

Резервирование каналов в Mikrotik при помощи рекурсивной маршрутизации

 interface31.ru/tech_it/2020/04/rezervirovanie-kanalov-v-mikrotik-pri-pomoshhi-rekursivnoy-marshrutizacii.html

Записки IT специалиста

Технический блог специалистов ООО "Интерфейс"

- [Главная](#)
- Резервирование каналов в Mikrotik при помощи рекурсивной маршрутизации

Значение интернета в повседневной жизни и деятельности предприятий всех форм и размеров с каждым днем только растет, особенно сейчас, когда весь мир переходит на дистанционные методы работы в связи с пандемией коронавируса. Но этот вопрос актуален также и для торговых предприятий, платежные карты используются практически каждым вторым покупателем и отсутствие интернета мгновенно скажется на выручке предприятия. В данной статье мы рассмотрим, каким образом можно организовать резервный канал и обеспечить быстрое переключение на него и обратно штатными средствами RouterOS.



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.

Как показывает читательский отклик, маршрутизация для многих администраторов является достаточно сложной темой. Тем не менее если вы внимательно выполните все инструкции данной статьи, то всё будет работать. Однако слепое выполнение инструкций не добавляет понимания происходящих процессов и, если что-то пойдет не так вы окажетесь в весьма затруднительном положении, поэтому перед тем, как перейти к практике мы выполним небольшой теоретический ликбез.

Теория

Действительно, маршрутизация достаточно сложная тема, поэтому мы сознательно упростили многие моменты для того, чтобы обеспечить понимание процессов начинающими в объеме необходимом для решения поставленной задачи.

Что такое маршрутизация? Это процесс определения пути следования данных в сетях связи. Его можно сравнить с прокладкой маршрута с использованием навигатора, вы задаете начальную и конечную точку и получаете несколько вариантов проезда с указанием всех промежуточных точек. В некоторых точках маршрут может разделяться на несколько вариантов, из которых вы можете выбрать наиболее оптимальный, исходя из оценки текущей дорожной обстановки.

Для каждого узла маршрута, а наш роутер является одним из них, решение о маршрутизации принимается на основании таблицы маршрутизации. Рассмотрим ее простейший вариант, за основу мы взяли стандартный вывод в терминале Mikrotik и добавили дополнительные данные, сейчас они могут быть непонятны, но ниже мы будем к ним обращаться.

```
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static
#          DST-ADDRESS      PREF-SRC      GATEWAY      DISTANCE  SCOPE  TARGET
SCOPE
0 DAS  0.0.0.0/0                192.168.3.1      1        30
10
1 DAC  192.168.3.0/24    192.168.3.113    ether5        0        10
10
2 DAC  192.168.186.0/24  192.168.186.1    bridge1       0        10
10
```

Прежде всего разберемся как заполняется таблица маршрутизации. Это может происходить несколькими способами:

- **Непосредственно присоединенные сети** - в данном случае они имеют флаг **C - connect** - это сети к которым физически подключен маршрутизатор, они добавляются при подключении интерфейса и удаляются при его отключении.
- **Статические маршруты** - добавляются системным администратором, либо динамически, при подключении коммутируемого соединения или получения сетевых параметров по DHCP. В этом случае к флагу **S -static**, добавляется флаг **D - dynamic**.
- **Динамические маршруты** - маршруты добавляемые при помощи протоколов динамической маршрутизации, таких как OSPF, RIP и т.д. Не следует путать их с динамически добавляемыми статическими маршрутами и маршрутами непосредственно присоединенных сетей. Флаг **D - dynamic** в Mikrotik обозначает только то, что маршрут добавлен автоматически, без участия пользователя. А динамические маршруты обозначаются флагами **r - rip**, **b - bgp**, **o - ospf**, **m - mme**, однако их рассмотрение выходит за рамки данной статьи.

К одной цели могут вести несколько маршрутов, приоритетом их выбора является значение **административной дистанции**, чем ниже это значение, тем выше приоритет у маршрута. Если вернуться к нашей аналогии с навигатором, то он может проложить нам несколько маршрутов, но приоритет получит самый короткий. Но если на этом маршруте возникнут серьезные затруднения, скажем, пробка или

авария, то мы можем выбрать более длинный маршрут. Также и здесь, если маршрут с наименьшей административной дистанцией недоступен, то выбирается маршрут с более высокой дистанцией.

Теперь разберемся как все это работает. Допустим мы хотим обратиться к узлу **8.8.8.8**, роутер берет таблицу маршрутизации и осуществляет в ней поиск наилучшего маршрута к запрашиваемому узлу. В нашем случае искать особо нечего и это будет **"нулевой"** маршрут со шлюзом **192.168.3.1**. Но маршрутизаторы крупных сетевых узлов могут содержать тысячи и десятки тысяч маршрутов и поэтому поиск может оказаться совсем не простым и дешевым действием.

Коротко напомним о том, что такое **"нулевой"** маршрут или **шлюз по умолчанию**, он используется в том случае, если в таблице для узла назначения не было найдено ни одного подходящего маршрута. Проще говоря, если роутер не знает куда отправить данные, он отправляет их шлюзу по умолчанию. Особой записи для узла **8.8.8.8** в нашей таблице нет, поэтому будет использоваться **"нулевой"** маршрут.

Хорошо, адрес шлюза мы определили, но это не решило основной задачи - куда отправить данные. Почему? Вернемся к нашей аналогии, вы решили поехать из Белгорода в Москву, карта показывает, что сначала вам нужно приехать в Курск. Отлично, но как именно в него попасть? Навигатор услужливо подскажет нам, что требуется ехать на север по федеральной трассе М2 «Крым».

В нашем случае основной задачей маршрутизатора будет определить **интерфейс выхода**, т.е. куда именно физически нужно направить данные, чтобы они добрались до требуемого места назначения. Поэтому роутер снова начинает поиск в таблице маршрутизации чтобы определить каким образом следует добраться до искомого адреса шлюза **192.168.3.1**, согласно таблице, будет найден маршрут к сети **192.168.3.0/24** через интерфейс **ether5**. Так как интерфейс выхода определен, то поиск прекращается и роутер формирует необходимый тип данных для передачи на канальном уровне.

Здесь тоже следует дать краткие пояснения, если интерфейс выхода является сетью Ethernet, то роутер выполнит ARP-запрос для адреса шлюза и сформирует Ethernet-кадр, а если это туннельный интерфейс точка-точка (VPN, PPPoE), то выполнит инкапсуляцию IP-пакета и отправит его на другую сторону туннеля.

И здесь мы приходим к следующим соображениям: таблица маршрутизации может быть большая и поиск по ней может быть затратным, в тоже время шлюз будет располагаться в одной из непосредственно присоединенных сетей, возможно следует ограничить поиск только по ним?

Для решения этой проблемы в Linux, а следовательно, и в RouterOS, введены понятия области маршрута - **Scope**, и области поиска - **Target-scope**. Теперь вернемся к нашей таблице маршрутизации, **"нулевой"** маршрут имеет **область - 30**

и **область поиска - 10**. Следовательно дальнейший поиск будет вестись только среди маршрутов с **областью 10**.

По умолчанию данные опции имеют следующее значение:

- **Непосредственно присоединенные сети: Scope - 10, Target Scope** не имеет смысла, так как данный маршрут указывает на **интерфейс выхода**
- **Динамические маршруты - OSPF, RIP, MME: Scope - 20, Target Scope - 10**
- **Статические маршруты: Scope - 30, Target Scope - 10**

Таким образом область поиска существенно снижается и маршрут к шлюзу ищется только среди непосредственно присоединенных сетей. Теперь самое время перейти к рекурсивной маршрутизации.

Mikrotik имеет возможность проверять доступность шлюза, используя для этого пинг, если шлюз оказывается недоступным, то маршрут становится неактивным. Это хорошо, но позволяет выявлять только проблемы в сети провайдера, и если шлюз остается доступным, то маршрут будет продолжать оставаться активным, несмотря на отсутствие доступа в интернет. Но что, если указать в качестве шлюза удаленный высокодоступный узел?

Давайте рассмотрим следующую таблицу маршрутизации:

Flags: X - disabled, A - active, D - dynamic, C - connect, S - static

#		DST-ADDRESS	PREF-SRC	GATEWAY	DISTANCE	SCOPE	TARGET
SCOPE							
0	AS	0.0.0.0/0		1.1.1.1	1	30	
10							
1	AS	1.1.1.1/32		192.168.3.1	1	10	
10							
2	DAC	192.168.3.0/24	192.168.3.113	ether5	0	10	
10							
3	DAC	192.168.186.0/24	192.168.186.1	bridge1	0	10	
10							

В качестве шлюза по умолчанию мы указали адрес одного из DNS-серверов Cloudflare, обратите внимание, что такой адрес не должен использоваться в вашей сети, потому что после отказа связанного с ним провайдера он окажется недоступен. Теперь мы можем проверять наличие интернета за пределами сети провайдера и своевременно переключать маршруты в случае его недоступности.

Теперь разберемся как это работает. Когда мы хотим связаться с каким-либо узлом в сети интернет маршрутизатор выполнит поиск по таблице маршрутизации и, если не указано иных маршрутов, выберет "нулевой" в котором указан шлюзом **1.1.1.1**, так как найден адрес, то поиск будет продолжен, чтобы определить интерфейс выхода. Так как указанный нами шлюз не находится ни в одной из непосредственно присоединенных сетей, то сделать это не удастся и такой маршрут работать не будет.

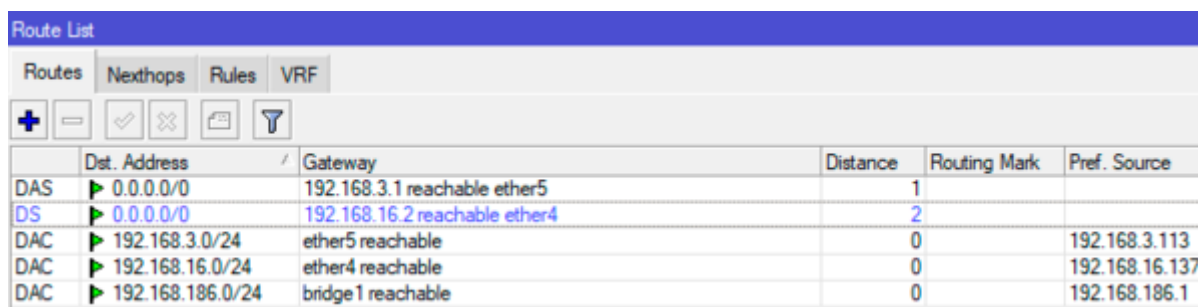
Поэтому мы "подскажем" маршрутизатору как достичь указанного нами узла и добавим маршрут к 1.1.1.1 через сеть провайдера, а чтобы он участвовал в поиске его область должна быть меньше или равна области поиска "нулевого" маршрута, т.е. 10. Это очень важный момент, если вы неправильно укажете область второго маршрута - рекурсивная маршрутизация работать не будет.

Итак, роутер находит адрес первого шлюза - 1.1.1.1 и начинает искать маршрут к нему с среди маршрутов с областью 10, там он находит второй маршрут и получает из него новое значение шлюза - 192.168.3.1 - который является шлюзом нашего провайдера, затем еще раз выполняет поиск и находит связанный с ним интерфейс выхода - ether5. После чего пакет отправляется провайдеру, а удаленный узел 1.1.1.1 используется нами только как способ проверки наличия интернета через данный канал. Это и есть рекурсивная маршрутизация.

Практика

Будем считать что на вашем роутере уже настроен доступ к двум провайдерам и мы не будем останавливаться на этапе базовой настройки, если вы испытываете затруднения с этим, то обратитесь к нашей статье: [Базовая настройка роутера MikroTik](#).

В нашем примере будут использоваться два условных провайдера, основной и резервный, подключенные к интерфейсам **ether5** и **ether4**, тип интерфейса и подключения в данном случае никакой роли не играют. Откроем таблицу маршрутизации **IP - Routes**, в простейшем виде она будет выглядеть приблизительно так:



	Dest. Address	Gateway	Distance	Routing Mark	Pref. Source
DAS	0.0.0.0/0	192.168.3.1 reachable ether5	1		
DS	0.0.0.0/0	192.168.16.2 reachable ether4	2		
DAC	192.168.3.0/24	ether5 reachable	0		192.168.3.113
DAC	192.168.16.0/24	ether4 reachable	0		192.168.16.137
DAC	192.168.186.0/24	bridge1 reachable	0		192.168.186.1

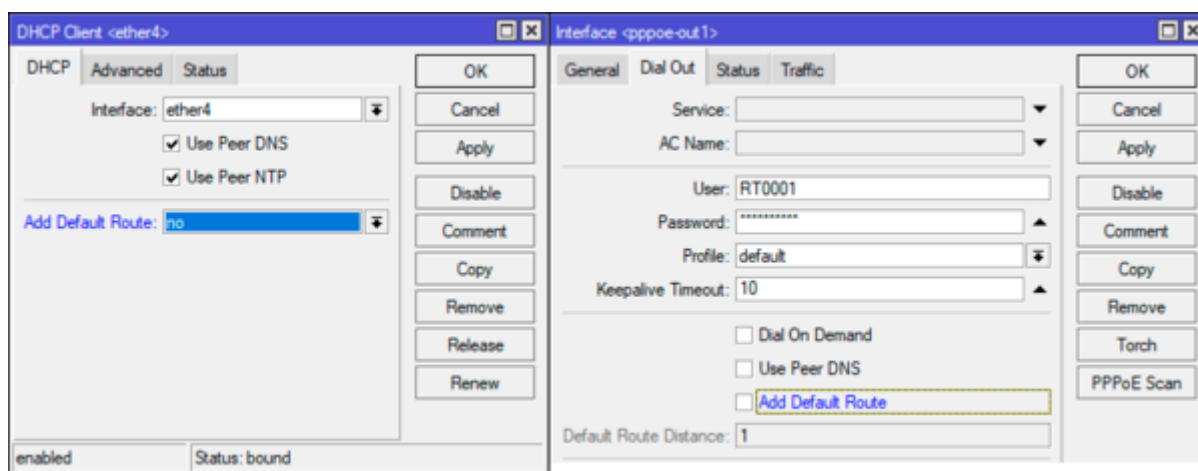
У нас есть два нулевых маршрута с разными административными дистанциями (1 для основного провайдера, 2 для резервного) и три маршрута для непосредственно присоединенных сетей, где 192.168.3.0/24 - сеть основного провайдера, а 192.168.16.0/24 - резервного.

Перед тем, как переходить к дальнейшим настройкам нам нужно выяснить и запомнить адреса шлюзов провайдеров, для этого можно заглянуть на закладку **Nexthops**, для коммутируемых подключений (VPN, PPPoE) следует проверить свойства соответствующего соединения.

Route List							
<div>Routes</div> <div>Nexthops</div> <div>Rules</div> <div>VRF</div>							
<div> <div></div> </div>							
Address	Gateway St...	Forwarding N...	Interface	Scope	Check Gat...	Table	
192.168.3.1	reachable			10			
192.168.16.2	reachable			10			

Внимание! Дальнейшие действия следует производить имея физический доступ к устройству!

Прежде всего отключим автоматическое создание нулевых маршрутов для сетей провайдеров, для этого следует **снять** флаг **Add Default Route** в свойствах подключения, либо установить одноименную опцию в значение - **no**.



После этого доступ в интернет по обоим каналам пропадет. Имейте это ввиду при планировании работ.

Теперь начнем добавлять собственные маршруты, но сначала выберем два высокодоступных узла для первого и второго провайдера. Допустим это будут **1.1.1.1** (Cloudflare DNS) и **9.9.9.9** (Quad9 DNS). Каких-либо ограничений по их выбору нет, это могут быть как ваши собственные, так и публичные узлы, единственное условие - они не должны использоваться в вашей сети, потому как при отказе одного из провайдеров окажутся недоступными.

В первую очередь создадим маршруты к этим узлам. Для этого снова перейдем в **IP - Routes** и создадим новый маршрут:

Где **Dst. Address** - 1.1.1.1 - узел для проверки первого провайдера, **Gateway** - 192.168.3.1 - шлюз первого провайдера, **Distance** - 1, **Scope** - 10. Обратите внимание на значение области (Scope), если вы оставите там значение по умолчанию - 30, то рекурсивная маршрутизация работать не будет! Затем добавим второй аналогичный маршрут, но уже к узлу 9.9.9.9 через шлюз второго провайдера.

Эти же действия через терминал:

```
/ip route
add distance=1 dst-address=1.1.1.1/32 gateway=192.168.3.1 scope=10
add distance=1 dst-address=9.9.9.9/32 gateway=192.168.16.2 scope=10
```

Затем добавим два рекурсивных "нулевых" маршрута. Для первого провайдера:

Dst. Address оставляем по умолчанию - **0.0.0.0/0**, **Gateway** - **1.1.1.1** - высокодоступный узел для первого провайдера, **Check Gateway** - **ping** - указываем проверку доступности шлюза, **Distance** - **1**.

Для второго провайдера:

Dst. Address - **0.0.0.0/0**, **Gateway** - **9.9.9.9**, **Check Gateway** - **ping**, **Distance** - **2**.

Обратите внимание на значение административной дистанции второго маршрута, она должна быть больше, чем у основного провайдера, в нашем случае - 2.

То же самое быстро в терминале:

```
/ip route
add check-gateway=ping distance=1 gateway=1.1.1.1
add check-gateway=ping distance=2 gateway=9.9.9.9
```

Теперь таблица маршрутизации будет выглядеть следующим образом:

Route List								
Routes		Nexthops	Rules	VRF				
+	-	✓	✗	📄	🔍			
	Dst. Address	Gateway	Distance	Scope	Target Scope	Routing Mark	Pref.	Source
AS	▶ 0.0.0.0/0	1.1.1.1 recursive via 192.168.3.1 ether5	1	30	10			
S	▶ 0.0.0.0/0	9.9.9.9 recursive via 192.168.16.2 ether4	2	30	10			
AS	▶ 1.1.1.1	192.168.3.1 reachable ether5	1	10	10			
AS	▶ 9.9.9.9	192.168.16.2 reachable ether4	1	10	10			
DAC	▶ 192.168.3.0/24	ether5 reachable	0	10	10			192.168.3.113
DAC	▶ 192.168.16.0/24	ether4 reachable	0	10	10			192.168.16.137
DAC	▶ 192.168.186.0/24	bridge1 reachable	0	10	10			192.168.186.1

Наши маршруты добавились как рекурсивные и активным является маршрут через основного провайдера. Теперь имитируем аварию у первого провайдера, буквально через 10-20 секунд таблица маршрутизации изменится и рабочим станет маршрут через резервный канал:

Route List							
Routes							
Next Hops							
Rules							
VRF							
	Dest. Address	Gateway	Distance	Scope	Target Scope	Routing Mark	Pref. Source
S	0.0.0.0/0	1.1.1.1 recursive via 192.168.3.1 ether5	1	30	10		
AS	0.0.0.0/0	9.9.9.9 recursive via 192.168.16.2 ether4	2	30	10		
AS	1.1.1.1	192.168.3.1 reachable ether5	1	10	10		
AS	9.9.9.9	192.168.16.2 reachable ether4	1	10	10		
DAC	192.168.3.0/24	ether5 reachable	0	10	10		192.168.3.113
DAC	192.168.16.0/24	ether4 reachable	0	10	10		192.168.16.137
DAC	192.168.186.0/24	bridge1 reachable	0	10	10		192.168.186.1

После того, как первый провайдер устранит неисправность снова произойдет изменение маршрутов и рабочим снова станет основной канал.

Как видим на практике все оказалось достаточно просто, но мы уверены, что слепое повторение инструкций без понимания сути производимых действий никого и никогда ни к чему хорошему не приводило, поэтому настоятельно советуем не пренебрегать теоретической частью материала.

Рекурсивная маршрутизация и PPPoE

При использовании PPPoE подключения вы рано или поздно столкнетесь с проблемой изменения адреса шлюза провайдера, это может произойти даже при наличии выделенного IP адреса. После чего маршрут к высокодоступному узлу перестает работать, что приводит к полной неработоспособности этого провайдера. Диагностика такой неисправности может оказаться весьма затруднительной если вы не сталкивались с данной проблемой ранее и недостаточно четко представляете себе принципы работы рекурсивных маршрутов.

Чтобы этого избежать используем одну небольшую хитрость. Вспомним, о чем мы говорили в теоретической части. При использовании PPPoE соединения адрес шлюза провайдера роутеру как таковой не нужен. Он используется только для определения интерфейса выхода. Для работы протокола PPP, который лежит в основе PPPoE, IP-адреса не требуются. Это дает возможность самостоятельно присвоить произвольный адрес для удаленного конца туннеля и использовать его в качестве шлюза.

Для этого перейдем в **PPP - Profiles** и создадим новый профиль, укажем для него понятное имя и зададим параметр - **Remote Address**, адрес можно указать любой, единственное условие - он не должен пересекаться с вашими внутренними сетями, в нашем случае это **10.253.252.251**. Также установите переключатель **Change TCP MSS** в положение **yes**.

В терминале это можно сделать так:

```
/ppp profile
add change-tcp-mss=yes name=pppoe-rt remote-address=10.253.252.251
```

После чего назначим этот профиль вашему PPPoE соединению:

И после его перезапуска убедимся, что удаленный адрес соединения теперь соответствует назначенному нами:

Теперь можно использовать данный адрес как шлюз для этого провайдера, не беспокоясь о возможных изменениях.

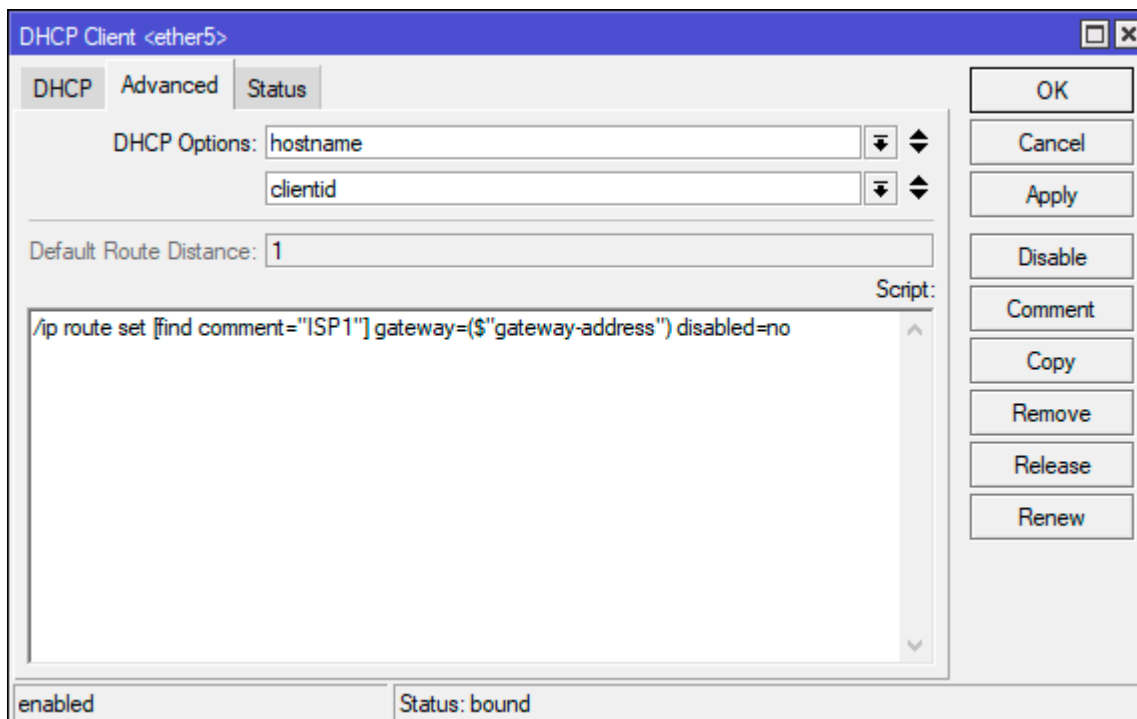
Рекурсивная маршрутизация и DHCP

Описанная выше проблема актуальна и для провайдеров, выдающих настройки по DHCP, в том случае если у клиента нет выделенного IP-адреса, текущий адрес может быть назначен из нескольких диапазонов, а следовательно, будет изменен и адрес шлюза. В отличие от PPPoE здесь мы не сможем задать произвольный адрес, поэтому нам на выручку придут скрипты.

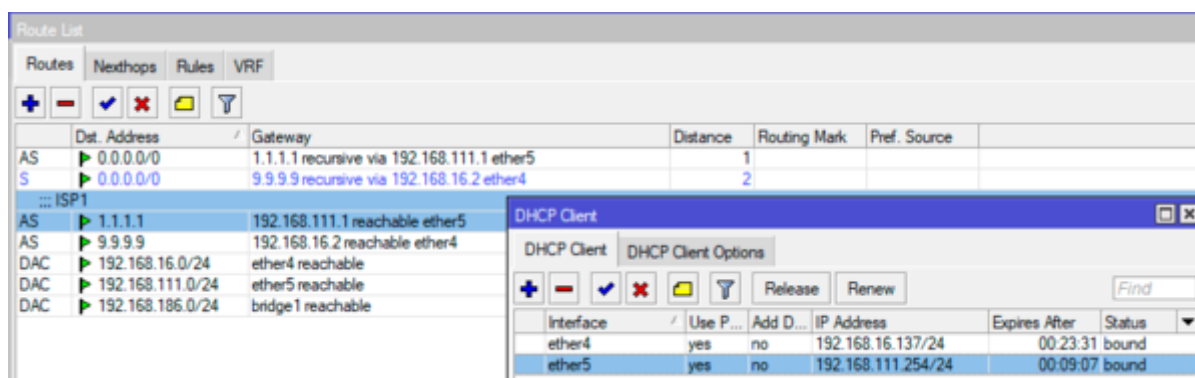
Прежде всего добавим к маршруту для высокодоступного узла через шлюз провайдера уникальный комментарий, например, **ISP1**.

Затем перейдем в настройки DHCP-клиента - **IP - DHCP Client** - и на закладке **Advanced** внесем в соответствующее поле короткий скрипт:

```
/ip route set [find comment="ISP1"] gateway=("$gateway-address") disabled=no
```



Теперь при каждом новом получении адреса по DHCP скрипт будет находить наш маршрут и изменять в нем значение шлюза на реально полученный адрес. Попробуем смоделировать данную ситуацию и выдадим нашему роутеру адрес первого провайдера из другого диапазона. Как видим, все автоматически изменилось согласно вновь полученных настроек:



Таким образом мы снова успешно решили задачу поддержания маршрутной информации в актуальном состоянии несмотря на ее возможные изменения со стороны провайдера.

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.

