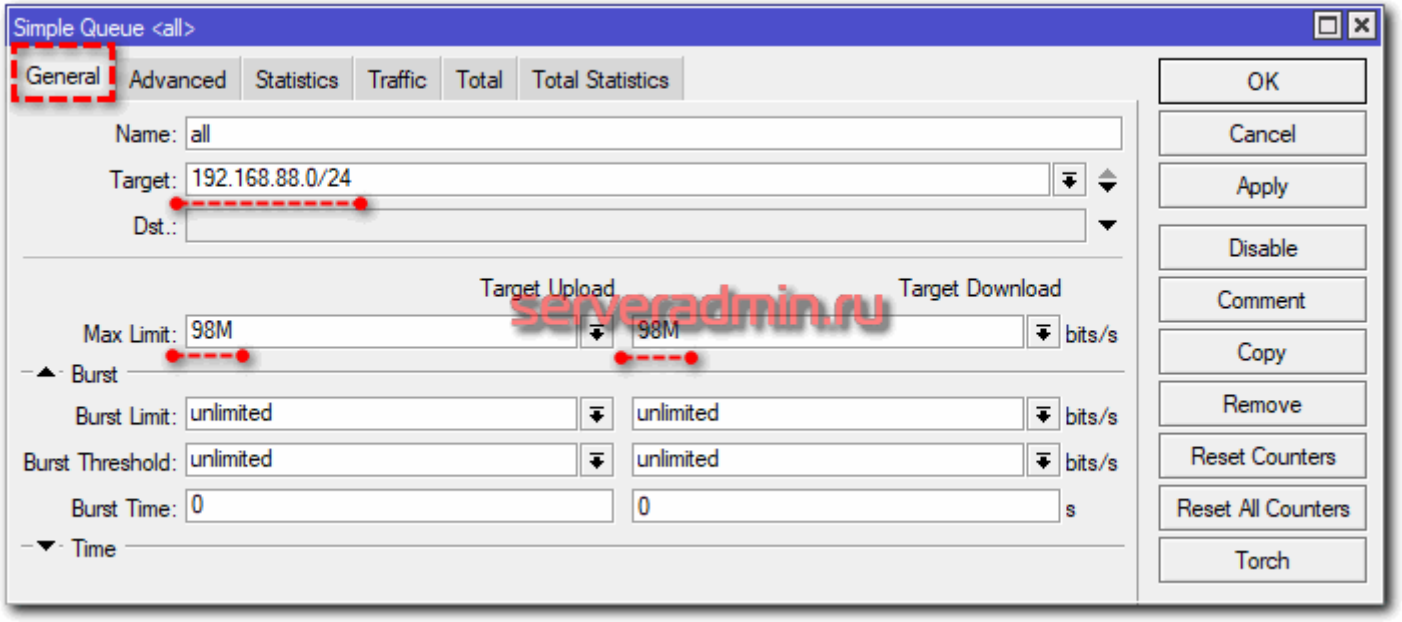


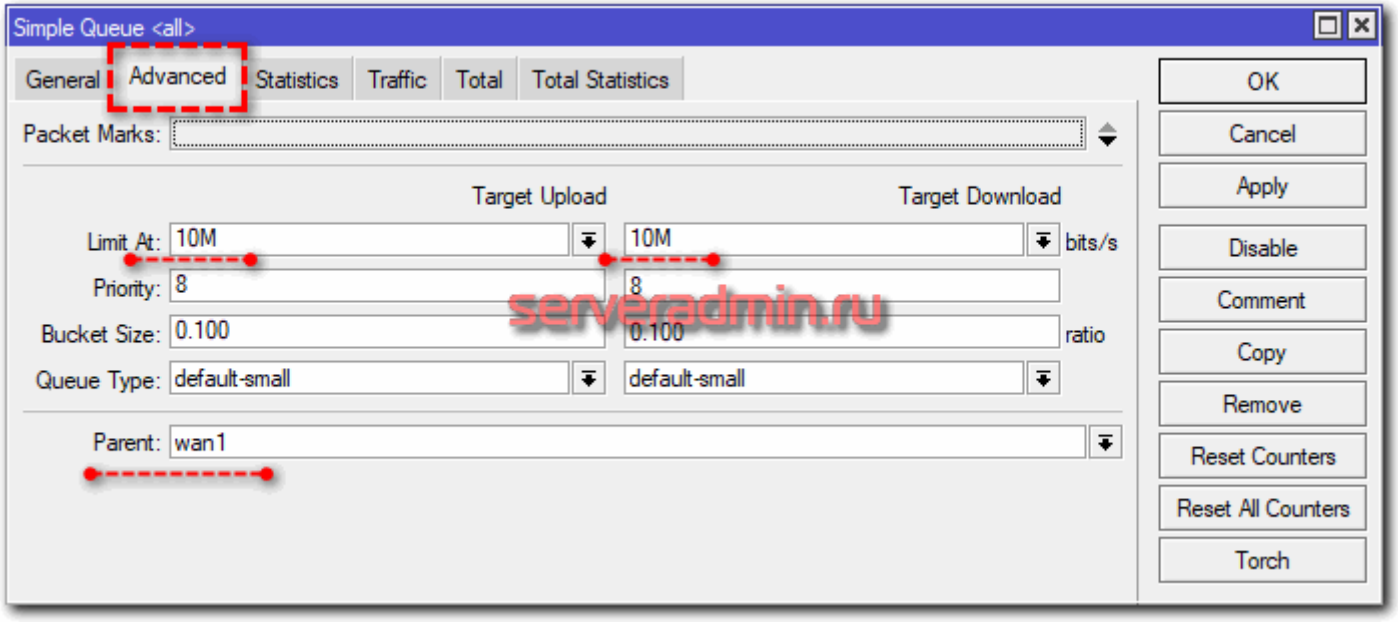
Общая родительская очередь. Дальше идут потомки.


Simple Queue <IP-196-hi-Ppriority>
General
Advanced
Statistics
Traffic
Total
Total Statistics
Name: IP-196-hi-Ppriority
Target: 192.168.88.196
Dst.:
Target Upload
Target Download
Max Limit: 98M 98M bits/s
Burst
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
serveradmin.ru



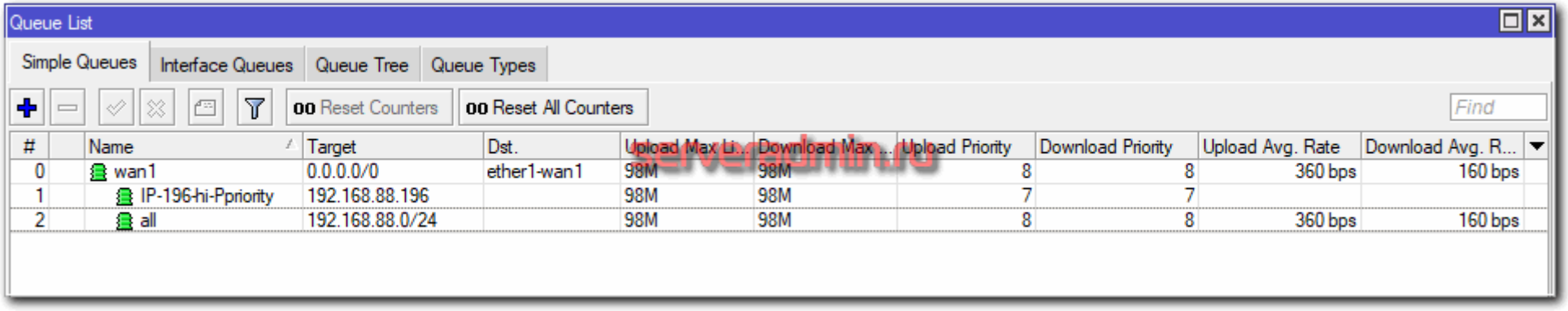
Указываем ip адрес источника, для которого увеличиваем приоритет, и выбираем родителя. Остальное все то же самое, что в первом правиле.


Simple Queue <all>
General
Advanced
Statistics
Traffic
Total
Total Statistics
Name: all
Target: 192.168.88.0/24
Dst.:
Target Upload
Target Download
Max Limit: 98M
98M
bits/s
Burst
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
serveradmin.ru



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.

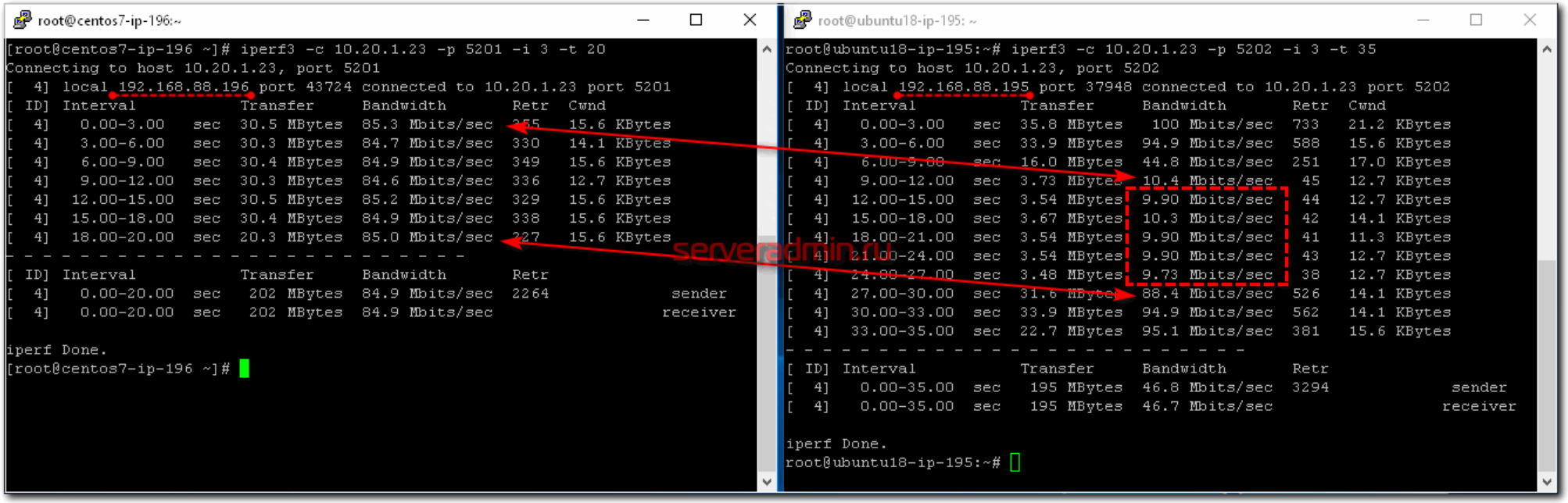
Должен получиться такой набор иерархически упорядоченных правил.




```
/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.

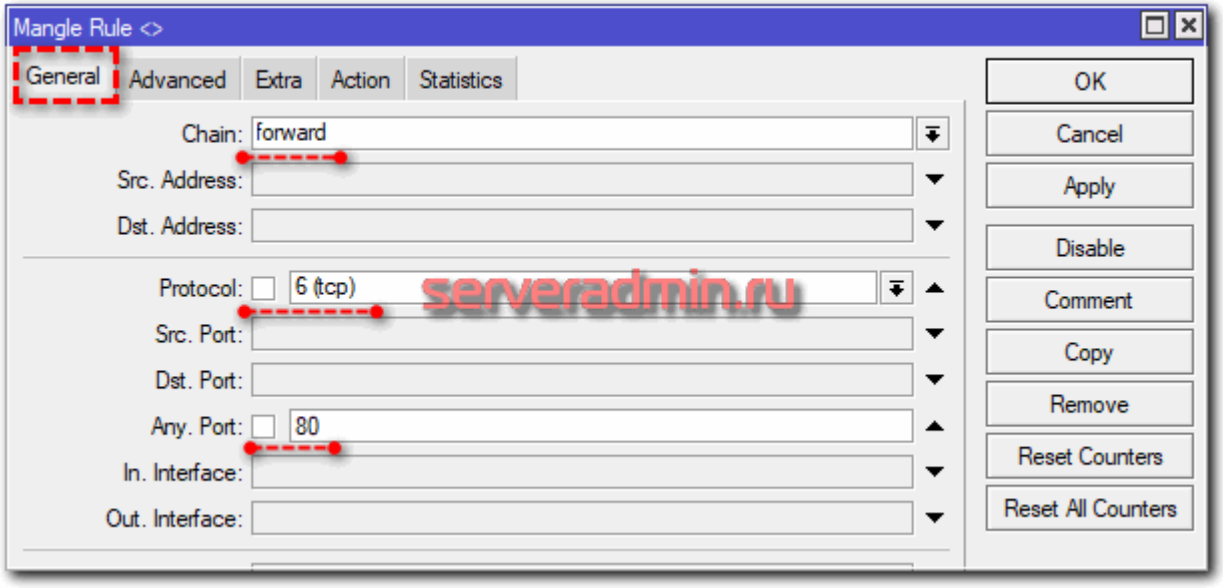


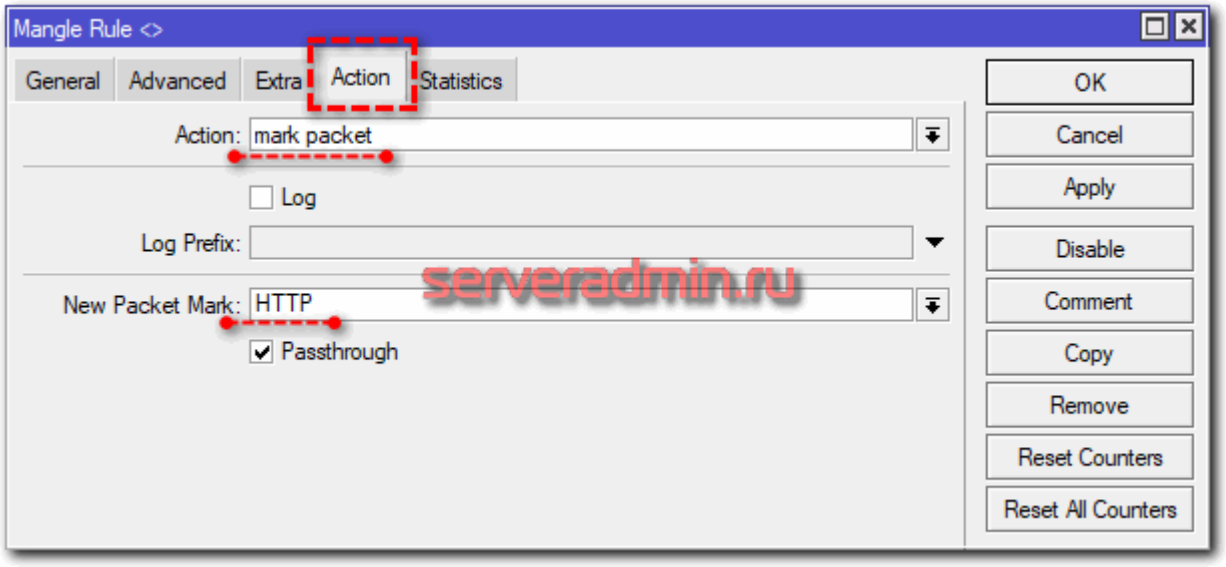
В итоге мы видим, что сначала была максимальная загрузка трафика на первой машине, потом приоритет был отдан второй, а у первой осталась скорость из значения limit-at. После того, как более приоритетный трафик кончился, вся доступная полоса пропускания опять досталась первому ip.

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.

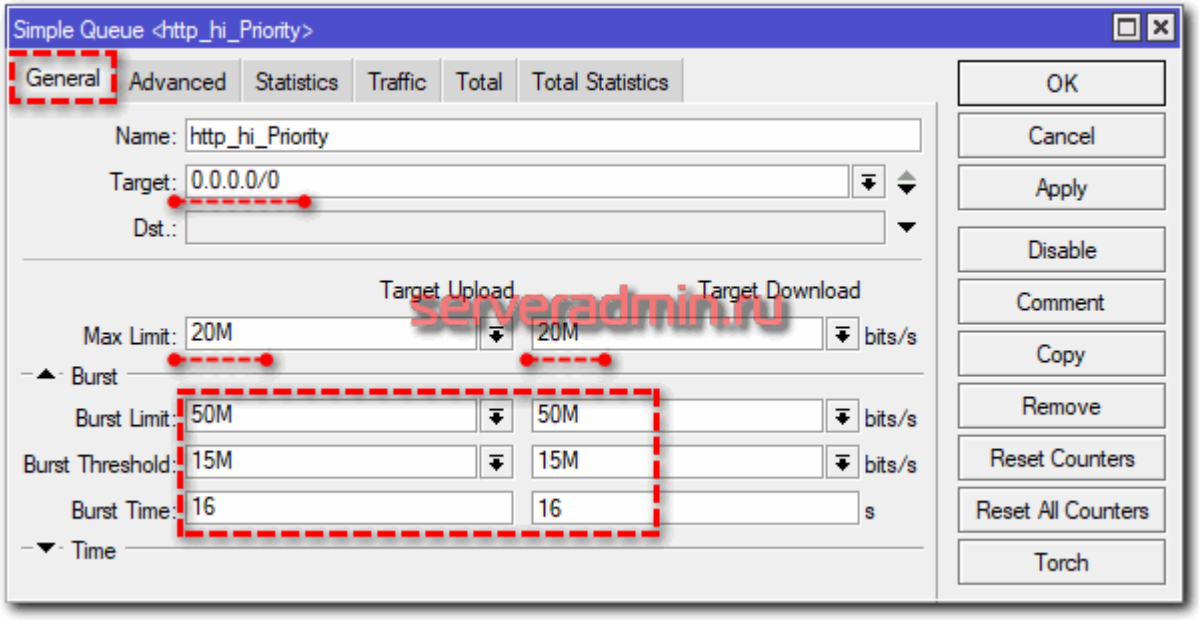

Mangle Rule <>
General
Advanced
Extra
Action
Statistics
Chain: forward
Src. Address:
Dst. Address:
Protocol: 6 (tcp)
Src. Port:
Dst. Port:
Any. Port: 80
In. Interface:
Out. Interface:
OK
Cancel
Apply
Disable
Comment
Copy
Remove
Reset Counters
Reset All Counters
serveradmin.ru

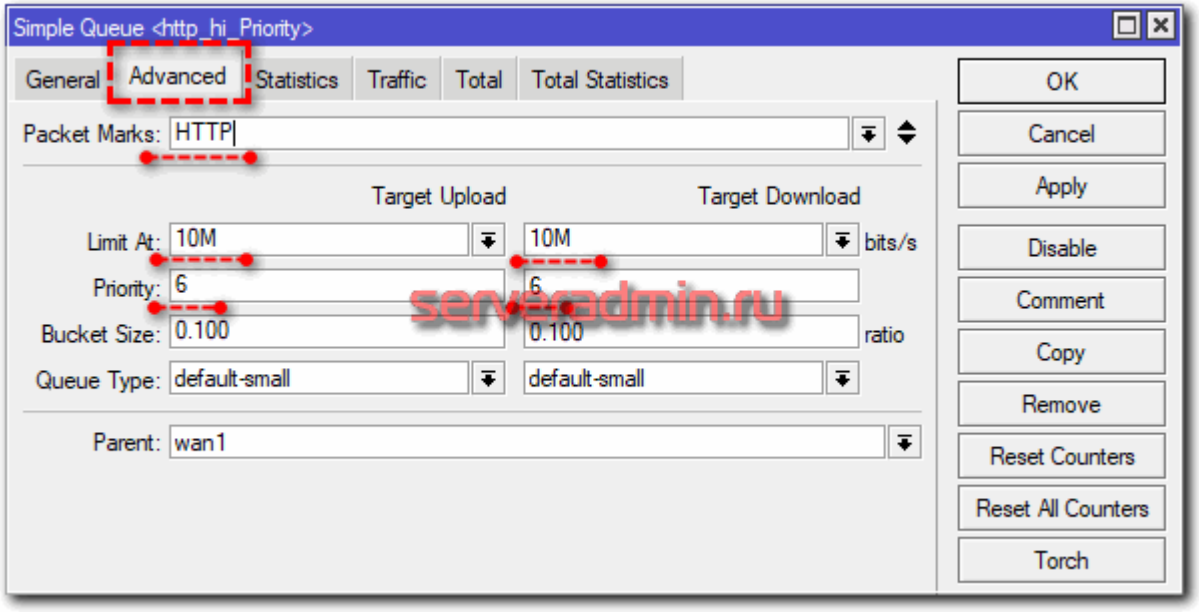


То же самое делаем для https трафика, просто указав порт 443. Вот оба правила.

```
/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
```

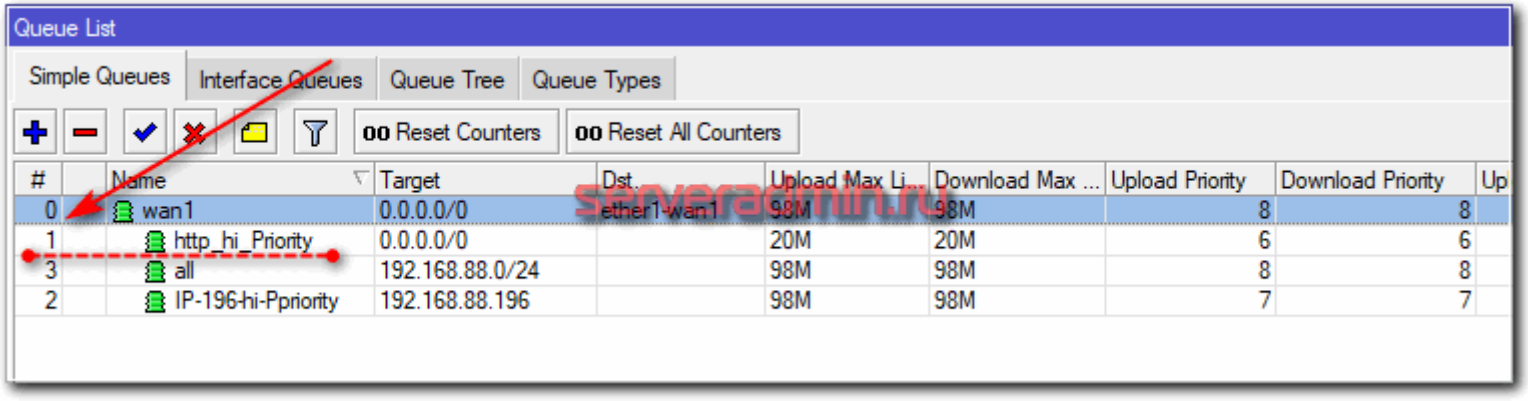
Теперь идем добавлять новую очередь с более высоким приоритетом для веб трафика.


Simple Queue <http_hi_Priority>
General
Advanced
Statistics
Traffic
Total
Total Statistics
OK
Cancel
Apply
Disable
Comment
Copy
Remove
Reset Counters
Reset All Counters
Torch
Name: http_hi_Priority
Target: 0.0.0.0/0
Dst.:
Target Upload
Target Download
Max Limit: 20M 20M bits/s
Burst
Burst Limit: 50M 50M bits/s
Burst Threshold: 15M 15M bits/s
Burst Time: 16 16 s
Time
serveradmin.ru



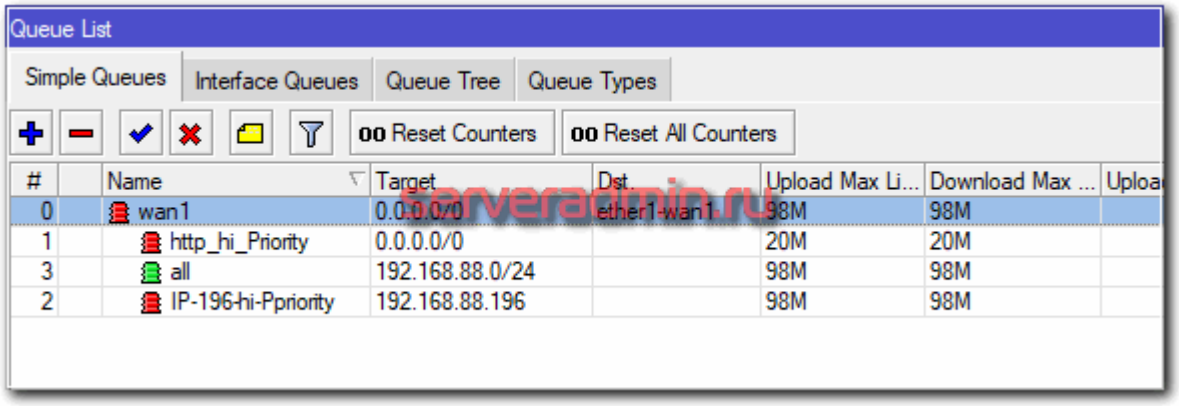

```
add burst-limit=50M/50M burst-threshold=15M/15M burst-time=16s/16s limit-at=10M/10M max-limit=20M/20M
name=http_hi_Priority packet-marks=HTTP parent=wan1 priority=6/6 target=""
```

В итоге имеем следующий набор правил:

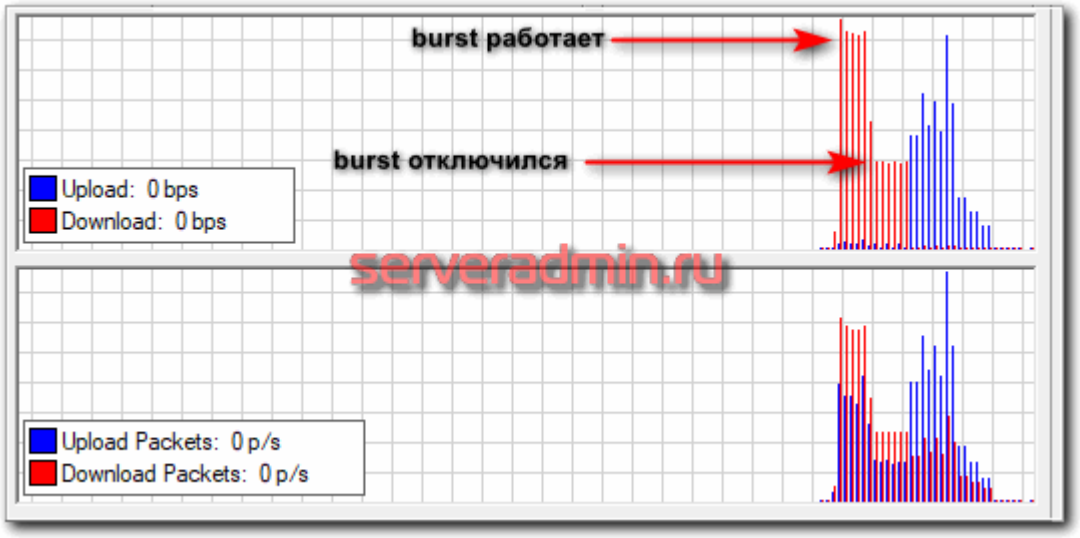


Не забываем про порядок правил в простых очередях. Он имеет значение, в отличие от queue tree.

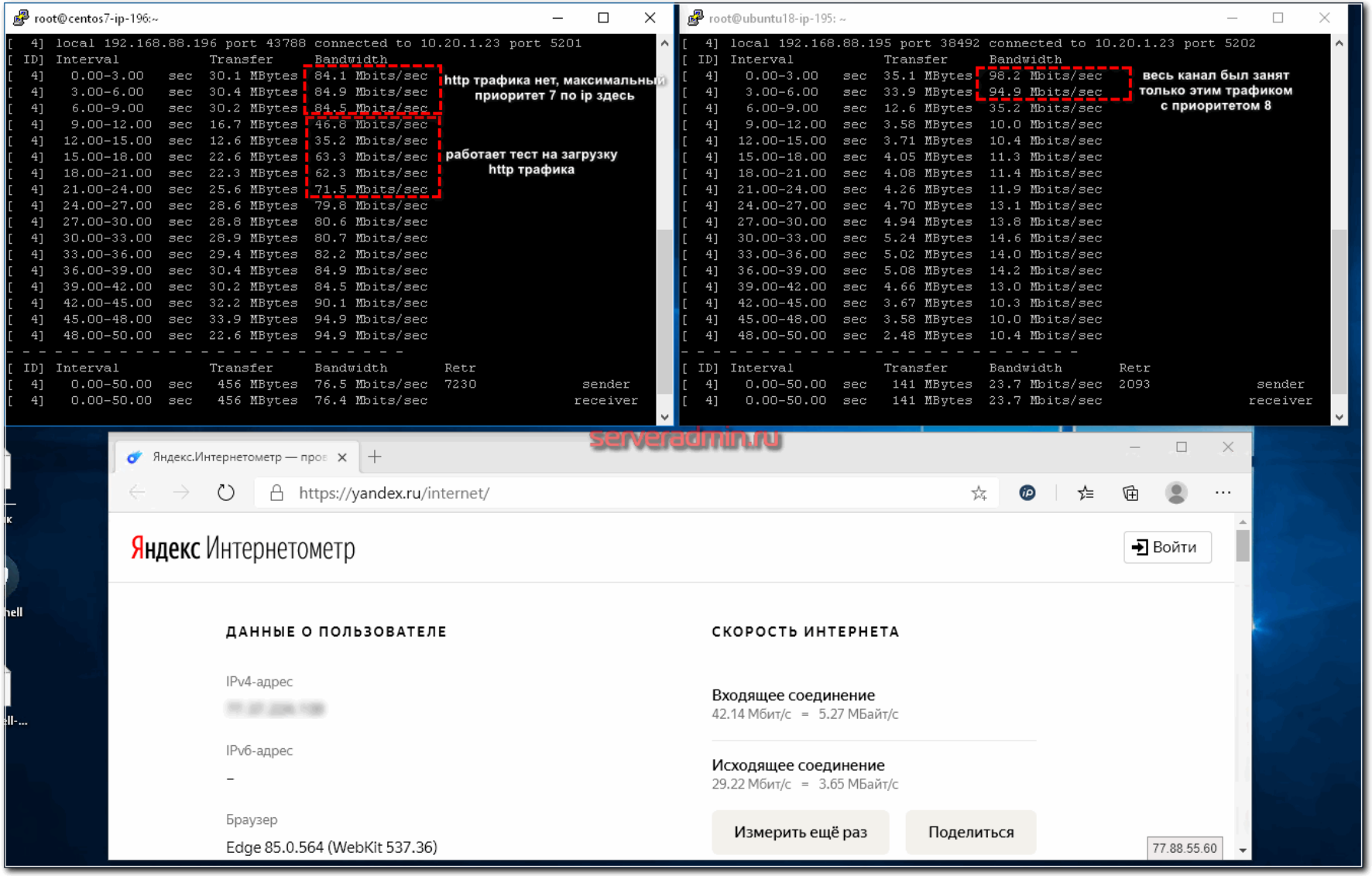
Теперь проверяем, насколько заметно и корректно в итоге работает приоритизация http трафика. Я, как обычно, запущу iperf на двух других машинах в сети и параллельно с этим запущу веб серфинг на еще одном компьютере. Вот какая картина будет в списке очередей.



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 бегает с заданными лимитами на ширину канала.


root@centos7-ip-196:~
[4] local 192.168.88.196 port 43788 connected to 10.20.1.23 port 5201
[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
[4] 0.00-50.00 sec 456 MBytes 76.5 Mbits/sec 7230 sender
[4] 0.00-50.00 sec 456 MBytes 76.4 Mbits/sec receiver
http трафика нет, максимальный приоритет 7 по ip здесь
работает тест на загрузку http трафика
root@ubuntu18-ip-195: ~
[4] local 192.168.88.195 port 38492 connected to 10.20.1.23 port 5202
[ID] Interval Transfer Bandwidth
[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
[4] 0.00-50.00 sec 141 MBytes 23.7 Mbits/sec 2093 sender
[4] 0.00-50.00 sec 141 MBytes 23.7 Mbits/sec receiver
весь канал был занят только этим трафиком с приоритетом 8
serveradmin.ru
Яндекс.Интернетометр — про
https://yandex.ru/internet/
Яндекс Интернетометр
Войти
ДАННЫЕ О ПОЛЬЗОВАТЕЛЕ
IPv4-адрес
IPv6-адрес
–
Браузер
Edge 85.0.564 (WebKit 537.36)
СКОРОСТЬ ИНТЕРНЕТА
Входящее соединение
42.14 Мбит/с = 5.27 МБайт/с
Исходящее соединение
29.22 Мбит/с = 3.65 МБайт/с
Измерить ещё раз
Поделиться
77.88.55.60

Я заметил небольшую особенность, причину которой не понял. Во время теста на ip 195, что на картинке справа, четко действует лимит в 10M, 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 3-4M и становится в районе 13-14M, 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 10M. С чем это связано, я не понял.

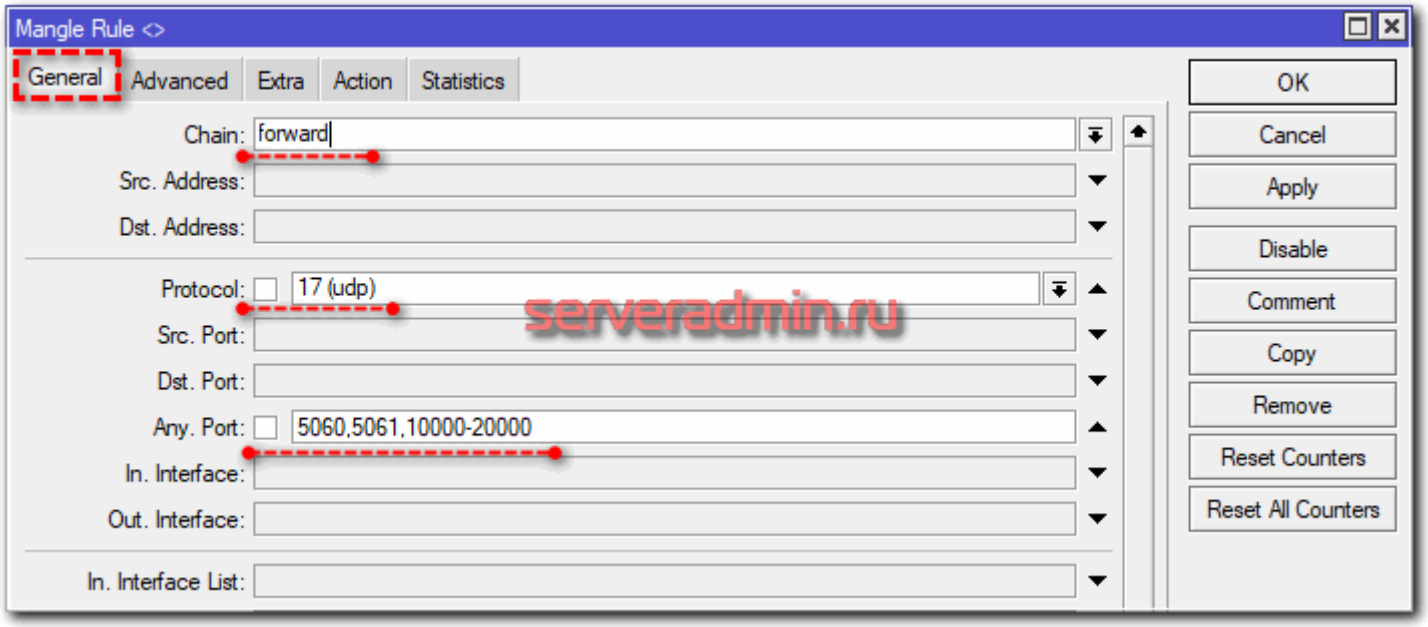
Надеюсь, общая идея понятна. По такому принципу можно маркировать любой трафик и встраивать в дерево очередей на основе маркировки.

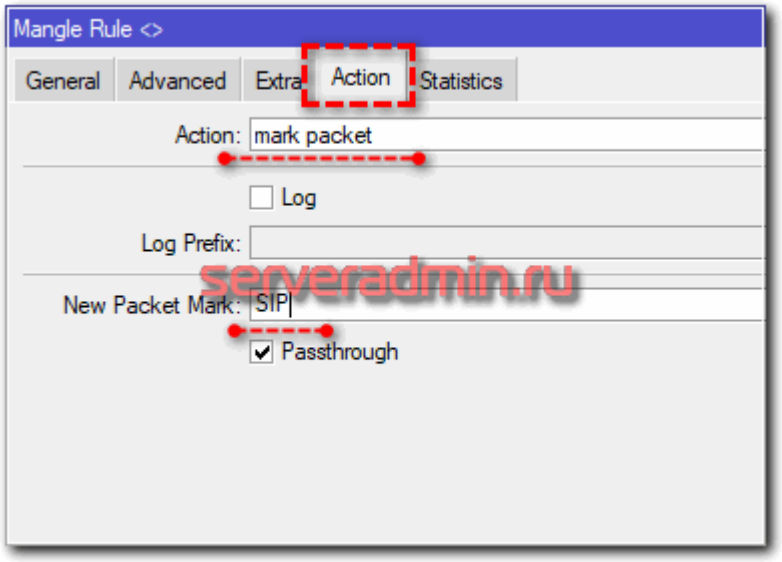
Приоритет SIP трафика для VOIP

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.

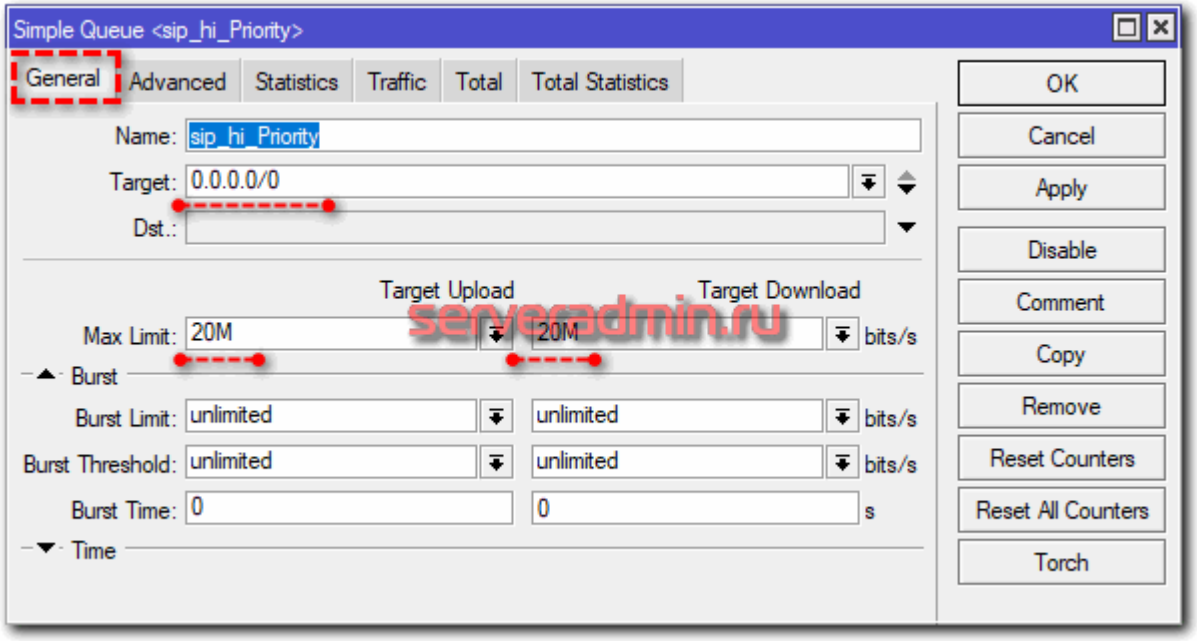
Так что в зависимости от своей ситуации, подбирайте реализацию приоритета конкретно под ваш случай. Я покажу пример второго способа с маркировкой. Снова идем в **IP** -> **Firewall** -> **Mangle** и добавляем правило маркировки для sip трафика.

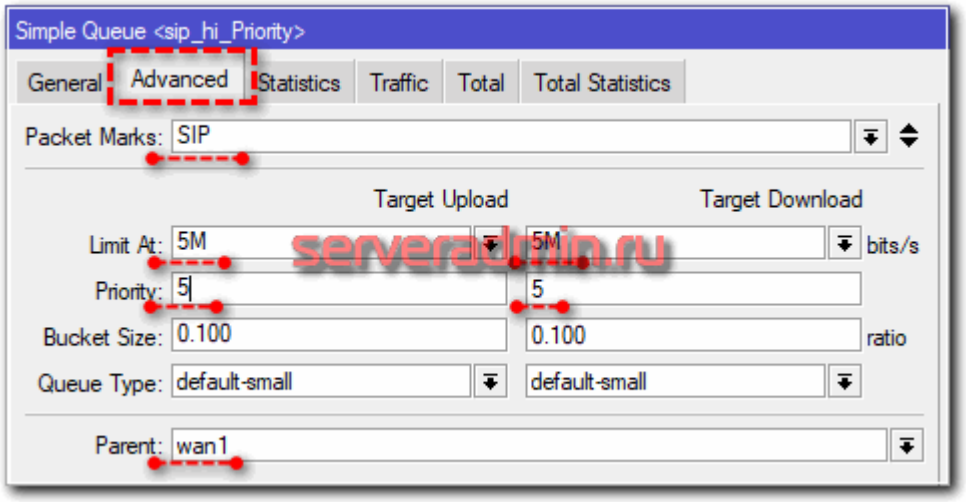

Mangle Rule <>
General
Advanced
Extra
Action
Statistics
Chain: forward
Src. Address:
Dst. Address:
Protocol: 17 (udp)
Src. Port:
Dst. Port:
Any. Port: 5060,5061,10000-20000
In. Interface:
Out. Interface:
In. Interface List:
OK
Cancel
Apply
Disable
Comment
Copy
Remove
Reset Counters
Reset All Counters
serveradmin.ru




```
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.


Simple Queue <sip_hi_Priority>
General
Advanced
Statistics
Traffic
Total
Total Statistics
Name: sip_hi_Priority
Target: 0.0.0.0/0
Dst.:
Target Upload
Target Download
Max Limit: 20M 20M bits/s
Burst
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
serveradmin.ru



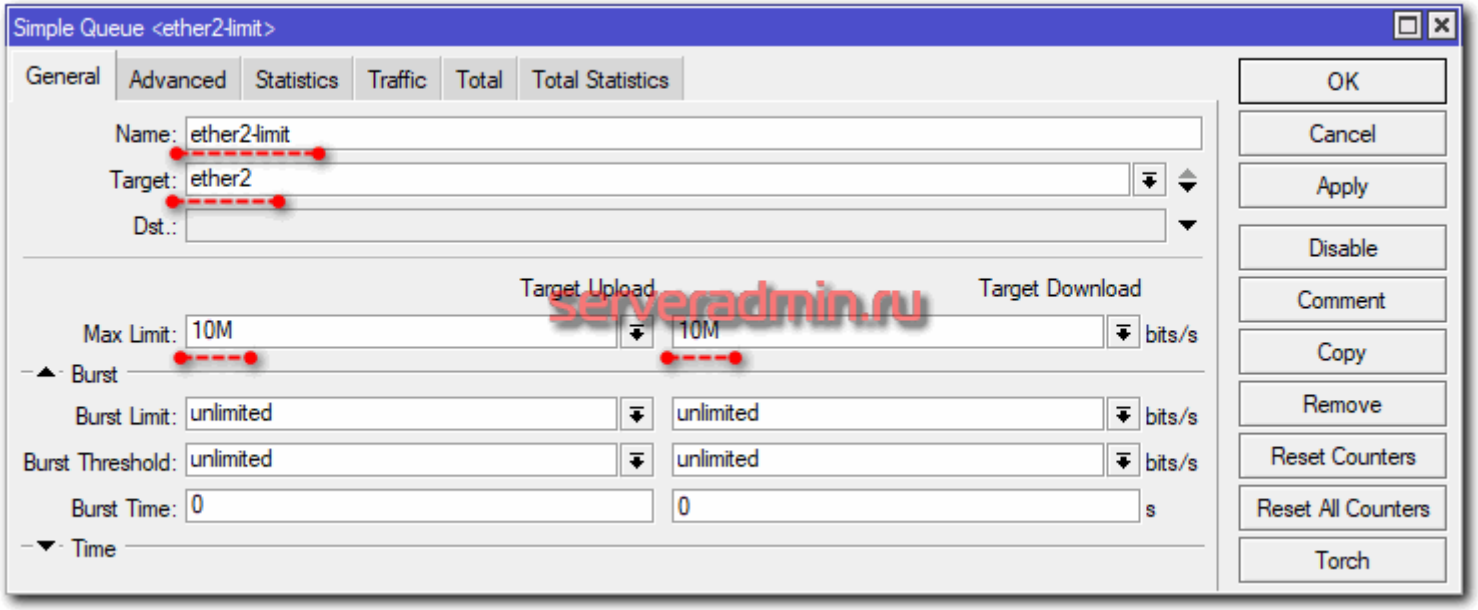
Думаю, что для sip трафика burst не нужен. Я не могу себе представить ситуацию, когда возможен какой-то резкий скачек трафика у отдельно взятого абонента. Не забывайте про порядок правил очередей. С более высоким приоритетом правила ставим выше.

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.

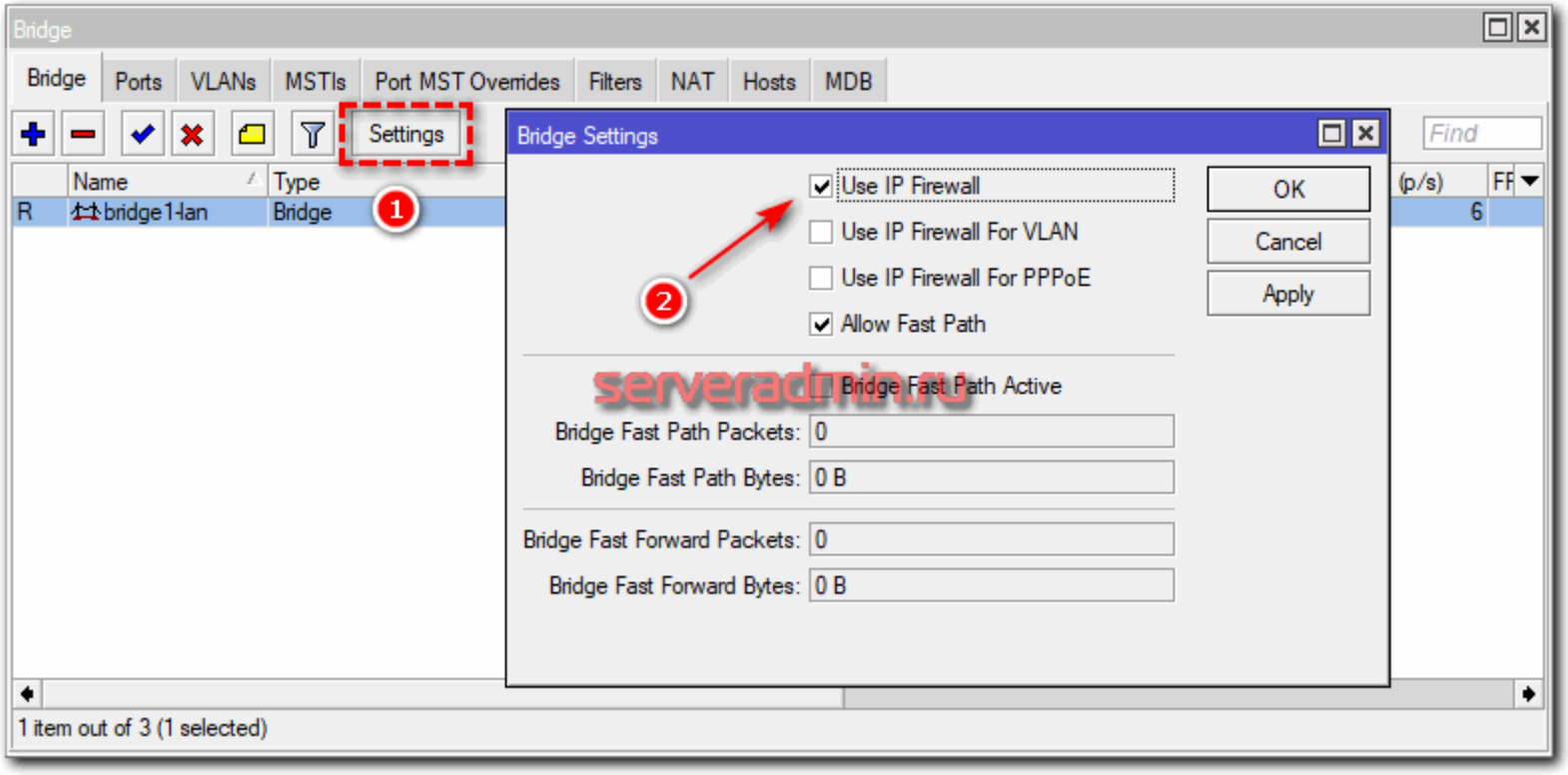
Ограничение скорости на интерфейсе

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.



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 твоем распоряжении. Расскажи, как сделать правильно!

На этом у меня по настройке qos в Микротик 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 приоритезации трафика удавалось решить возможностями Simple Queues.

Важное замечание по написанному. Я не являюсь большим специалистом по сетям и по мироктикам 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 Микротиков:

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

И многие другие в разделе, полностью посвящённому Mikrotik.

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.