

# Мikrotik и несколько провайдеров. Балансировка каналов

 [interface31.ru/tech\\_it/2023/04/mikrotik-i-neskol-ko-provayderov-balansirovka-kanalov.html](https://interface31.ru/tech_it/2023/04/mikrotik-i-neskol-ko-provayderov-balansirovka-kanalov.html)

## Записки IT специалиста

Технический блог специалистов ООО "Интерфейс"

- [Главная](#)
- Mikrotik и несколько провайдеров. Балансировка каналов

Как мы уже говорили, сегодня очень многие рабочие процессы прямо зависят от наличия доступа в интернет, поэтому несколько каналов доступа - это не прихоть и не роскошь, а насущная необходимость. Одной из первых задач, которые решаются несколькими каналами, является отказоустойчивость, но потом возникают иные вопросы, а именно полноценное использование двух каналов, ведь это совсем не дело если оплаченный резервный канал простаивает. Поэтому в данной статье мы разберем методы балансировки каналов на оборудовании Mikrotik и рассмотрим связанные с этим проблемы и способы их решения.



### Онлайн-курс по MikroTik

Научиться настраивать MikroTik с нуля или систематизировать уже имеющиеся знания можно на [углубленном курсе по администрированию MikroTik](#). Автор курса, сертифицированный тренер MikroTik Дмитрий Скоромнов, лично проверяет лабораторные работы и контролирует прогресс каждого своего студента. В три раза больше информации, чем в вендорской программе MTCNA, более 20 часов практики и доступ навсегда.

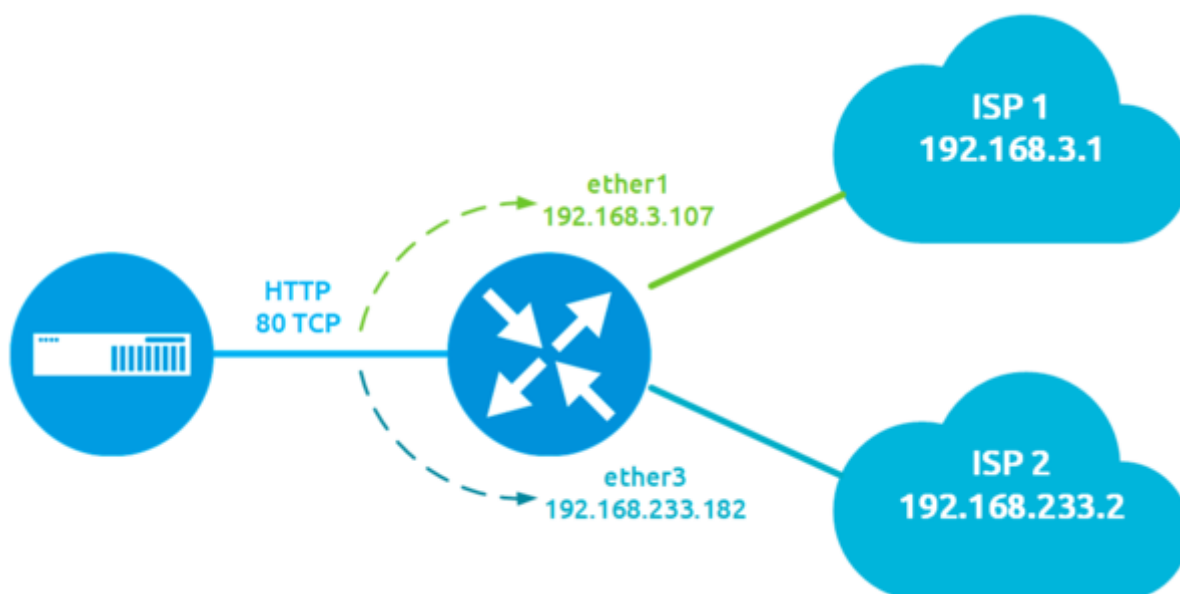
Сразу начнем с того, что балансировка и обеспечение отказоустойчивости - это две не связанные друг с другом задачи, хотя очень часто их рассматривают совместно, что только вносит дополнительную путаницу. Обработка отказа предусматривает переключение каналов при отказе одного из них, а балансировка распределяет трафик между ними. Поэтому и рассматривать мы их будем отдельно, хотя и коснемся немного вопроса отказоустойчивости, чтобы у вас было понимание как совместить одно с другим.

Тема балансировки тесно связана с маршрутизацией и маркировкой трафика, достаточно сложными для начинающих, поэтому мы советуем внимательно читать и пытаться понять прочитанное, бездумно повторить инструкции в данном случае скорее всего не получится. Также из-за сложности информации часть настроек мы

будем давать исключительно в виде консольных команд, особенно выполняющих повторяющиеся и второстепенные задачи. Будем считать, что вы знаете, как выполнить аналогичные действия в WinBox.

Данная статья предназначена для **RouterOS 6.x**

Во всех примерах ниже будет рассматриваться следующая схема, ее же мы использовали в предыдущей статье цикла. Но здесь во внутренней сети у нас появился дополнительный узел - условный веб-сервер, который должен быть доступен сразу на обоих внешних адресах.



А начнем мы с необходимого теоретического минимума, без которого настройка балансировки выльется в непонятные камлания с непредсказуемым результатом.

## Общие вопросы, проблемы и решения

---

Начнем с **соединений**, все пакеты установленного соединения должны проходить через один и тот же канал. Это аксиома. Поэтому балансировка должна выполняться не на уровне пакетов, а на уровне соединений. А так как соединения бывают разные и трафик по ним передается разный, то даже равномерно сбалансировав каналы по соединениям вы не получите равномерной балансировки по нагрузке. Это сразу следует понимать.

Второй момент, который связан с протоколами прикладного уровня, в рамках одного сеанса которых могут быть созданы несколько соединений. Если бездумно балансировать по соединениям, то может получиться так, что в рамках одного сеанса мы получим соединения с разными адресами источниками. В ряде случаев это не критично, а в других может доставить ряд существенных неудобств, особенно если на другой стороне существует привязка сессии к IP-адресу. Наиболее критично это

для финансовых приложений, например, онлайн-банков, в лучшем случае это приведет к постоянной необходимости заново аутентифицироваться в личном кабинете, в худшем - стать причиной блокировки доступа.

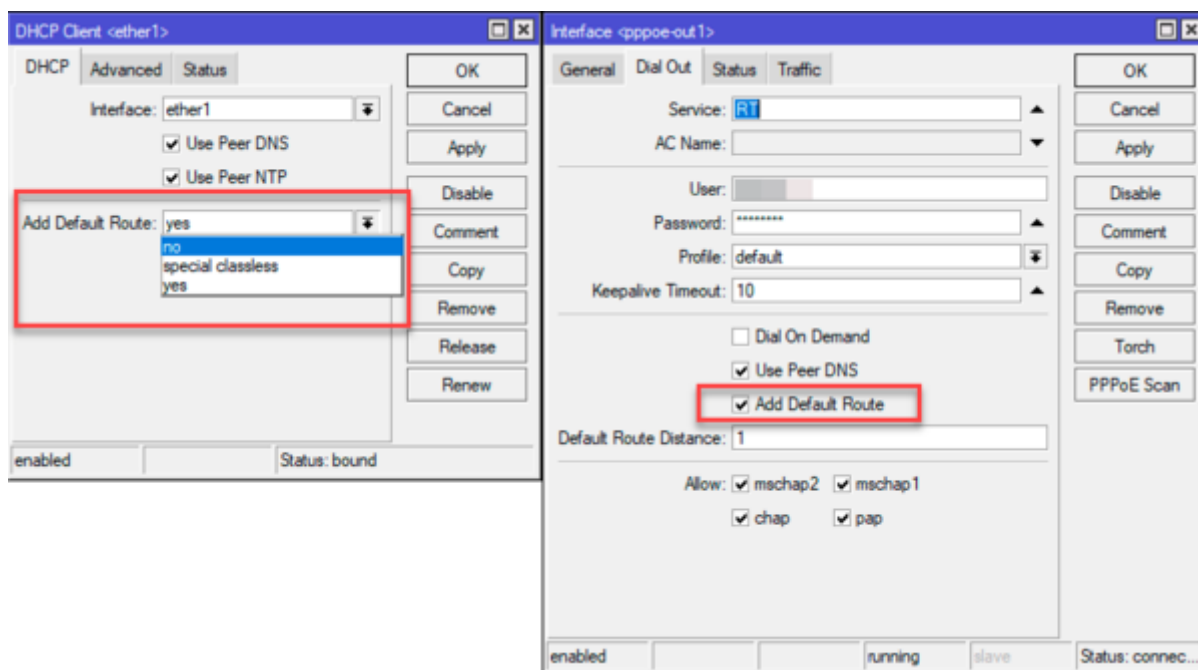
Также могут возникнуть проблемы с IPsec, туннелями, VPN и т.д., т.е. со всем тем, что чувствительно к IP-адресу противоположного узла.

Для того, чтобы этого избежать следует привязать балансировку к еще одному параметру - адресу источника и направлять все соединения одного узла в текущий промежуток времени через один и тот-же канал.

Следующая сложность - **проброс портов и собственные сервисы роутера**. Мы всегда должны отправлять ответ на тот интерфейс, с которого пришел запрос, а, следовательно должны исключить такие соединения из балансировки. Это решается дополнительной маркировкой и всегда должно делаться в первую очередь.

## Предварительная настройка роутера

Прежде всего вам потребуется отключить динамическое добавление маршрутов, которое используется при создании коммутируемых подключений, например, PPPoE или DHCP:



**Важно!** Во время выполнения операций с настройкой маршрутов у вас кратковременно пропадет доступ в интернет. Поэтому все указанные действия следует выполнять **имея физический доступ** к устройству.

Если вы все-таки работаете удаленно, то прежде, чем отключать динамические маршруты создайте нужные таблицы маршрутизации вручную (о чем будет разговор ниже), но помните, что любая ошибка может лишить вас доступа к устройству.

Затем создайте второе правило маскардинга или SNAT в разделе **IP - Firewall - NAT** для выхода в интернет через второй интерфейс, а также продублируйте для него все правила проброса портов.

Примерно это будет выглядеть так для маскардинга, где **192.168.111.0/24** - диапазон внутренней сети:

```
/ip firewall nat
add action=masquerade chain=srcnat out-interface=ether1 src-address=192.168.111.0/24
add action=masquerade chain=srcnat out-interface=ether3 src-address=192.168.111.0/24
```

И так для проброса портов, в данном случае **192.168.111.152** внутренний адрес условного веб-сервера:

```
/ip firewall nat
add action=dst-nat chain=dstnat dst-address=192.168.3.107 dst-port=80 \
    in-interface=ether1 protocol=tcp to-addresses=192.168.111.152 to-ports=80
add action=dst-nat chain=dstnat dst-address=192.168.233.182 dst-port=80 \
    in-interface=ether3 protocol=tcp to-addresses=192.168.111.152 to-ports=80
```

Отличия в правилах мы выделили, как видно правила маскардинга отличаются только интерфейсом выхода, а правила проброса портов - адресом назначения пакетов, в качестве которых используются внешние адреса роутера. В Winbox можете просто скопировать текущее правила и изменить в них указанные критерии.

## Таблицы маршрутизации

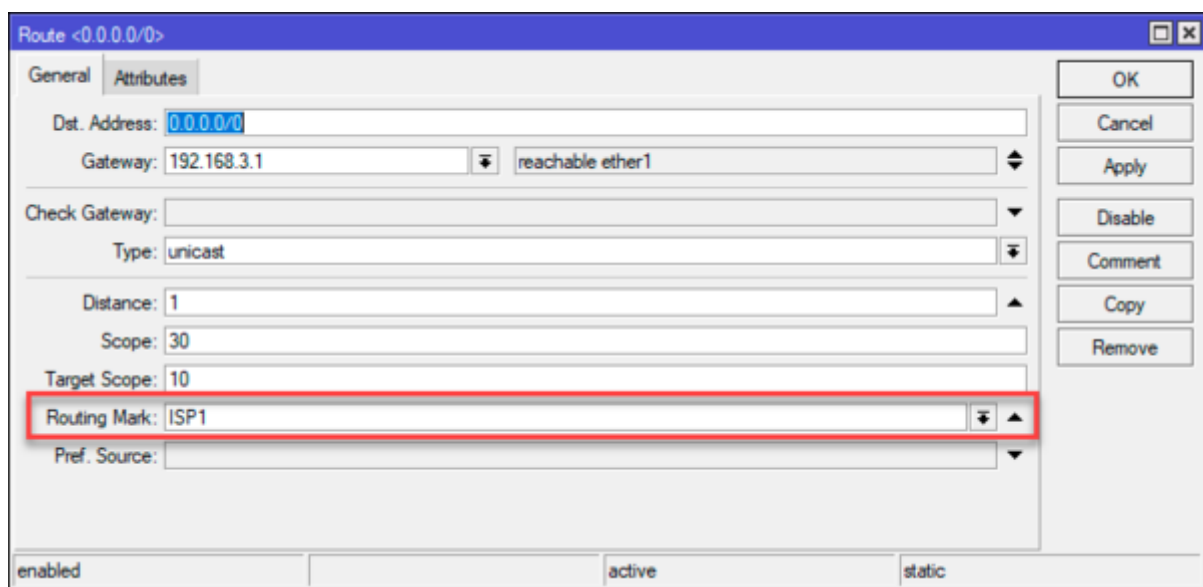
Все указанные ниже действия мы будем делать в **IP - Routes**, перейдем туда и сначала заполним **основную (main)** таблицу маршрутизации, которая будет использоваться по умолчанию, если никакая иная таблица не указана. Добавим маршрут к первому провайдеру: **Dst. Address** - оставляем по умолчанию - **0.0.0.0/0**, **Gateway** - указываем шлюз первого провайдера, в нашем случае **192.168.3.1** и устанавливаем административную дистанцию маршрута - **Distance** - **1**.

В терминале это сделать еще проще:

```
/ip route  
add distance=1 gateway=192.168.3.1
```

В принципе этого уже достаточно для того, чтобы наша сеть продолжила нормально выходить в интернет после того, как вы отключите динамическое добавление маршрутов.

Затем создадим таблицы маршрутизации для каждого из провайдеров. Это делается точно также, как и для основной, но с обязательным указанием имени таблицы в поле **Routing Mark**, мы не будем придумывать ничего сложного и просто назовем наши таблицы **ISP1** и **ISP2**. В качестве шлюзов - **Gateway** - указываем шлюзы соответствующих провайдеров.



Либо выполните в терминале:

```
/ip route  
add check-gateway=ping distance=1 gateway=192.168.3.1 routing-mark=ISP1  
add check-gateway=ping distance=1 gateway=192.168.233.2 routing-mark=ISP2
```

В результате мы имеем теперь три таблицы маршрутизации: основную - через нее будут работать все соединения по умолчанию, и две дополнительных, куда мы будем направлять маркированный трафик.

А как с отказоустойчивостью? А ее настраиваем любым желательным способом отдельно для каждой таблицы маршрутизации. Для этого можете воспользоваться нашей статьей:

[Mikrotik и несколько провайдеров. Резервирование каналов](#)

Например, самый простейший способ на основе дистанции маршрутов будет выглядеть так:

```
/ip route
add check-gateway=ping distance=1 gateway=192.168.3.1
addcheck-gateway=ping distance=2 gateway=192.168.233.2
```

```
addcheck-gateway=ping distance=1 gateway=192.168.3.1 routing-mark=ISP1
add check-gateway=ping distance=2 gateway=192.168.233.2 routing-mark=ISP1
```

```
add check-gateway=ping distance=1 gateway=192.168.233.2 routing-mark=ISP2
add check-gateway=ping distance=2 gateway=192.168.3.1 routing-mark=ISP2
```

Как видим таблицы **main** и **ISP1** полностью повторяют друг друга и основным шлюзом (с меньшей дистанцией) в них первый провайдер, таблица **ISP2**, наоборот, использует основным шлюзом второго провайдера. Таким образом каждая из дополнительных таблиц при нормальной работе сети будет отправлять трафик через свой канал, а при аварии - через оставшийся рабочий.

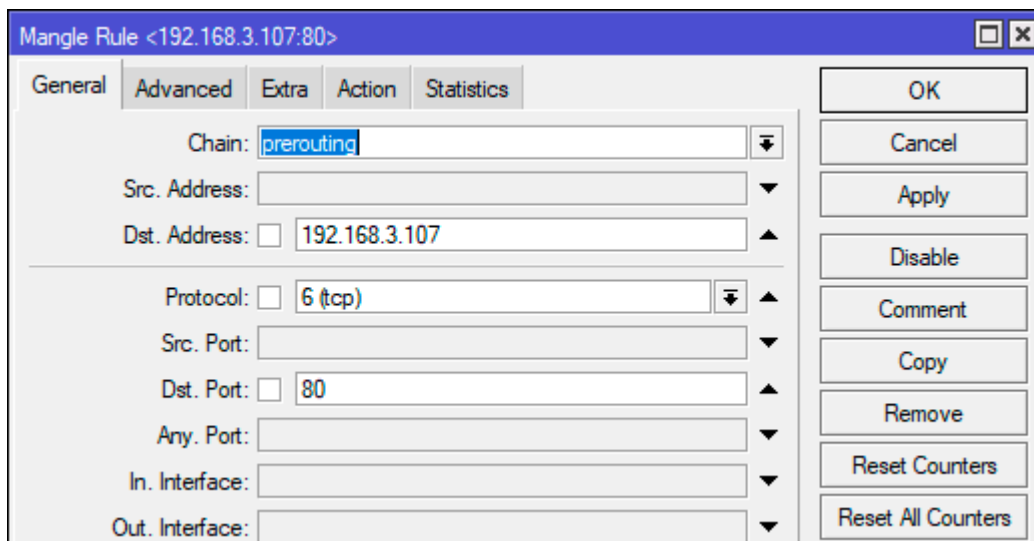
## Маркировка трафика для проброшенных портов

---

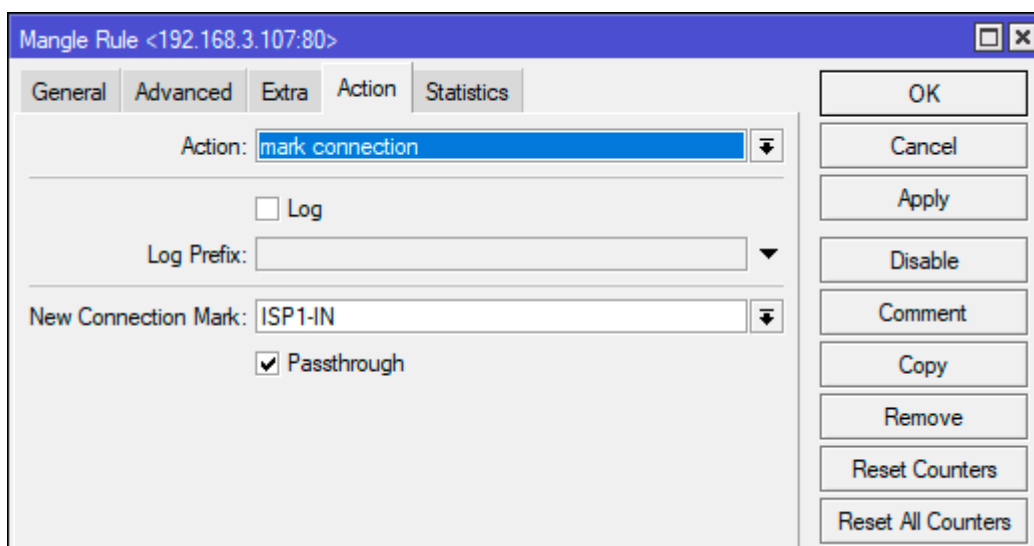
Как мы уже говорили выше при балансировке важно исключить из нее входящие соединения на проброшенные порты, так как ответ должен уходить в тот же канал, что и пришел запрос. Это можно сделать разными способами, чаще всего трафик маркируют по интерфейсу входа, но в этом случае в него попадает весь входящий трафик, в т.ч. и предназначенный роутеру, что заставляет вводить дополнительные правила для маркировки таких соединений. Но можно решить вопрос по-другому, более точно.

Какой именно трафик, на какие именно порты и по какому протоколу приходит нам известно, поэтому будем выборочно маркировать только его. Все действия по маркировке мы будем выполнять в цепочке **PREROUTING** таблицы **mangle**, при этом помним, что правила обрабатываются в порядке очереди до первого терминирующего правила.

Переходим в **IP - Firewall - Mangle** и создаем следующее правило: **Chain - prerouting**, **Dst Address - 192.168.3.107** - внешний адрес на канале первого провайдера, **Protocol - tcp**, **Dst. Port - 80**. Если проброшено несколько портов, то просто перечисляем их через запятую. В результате у вас должно получиться два правила, одно для протокола TCP, второе для UDP.



На вкладке **Action** выбираем действие **mark connection**, в поле **New Connection Mark** ставим марку **ISP1-IN**, т.е. входящие соединения через первого провайдера (название марки можете выбрать на собственное усмотрение), также обязательно ставим флаг **Passthrough**, чтобы пакет продолжил движение по таблице.



Быстрее выполнить действия в терминале:

```
/ip firewall mangle
add action=mark-connection chain=prerouting dst-address=192.168.3.107 dst-port=80 \
    new-connection-mark=ISP1-IN passthrough=yes protocol=tcp
```

Затем создаем аналогичное правило для входящих соединений второго провайдера, только указываем второй внешний IP адрес и ставим марку ISP2-IN.

Соединения мы промаркировали, теперь нужно направить трафик в нужные таблицы маршрутизации, поэтому ниже создадим следующее правило: **Chain - prerouting, Src/ Address - 192.168.111.0/24, Connection Mark - ISP1-IN.**

Mangle Rule <192.168.111.0/24>

General Advanced Extra Action Statistics

Chain: prerouting

Src. Address: 192.168.111.0/24

Dst. Address:

Protocol:

Src. Port:

Dst. Port:

Any. Port:

In. Interface:

Out. Interface:

In. Interface List:

Out. Interface List:

Packet Mark:

Connection Mark: ISP1-IN

Routing Mark:

Routing Table:

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

На закладке **Action** выбираем **mark routing** и в поле **New Routing Mark** указываем таблицу маршрутизации, в которую мы направляем трафик, в нашем случае **ISP1**, флаг **Passthrough** не ставим, данное правило будет для пакета терминирующим.

Mangle Rule <192.168.111.0/24>

General Advanced Extra Action Statistics

Action: mark routing

☐ Log

Log Prefix:

New Routing Mark: ISP1

☐ Passthrough

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

В терминале:

```
/ip firewall mangle
add action=mark-routing chain=prerouting connection-mark=ISP1-IN new-routing-
mark=ISP1 \
    passthrough=no src-address=192.168.111.0/24
```

Затем создаем такое же правило для второго провайдера, которое будет направлять пакеты соединения с маркой **ISP2-IN** в таблицу маршрутизации **ISP2**. Данный набор правил всегда должен стоять **раньше всех других правил по**



балансировке в таблице **Mangle**.

## Собственный трафик роутера

---

Эта настройка понадобится вам, если у вас на роутере есть публичные сервисы, которые должны быть доступны через обоих провайдеров, например, VPN-сервер. Обратите внимание, что это не касается проброшенных портов, а именно собственных служб роутера. Здесь у нас стоит точно такая же задача - отправить трафик в тот же канал, через который он пришел.

Здесь мы будем действовать более широко, критерием будет интерфейс входа, промаркируем входящий трафик следующим образом:

```
/ip firewall mangle
add action=mark-connection chain=prerouting in-interface=ether1 \
    new-connection-mark=ISP1-IN passthrough=yes
add action=mark-connection chain=prerouting in-interface=ether3 \
    new-connection-mark=ISP2-IN passthrough=yes
```

Нетрудно заметить, что в данные правила попадает весь входящий трафик, как собственный роутера, так и транзитный. А вот далее есть особенности. Выше мы маркировали пакеты в цепочке **PREROUTING** таблицы **mangle**, но исходящий трафик роутера туда не попадает, поэтому мы должны маркировать пакеты в двух местах. В **OUTPUT** для собственного трафика роутера:

```
/ip firewall mangle
add chain=output connection-mark=ISP1-IN action=mark-routing new-routing-mark=ISP1
add chain=output connection-mark=ISP2-IN action=mark-routing new-routing-mark=ISP2
```

И в **PREROUTING** для транзитного:

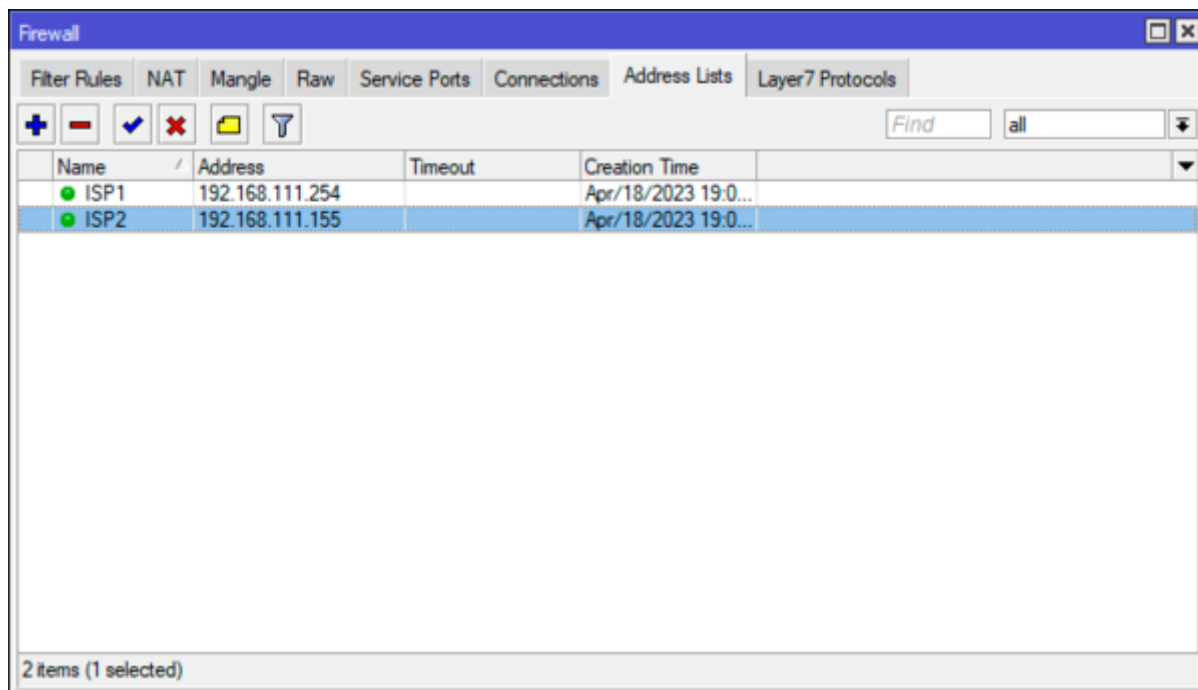
```
/ip firewall mangle
add action=mark-routing chain=prerouting connection-mark=ISP1-IN new-routing-mark=ISP1 \
    passthrough=no src-address=192.168.111.0/24
add action=mark-routing chain=prerouting connection-mark=ISP2-IN new-routing-mark=ISP2 \
    passthrough=no src-address=192.168.111.0/24
```

Эти правила точно также должны находиться в самом верху таблицы **mangle** и заменяют правила для проброшенного трафика.

## Ручная балансировка каналов

---

Самый простой способ, в котором мы можем вручную указать какой именно трафик в какую таблицу маршрутизации направить. При этом для выделения этого трафика мы можем использовать любые критерии доступные в цепочке **prerouting**. В нашем примере мы будем использовать в качестве критерия адрес источника. Создадим в IP - Firewall - Address Lists два адресных листа ISP1 и ISP2, затем добавим в них IP-адреса ПК, которые будут выходить в сеть через первый и второй канал соответственно.



Или:

```
/ip firewall address-list
add address=192.168.111.254 list=ISP1
add address=192.168.111.155 list=ISP2
```

Теперь выполним маркировку попадающих под правило соединений:

```
/ip firewall mangle
add action=mark-connection chain=prerouting connection-mark=no-mark \
    new-connection-mark=ISP1 passthrough=yes src-address-list=ISP1
```

Данное правило промаркирует все соединения, **адрес источник** которых входит в **лист ISP1** и **не имеет других марок** - критерий **Connection Mark - no mark** - маркой **ISP1**. Немного о последнем критерии, с его помощью мы маркируем только те соединения, которые не имеют уже установленной марки. Это сделано для того, чтобы избежать возможной перемаркировки пакетов. И хотя в нашей конфигурации в данном месте таких быть не должно лучше все же подстраховаться на будущее.

Таким же образом маркируем соединения для второго провайдера на основании адресов из второго списка.

Затем перейдем к маршрутизации, создадим правило, которое будет направлять трафик, промаркированный как ISP1 в первую таблицу маршрутизации:

```
/ip firewall mangle
add action=mark-routing chain=prerouting connection-mark=ISP1 new-routing-
mark=ISP1 \
    passthrough=no src-address=192.168.111.0/24
```

Здесь же создадим правило для второй таблицы маршрутизации.

В результате полный набор правил будет выглядеть так:

```

/ip firewall mangle
add action=mark-connection chain=prerouting connection-mark=no-mark \
new-connection-mark=ISP1 passthrough=yes src-address-list=ISP1
add action=mark-connection chain=prerouting connection-mark=no-mark \
new-connection-mark=ISP2 passthrough=yes src-address-list=ISP2

    add action=mark-routing chain=prerouting connection-mark=ISP1 new-routing-
mark=ISP1 \
    passthrough=no src-address=192.168.111.0/24
add action=mark-routing chain=prerouting connection-mark=ISP2 new-routing-
mark=ISP2 \
    passthrough=no src-address=192.168.111.0/24

```

Аналогичным образом можно решать и иные задачи по разделению трафика между каналами, например, сделать выборочный обход блокировок. Для этого маркируем трафик на основании адреса назначения согласно спискам и направляем в нужную таблицу маршрутизации.

## Балансировка NTH (по соединениям)

---

Довольно простой метод балансировки, когда мы оперируем соединениями и полностью соответствует принципу "на первый-второй рассчитайсь". Т.е. первое соединение направляем в первый канал, следующее во второй, затем снова в первый и т.д. Но если реализовать этот способ без дополнительных настроек, то мы получим ситуацию, когда трафик от одного узла к другому внешнему узлу может идти сразу через оба канала, что является нежелательным. Поэтому мы будем действовать по следующему алгоритму:

- Маркировать новые соединения по принципу NTH
- Добавлять адрес источник промаркированного соединения в список
- В начале таблицы Mangle перехватываем трафик и принудительно маркируем на основании списков

Ниже будем рассматривать сразу готовое решение, начнем с перехвата трафика, для этого добавим уже известное нам по прошлому методу правило:

```

/ip firewall mangle
add action=mark-connection chain=prerouting connection-mark=no-mark \
new-connection-mark=ISP1 passthrough=yes src-address-list=ISP1

```

Затем продублируем его для списка второго провайдера и направим маркированный трафик в нужные таблицы маршрутизации:

```

/ip firewall mangle
add action=mark-routing chain=prerouting connection-mark=ISP1 new-routing-
mark=ISP1 \
    passthrough=no src-address=192.168.111.0/24

```

По сути, мы продублировали набор правил для ручной балансировки, но есть одно существенное отличие, если тогда мы формировали списки вручную, то теперь будем делать это динамически. Еще ниже добавляем следующее правило для

маркировки соединений: **Chain - prerouting**, **Src. Address - 192.168.111.0/24**,  
**Connection Mark - no mark**, **Connection state - new**:

Mangle Rule <192.168.111.0/24>

General Advanced Extra Action Statistics

Chain: **prerouting**

Src. Address: **192.168.111.0/24**

Dst. Address:

Protocol:

Src. Port:

Dst. Port:

Any. Port:

In. Interface:

Out. Interface:

In. Interface List:

Out. Interface List:

Packet Mark:

Connection Mark: **no-mark**

Routing Mark:

Routing Table:

Connection Type:

Connection State: ☐ invalid ☐ established ☐ related ☒ new ☐ untracked

Connection NAT State:

disabled

OK Cancel Apply Enable Comment Copy Remove Reset Counters Reset All Counters

Еще раз обращаем внимание, что мы маркируем только **новые** соединения, адрес источника которых - адреса локальной сети и которые не имеют других марок.

Затем переходим на закладку **Extra** и в разделе **Nth** ставим **Every 2, Packet 1** - это означает что под условия будет попадать каждый первый пакет из двух.

Mangle Rule <192.168.111.0/24>

General Advanced **Extra** Action Statistics

Connection Limit

Limit

Dst. Limit

Nth

Every: **2**

Packet: **1**

Time

Src. Address Type

Dst. Address Type

PSD

Hotspot

IP Fragment

OK Cancel Apply Enable Comment Copy Remove Reset Counters Reset All Counters

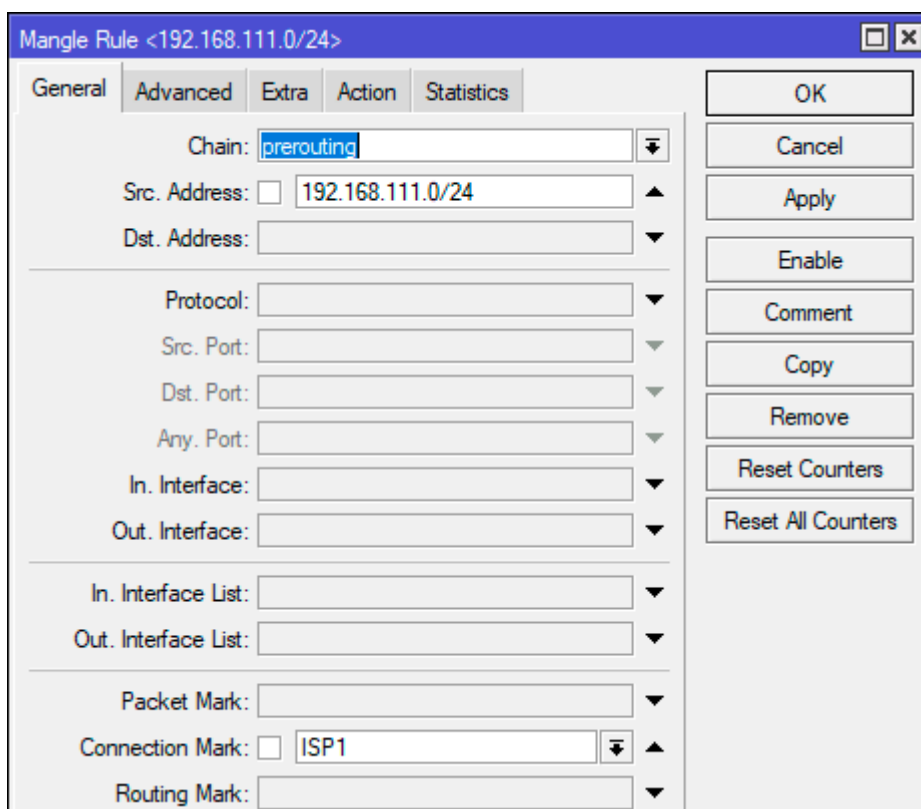
На закладке **Action** выбираем **mark connection**, **New Connection Mark - ISP1** и ставим флаг **Passthrough**.

В терминале:

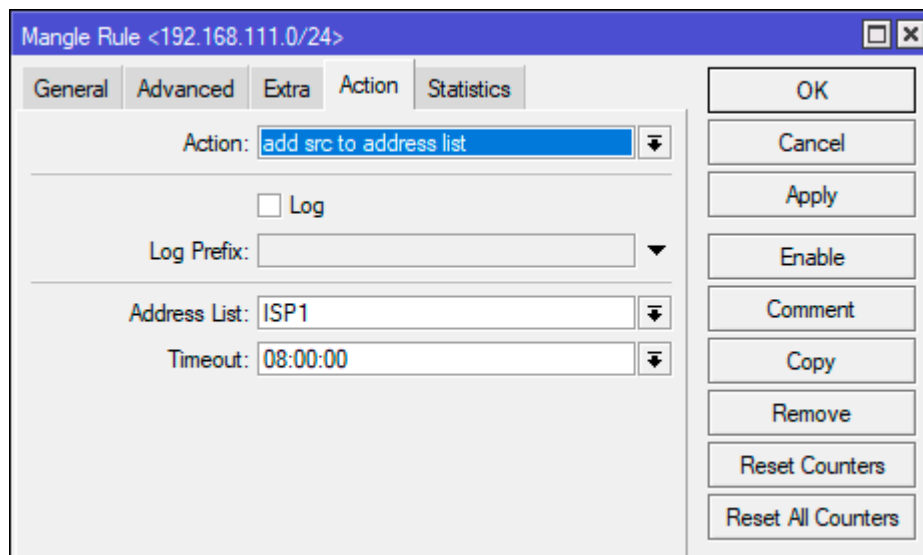
```
/ip firewall mangle
add action=mark-connection chain=prerouting connection-mark=no-mark connection-
state=new \
    new-connection-mark=ISP1 nth=2,1 passthrough=yes src-address=192.168.111.0/24
```

Затем аналогичное правило для второго провайдера, только в **Nth** ставим **Every 2, Packet 2** или **nth=2,2** для терминала.

Так как флаг **Passthrough** установлен - пакет идет дальше, следующим шагом будет добавление адреса источника в соответствующий список. Для этого добавим следующее правило: **Chain - prerouting**, **Src/ Address - 192.168.111.0/24**, **Connection Mark - ISP1**:



Переходим в **Actions** и указываем действие **add src to address list**, в поле **Address List** указываем список первого провайдера **ISP1**, ниже указываем **срок действия записи**, а нашем случае **8 часов**. Теперь после выполнения балансировки все соединения этого узла будут закреплены за выбранным каналом на 8 часов, срок действия записи можете указать по собственному усмотрению.



В терминале:

```
/ip firewall mangle
add action=add-src-to-address-list address-list=ISP1 address-list-timeout=8h \
chain=prerouting connection-mark=ISP1 src-address=192.168.111.0/24
```

Данное действие терминирующим не является и пакет идет по цепочке дальше, поэтому мы должны отправить его в нужную таблицу маршрутизации разместив еще одно правило.

```
/ip firewall mangle
add action=mark-routing chain=prerouting connection-mark=ISP1 new-routing-
mark=ISP1 \
passthrough=no src-address=192.168.111.0/24
```

Повторяем указанный набор правил для второго провайдера.

В результате у нас получится:

```

/ip firewall mangle
add action=mark-connection chain=prerouting connection-mark=no-mark \
new-connection-mark=ISP1 passthrough=yes src-address-list=ISP1
add action=mark-connection chain=prerouting connection-mark=no-mark \
new-connection-mark=ISP2 passthrough=yes src-address-list=ISP2

add action=mark-routing chain=prerouting connection-mark=ISP1 new-routing-
mark=ISP1 \
    passthrough=no src-address=192.168.111.0/24
add action=mark-routing chain=prerouting connection-mark=ISP2 new-routing-
mark=ISP2 \
    passthrough=no src-address=192.168.111.0/24

add action=mark-connection chain=prerouting connection-mark=no-mark connection-
state=new \
new-connection-mark=ISP1 nth=2,1 passthrough=yes src-address=192.168.111.0/24
add action=mark-connection chain=prerouting connection-mark=no-mark connection-
state=new \
new-connection-mark=ISP2 nth=2,2 passthrough=yes src-address=192.168.111.0/24

add action=add-src-to-address-list address-list=ISP1 address-list-timeout=8h \
    chain=prerouting connection-mark=ISP1 src-address=192.168.111.0/24
add action=add-src-to-address-list address-list=ISP2 address-list-timeout=8h \
    chain=prerouting connection-mark=ISP2 src-address=192.168.111.0/24

add action=mark-routing chain=prerouting connection-mark=ISP1 new-routing-
mark=ISP1 \
    passthrough=no src-address=192.168.111.0/24
add action=mark-routing chain=prerouting connection-mark=ISP2 new-routing-
mark=ISP2 \
    passthrough=no src-address=192.168.111.0/24

```

Внимательный читатель заметит, что у нас появилось два одинаковых блока с действием **mark routing** и сразу возникает резонный вопрос: а нельзя ли обойтись одним. Можно, оставив нижний, но при этом нужно учитывать, что большая часть трафика будет промаркирована именно первыми правилами, основываясь на нахождении адреса источника в списке и поэтому дальнейшее прохождение пакетов по цепочке является нежелательным с точки зрения производительности. Поэтому первый блок сразу отправляет пакеты в нужную таблицу маршрутизации прерывая их дальнейшее прохождение вниз по правилам.

Теперь некоторое время поработаем и заглянем в списки, как видим узлы равномерно распределены по каналам и добавлялись туда строго по очереди (обратите внимание на колонку с временем добавления):

Firewall					
Filter Rules NAT Mangle Raw Service Ports Connections Address Lists Layer7 Protocols					
<div> <div>+</div> <div>-</div> <div>✓</div> <div>✗</div> <div>📄</div> <div>🔍</div> </div> <div>Find all</div>					
Name	Address	Timeout	Creation Time		
D ● ISP1	192.168.111.155	07:55:04	Apr/18/2023 19:48:49		
D ● ISP2	192.168.111.254	07:54:45	Apr/18/2023 19:49:55		
D ● ISP1	192.168.111.152	07:57:24	Apr/18/2023 19:53:40		
D ● ISP2	192.168.111.154	07:59:55	Apr/18/2023 19:56:10		

Если вам нужно, чтобы какой-то узел постоянно работал через один и тот же канал, просто добавьте его в нужный список вручную.

Следующий вопрос: как быть при разной пропускной способности каналов. Допустим ISP1 предоставляет канал в 100 Мбит/с, а ISP2 только 50 Мб/с. В таком случае нам нужно изменить условия балансировки, как показывают несложные математические вычисления, вместо балансировки 1/2 нам нужно балансировать 1/3. Т.е. в условиях **Nth** ставим **Every 3**, а затем два пакета маркируем для первого провайдера, а третий для второго.

```
add action=mark-connection chain=prerouting connection-mark=no-mark connection-
state=new \
    new-connection-mark=ISP1 nth=3,1 passthrough=yes src-address=192.168.111.0/24
add action=mark-connection chain=prerouting connection-mark=no-mark connection-
state=new \
    new-connection-mark=ISP1 nth=3,2 passthrough=yes src-address=192.168.111.0/24
add action=mark-connection chain=prerouting connection-mark=no-mark connection-
state=new \
    new-connection-mark=ISP2 nth=3,3 passthrough=yes src-address=192.168.111.0/24
```

Никакие другие правила данное изменение не затрагивает и больше изменять ничего не нужно.

## Балансировка PCC (per connection classifier)

Это более сложный способ и в качестве критериев балансировки используются критерии соединения, такие как адрес и порт источника или назначения. При этом данный способ более прост в реализации за счет того, что нам не нужно контролировать ряд аспектов, например, привязку всех соединений узла к каналу.

Начнем. Прежде всего добавим правило маркировки соединений, маркировать мы все также будем трафик с адресами источника **Src. Address - 192.168.111.0/24** и не имеющих других марок **Connection Mark - no mark**. Только на закладке **Advanced** добавим критерий **Per Connection Classifier - src. address 2/0**.

Это означает что в качестве критерия классификации мы выбираем адрес источник (доступны также адрес и порт источника, адрес назначения, адрес и порт назначения, оба адреса и т.д.) на основании которого вычисляется 32-битный хеш, затем этот хеш делится на указанное в числителе дроби число и остаток сравнивается со знаменателем, при совпадении происходит срабатывание правила.



Так как делим мы на 2, то возможные остатки у нас 0 и 1, для первого провайдера указываем ноль и присваиваем соединению метку ISP1, а чтобы пакет пошел дальше не забываем про **Passthrough**.

Mangle Rule <192.168.111.0/24>

General Advanced Extra Action Statistics

Src. Address List: 192.168.111.0/24

Dst. Address List:

Layer7 Protocol:

Content:

Connection Bytes:

Connection Rate:

Per Connection Classifier: ☐ src address : 2 / 0

Src. MAC Address:

Out. Bridge Port:

In. Bridge Port:

In. Bridge Port List:

Out. Bridge Port List:

IPsec Policy:

TLS Host:

Ingress Priority:

Priority:

DSCP (TOS):

enabled

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

Аналогичное правило создаем и для второго провайдера, только в критерии ставим в знаменатель остаток 2. Если нам нужно балансировать каналы в иной пропорции, то можем использовать иные параметры, так повторяя указанный выше пример с каналами 100 Мбит/с и 50 Мбит/с следует использовать числитель 3 и знаменатели 0,1,2. Правил будет также три, два из них должны ставить метку ISP1, одно - ISP2.

Добавить правило в терминале можно так:

```
/ip firewall mangle
add action=mark-connection chain=prerouting connection-mark=no-mark new-
connection-mark=ISP1 \
    passthrough=yes per-connection-classifier=src-address:2/0 src-
address=192.168.111.0/24
```

Затем ниже добавляем правила направляющие промаркированный трафик в нужную таблицу маршрутизации, пример для первого провайдера:

```
/ip firewall mangle
add action=mark-routing chain=prerouting connection-mark=ISP1 new-routing-
mark=ISP1 \
    passthrough=no src-address=192.168.111.0/24
```

Полный набор правил будет выглядеть так:

```
/ip firewall mangleadd action=mark-connection chain=prerouting connection-mark=no-
mark new-connection-mark=ISP1 \
    passthrough=yes per-connection-classifier=src-address:2/0 src-
address=192.168.111.0/24
add action=mark-connection chain=prerouting connection-mark=no-mark new-
connection-mark=ISP2 \
    passthrough=yes per-connection-classifier=src-address:2/1 src-
address=192.168.111.0/24
```

```
add action=mark-routing chain=prerouting connection-mark=ISP1 new-routing-
mark=ISP1 \
    passthrough=no src-address=192.168.111.0/24
add action=mark-routing chain=prerouting connection-mark=ISP2 new-routing-
mark=ISP2 \
    passthrough=no src-address=192.168.111.0/24
```

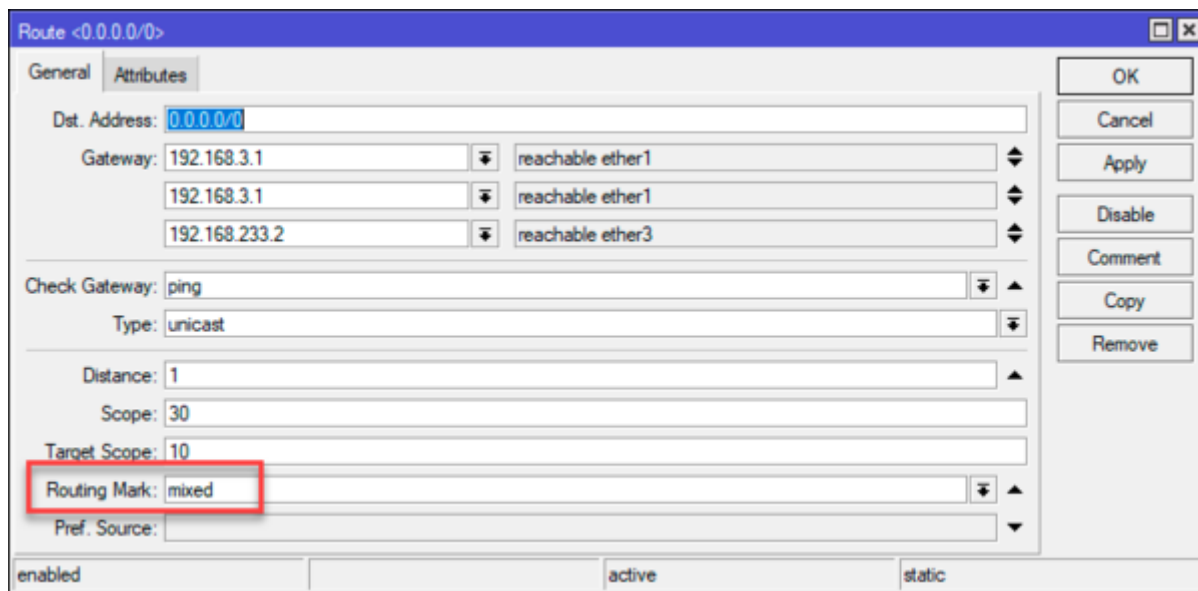
Как видим, несмотря на более сложный алгоритм реализация этого способа гораздо проще, так как классификацию соединений берет на себя роутер и нам не нужно создавать дополнительные списки и правила.

## Балансировка ECMP (equal cost multipath routing)

---

Аббревиатура ECMP расшифровывается как маршрутизация через множество путей равной стоимости. А так как все пути равнозначны, то трафик будет равномерно балансироваться между ними. Этот метод широко используется провайдерами, но для конечных сетей это не самый лучший вариант. Почему? Потому что мы не можем повлиять на выбор пути для соединения согласно каким-либо критериям и поэтому несколько соединений от одного внутреннего узла к одному внешнему могут пойти и пойдут разными путями. К чему это может привести мы писали в начале этой статьи. Поэтому подходите к выбору этого способа балансировки взвесив все за и против.

Прежде всего создадим еще одну таблицу маршрутизации и добавим в нее нулевой маршрут с несколькими шлюзами, если нужна балансировка в разных пропорциях, то следует добавить нужный шлюз несколько раз. Ниже показан пример для каналов 100 Мбит/с и 50 Мбит/с у первого и второго провайдера. Указанные шлюзы будут использоваться по кольцу (Round-robin). Обратите внимание, что новый маршрут мы поместили в новую таблицу **mixed**.



В терминале:

```
/ip route
add check-gateway=ping distance=1 gateway=\
    192.168.3.1,192.168.3.1,192.168.233.2 routing-mark=mixed
```

Теперь поймаем все исходящие соединения локальной сети без марок и назначим им марку **mixed**:

```
/ip firewall mangle
add action=mark-connection chain=prerouting connection-mark=no-mark \
    new-connection-mark=mixed passthrough=yes src-address=192.168.111.0/24
```

И сразу завернем этот трафик в нужную таблицу маршрутизации:

```
/ip firewall mangle
add action=mark-routing chain=prerouting connection-mark=mixed new-routing-
mark=mixed \
    passthrough=no src-address=192.168.111.0/24
```

Полностью конфигурация будет выглядеть так:

```
/ip route
add check-gateway=ping distance=1 gateway=\
192.168.3.1,192.168.3.1,192.168.233.2 routing-mark=mixed

/ip firewall mangle
add action=mark-connection chain=prerouting connection-mark=no-mark \
    new-connection-mark=mixed passthrough=yes src-address=192.168.111.0/24
add action=mark-routing chain=prerouting connection-mark=mixed new-routing-
mark=mixed \
    passthrough=no src-address=192.168.111.0/24
```

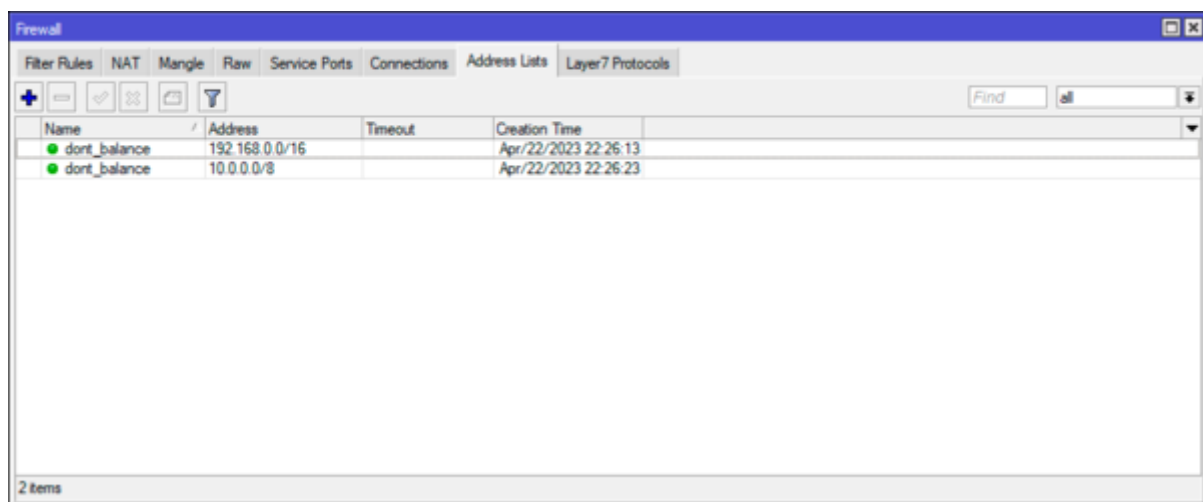
Как можно заметить, данный способ вообще предельно прост, но имеет ряд особенностей, которые следует учитывать.

## Другие сети за роутером

Все это время мы просто маркировали исходящие пакеты из локальной сети, не разбираясь их назначением, молча подразумевая что все они уходят во внешнюю сеть. Но как быть, если за роутером находятся другие внутренние сети, неважно, непосредственно подключенные или через VPN. Очевидно, что нам нужно исключить их из балансировки.

Первое, что приходит на ум - это создать дополнительный список, скажем **dont\_balance** и добавить его дополнительным критерием к правилам маркировки соединения через **Dst. Address List**, но таких правил может быть много и дополнительный критерий увеличит нагрузку на роутер. Поэтому мы пойдем другим путем, а именно вспомним, что во всех правилах маркировки мы использовали критерий **Connection Mark - no mark**. Следовательно, нам достаточно добавить таким соединениям собственную марку, и они не будут маркироваться правилами балансировки.

Чтобы не делать много записей можно сразу добавить в список диапазоны используемых частных сетей. На безопасность это не влияет, но упрощает администрирование.



В терминале создать и заполнить список можно так:

```
/ip firewall address-list
add address=192.168.0.0/16 list=dont_balance
add address=10.0.0.0/8 list=dont_balance
```

Теперь промаркируем входящие соединения из этих сетей в локальную сеть и исходящие из локальной во внутренние сети:

```
/ip firewall mangle
add action=mark-connection chain=prerouting dst-address-list=dont_balance \
    new-connection-mark=LOCAL passthrough=yes src-address=192.168.111.0/24
add action=mark-connection chain=prerouting dst-address=192.168.111.0/24 \
    new-connection-mark=LOCAL passthrough=yes src-address-list=dont_balance
```

Далее можно не делать ничего, соединения с уже существующей меткой пройдут сквозь остальные правила и останутся в основной таблице маршрутизации, но если такого трафика много, то с целью уменьшения нагрузки на роутер добавим ниже еще одно правило, которое принудительно отправит эти пакеты в основную таблицу и прекратит их прохождение по цепочке.

```
/ip firewall mangle  
add action=mark-routing chain=prerouting connection-mark=LOCAL new-routing-  
mark=main passthrough=no
```

Эти правила должны располагаться в самом верху таблицы Mangle, выше всех других правил балансировки.

### **Онлайн-курс по MikroTik**

Научиться настраивать MikroTik с нуля или систематизировать уже имеющиеся знания можно на [углубленном курсе по администрированию MikroTik](#). Автор курса, сертифицированный тренер MikroTik Дмитрий Скоромнов, лично проверяет лабораторные работы и контролирует прогресс каждого своего студента. В три раза больше информации, чем в вендорской программе MTCNA, более 20 часов практики и доступ навсегда.

---