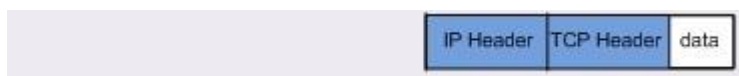


Настройка GRE over IPSEC

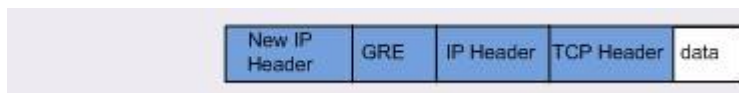
В работе будет рассмотрена конфигурация GRE туннеля поверх IPSEC на маршрутизаторе с RouterOS. GRE – это туннельный протокол разработанный компанией Cisco. Основная задача – инкапсуляция большого количества протоколов через виртуальный point-to-point линк. Работает на сетевом уровне TCP/IP и не имеет порта. Для лучшего понимания можно провести аналогию его с протоколами без отслеживания состояния, а именно IPIP и EoIP.

Это означает, что, если ваш линк на одной из сторон окажется недоступным, вы не сможете этого понять, находясь на другой стороне – это классическая работа подобных протоколов. Но компания Mikrotik добавила компонент keepralive. Теперь можно отслеживать состояние линка отправляя тем самые keepralive запросы. Отсутствует шифрование по понятным причинам, но это не проблема, поскольку дополнительно будет использован IPSEC. GRE туннель может пересылать только IPv4 и IPv6 пакеты. Не используйте «Check gateway» опцию «arp».

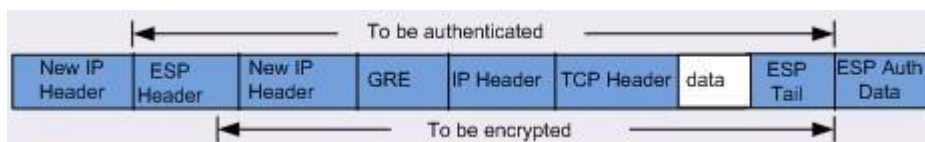
Взглянем на обычный IP пакет.



Теперь посмотрим с GRE.



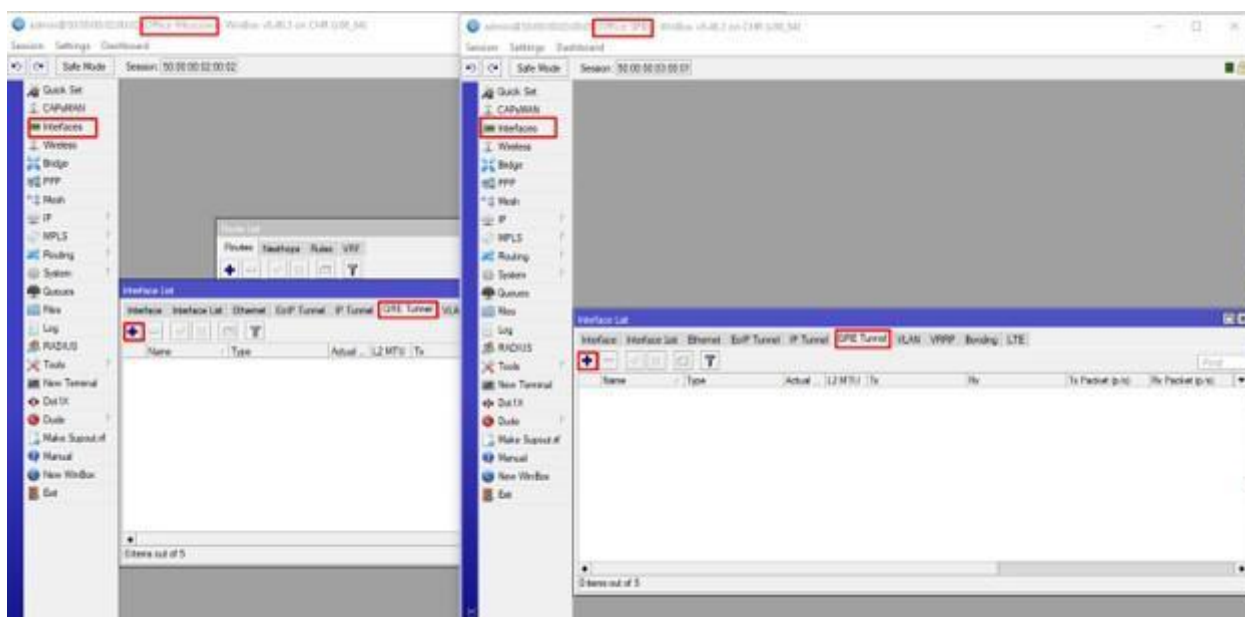
И еще с IPSEC в туннельном режиме.



Настройка GRE

Первым шагом рассмотрим простую настройку gre туннеля. Вся она будет одинакова на обоих роутерах. Подключившись к маршрутизаторам через Winbox и перейдем в Interfaces – GRE Tunnel.

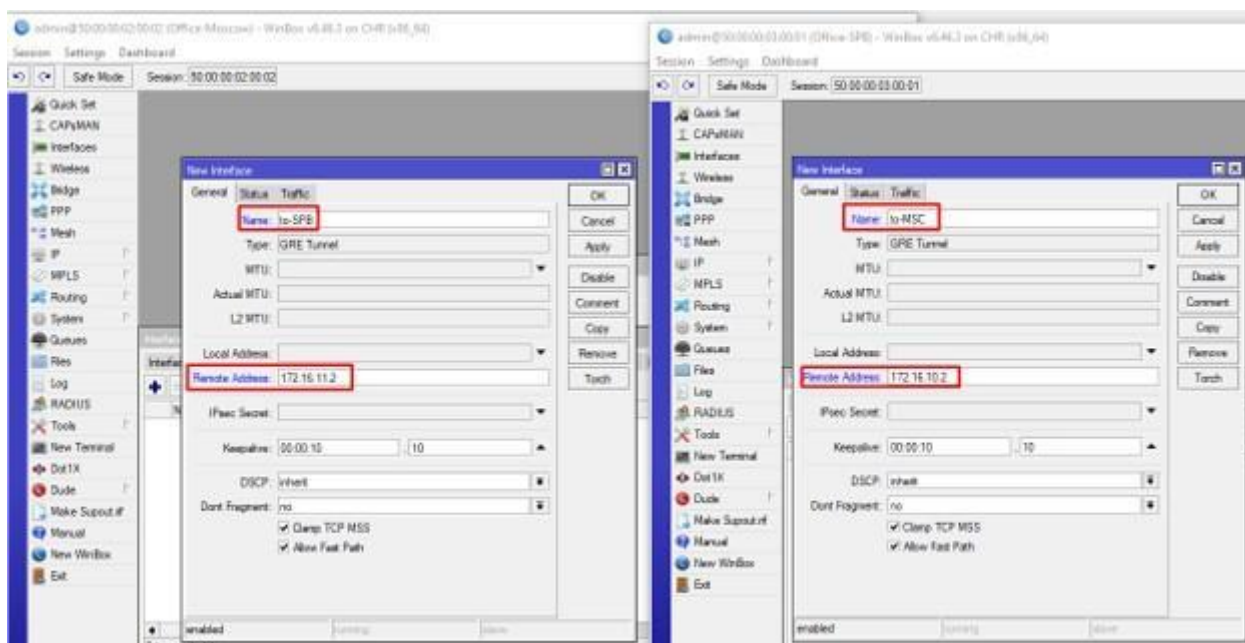
Для работы используются два устройства Mikrotik CHR или RouterOS, аналогично предыдущим лабораторным работам.



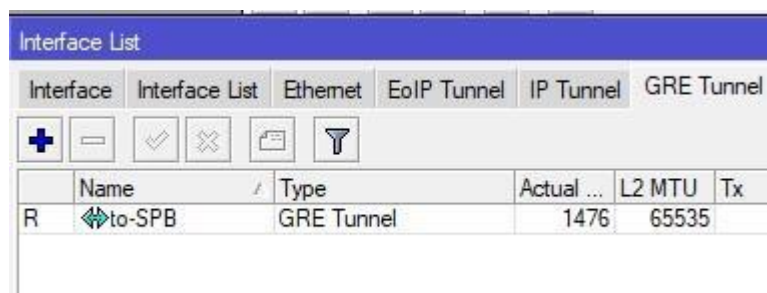
Создадим по интерфейсу. Укажем следующие параметры:

- Name – понятное имя;
- Remote Address – адрес соседа по туннелю. Основной принцип — направляем роутеры друг на друга;

Keepalive – тот самый параметр отслеживания состояния. Можно ничего не менять. Он означает следующее – если в течении 10 попыток по 10 секунд не отвечает удаленная сторона, считать туннель не активным.

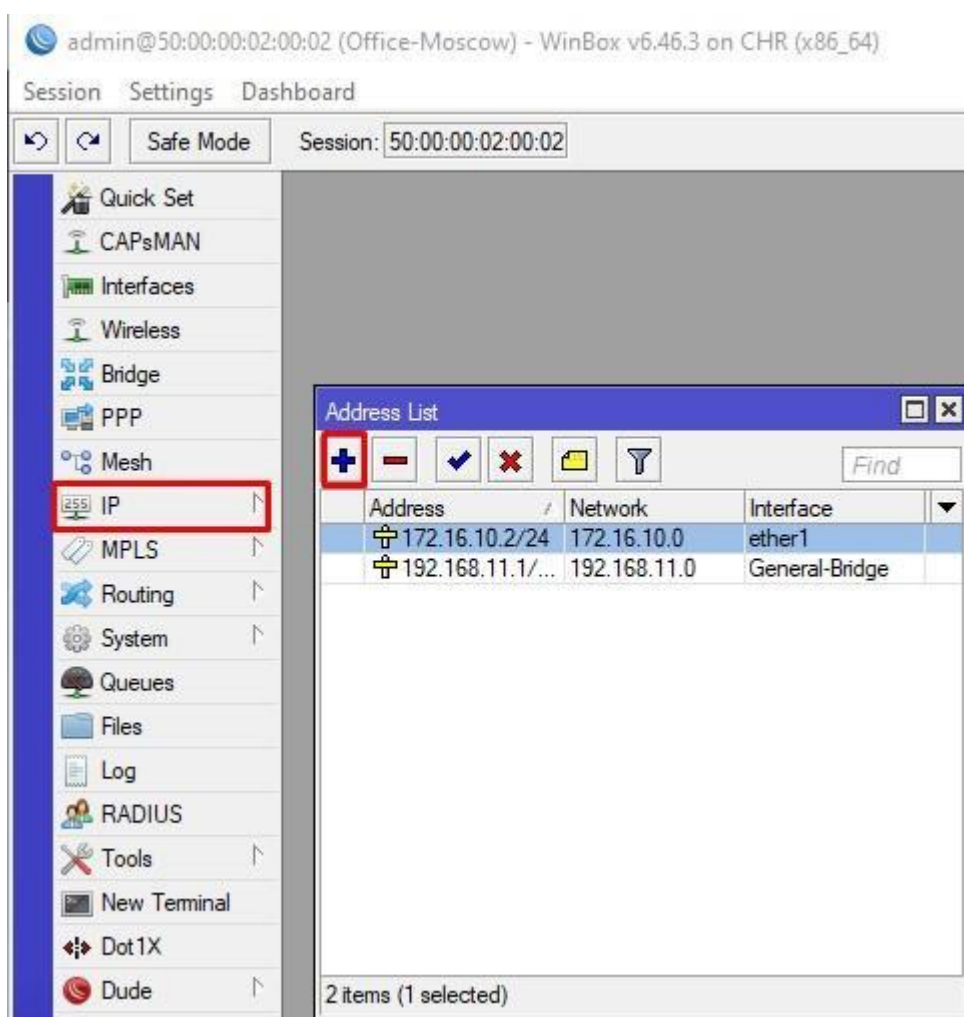


Сохраняем настройки и смотрим на состояние.



Name	Type	Actual ...	L2 MTU	Tx
R to-SPB	GRE Tunnel	1476	65535	

Как мы видим, интерфейс в состоянии running. Назначим адреса. Переходим в IP – Addresses.



admin@50:00:00:02:00:02 (Office-Moscow) - WinBox v6.46.3 on CHR (x86_64)

Session Settings Dashboard

Safe Mode Session: 50:00:00:02:00:02

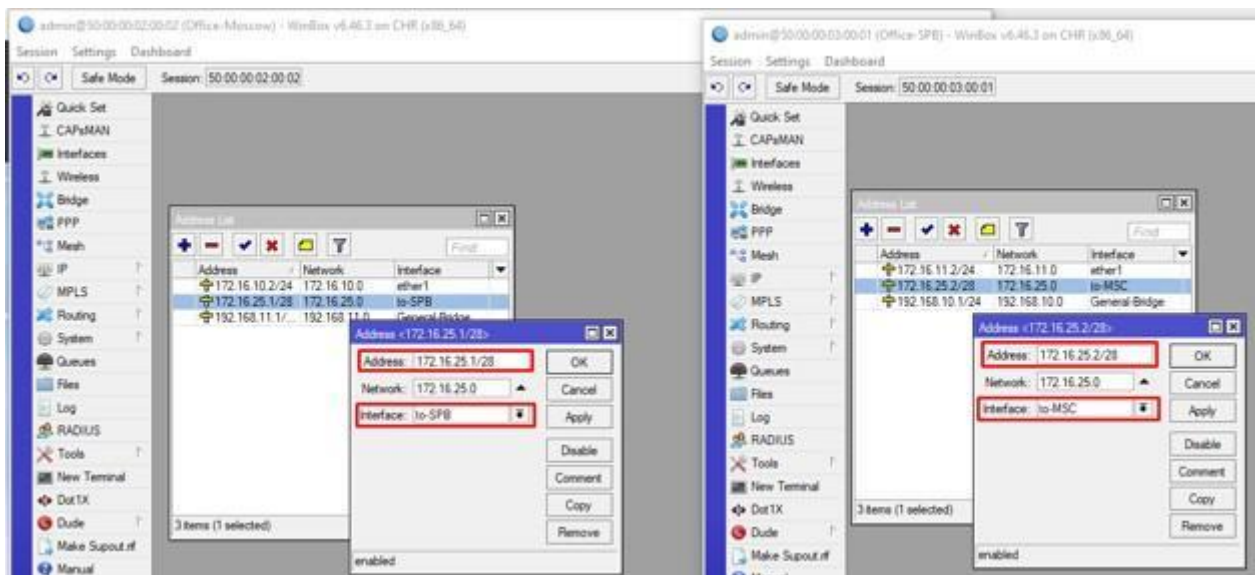
Quick Set
CAPsMAN
Interfaces
Wireless
Bridge
PPP
Mesh
IP
MPLS
Routing
System
Queues
Files
Log
RADIUS
Tools
New Terminal
Dot1X
Dude

Address List

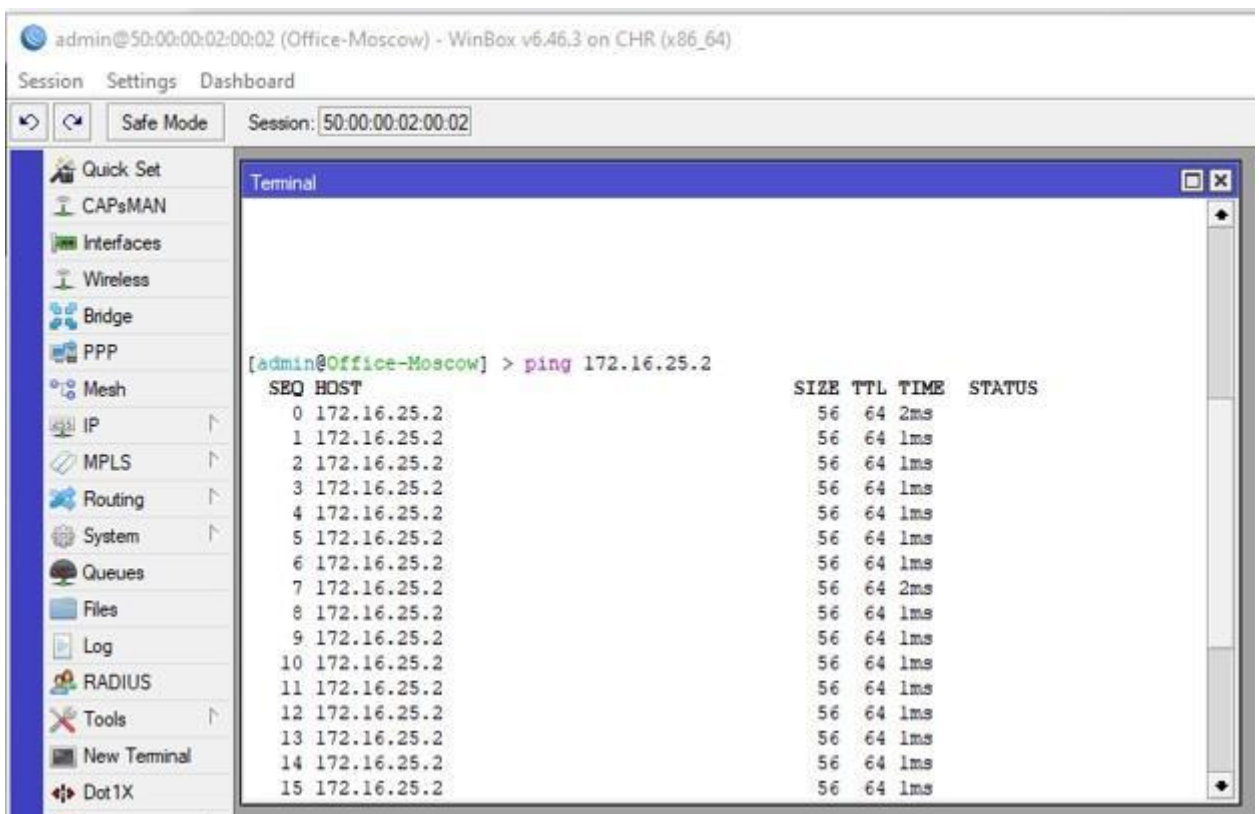
Address	Network	Interface
172.16.10.2/24	172.16.10.0	ether1
192.168.11.1/...	192.168.11.0	General-Bridge

2 items (1 selected)

Зададим адрес 172.16.25.1 для первого роутера и 172.16.26.2 для второго.



Для проверки связи запустим ping запросы.



Пинги идут – все хорошо. Далее пропишем маршруты в локальные сети. Открываем IP – Routes на на первом маршрутизаторе.

	Dst. Address	Gateway	Distance	Routing Mark	Pref. Source
AS	0.0.0.0/0	172.16.10.1 reachable ether1	1		
DAC	172.16.10.0/24	ether1 reachable	0		172.16.10.2
DAC	172.16.25.0/28	to-SPB reachable	0		172.16.25.1
DAC	192.168.11.0/24	General-Bridge reachable	0		192.168.11.1

4 items

Добавляем новый маршрут:

- Dst. Address – 192.168.10.0/24;
- Gateway – 172.16.25.2.

New Route

General | Attributes

Dst. Address: 192.168.10.0/24

Gateway: 172.16.25.2

Check Gateway: ☐

Type: unicast

Distance:

Scope: 30

Target Scope: 10

Routing Mark:

Pref. Source:

OK
Cancel
Apply
Disable
Comment
Copy
Remove

enabled active

Сохраняем и проделываем аналогичную операцию на втором роутере.

Route <192.168.11.0/24>

General Attributes

Dst. Address: 192.168.11.0/24

Gateway: 172.16.25.1 reachable to-MS

Check Gateway: Type: unicast

Distance: 1

Scope: 30

Target Scope: 10

Routing Mark:

Pref. Source:

enabled active

OK Cancel Apply Disable Comment Copy Remove

Основной принцип – прописать маршруты в сети через адреса в туннелях.
 Проверим ping до адреса бриджа Mikrotik к первому роутеру.

admin@50:00:00:03:00:01 (Office-SPB) - WinBox v6.46.3 on CHR (x86_64)

Session Settings Dashboard

Safe Mode Session: 50:00:00:03:00:01

Quick Set CAPsMAN Interfaces Wireless Bridge PPP Mesh IP MPLS Routing System Queues Files Log RADIUS Tools New Terminal Dot1X

Terminal

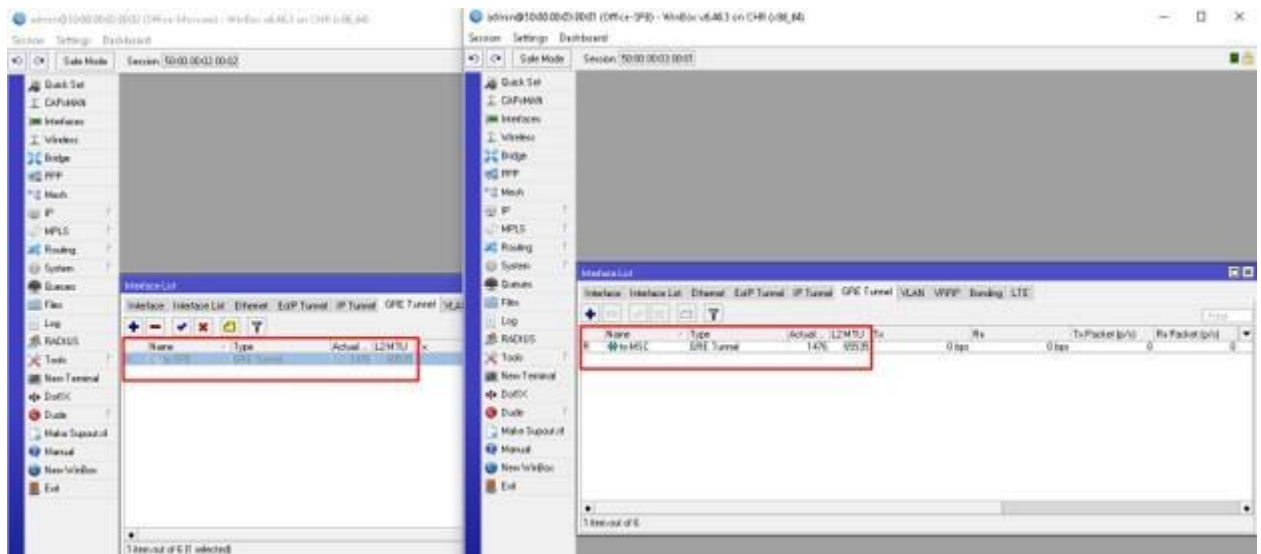
```
[admin@Office-SPB] > ping 192.168.11.1
```

SEQ	HOST	SIZE	TTL	TIME	STATUS
0	192.168.11.1	56	64	1ms	
1	192.168.11.1	56	64	1ms	
2	192.168.11.1	56	64	2ms	
3	192.168.11.1	56	64	1ms	
4	192.168.11.1	56	64	1ms	
5	192.168.11.1	56	64	1ms	
6	192.168.11.1	56	64	1ms	

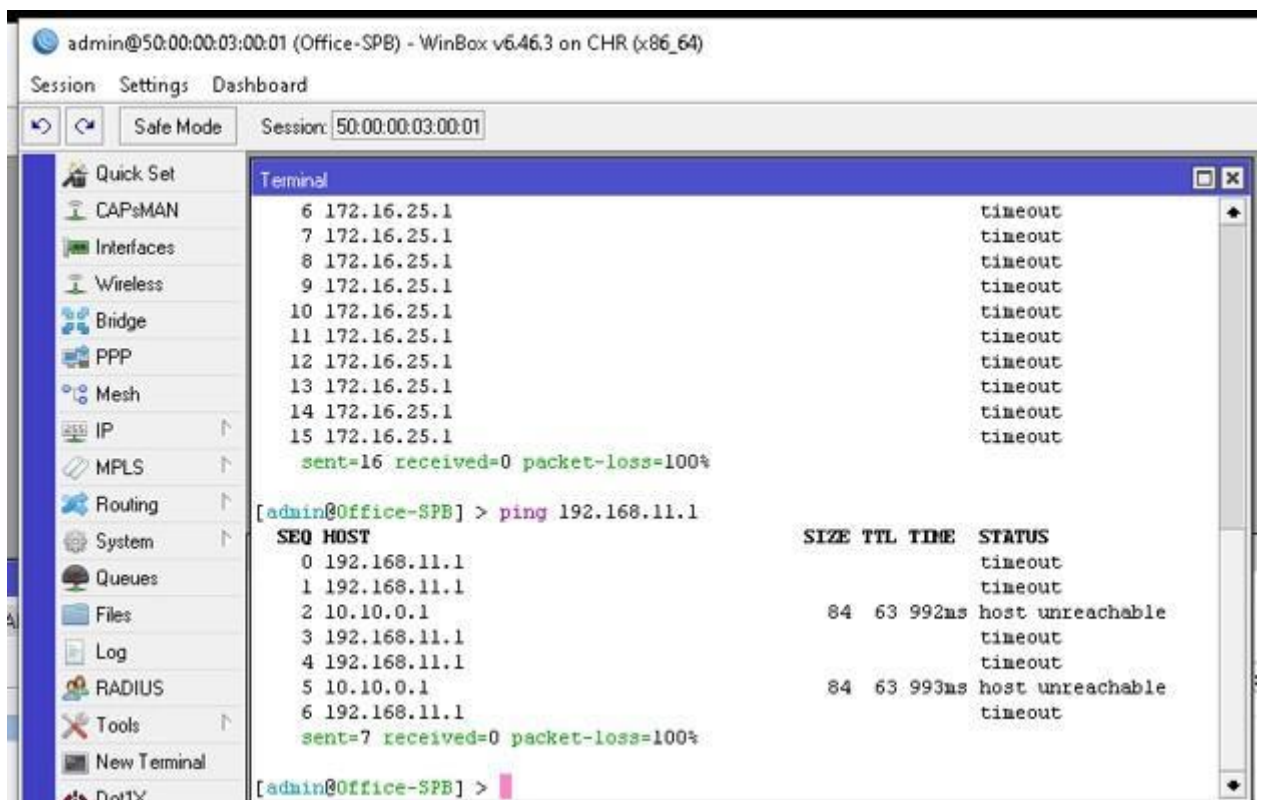
sent=7 received=7 packet-loss=0% min-rtt=1ms avg-rtt=1ms max-rtt=2ms

[admin@Office-SPB] >

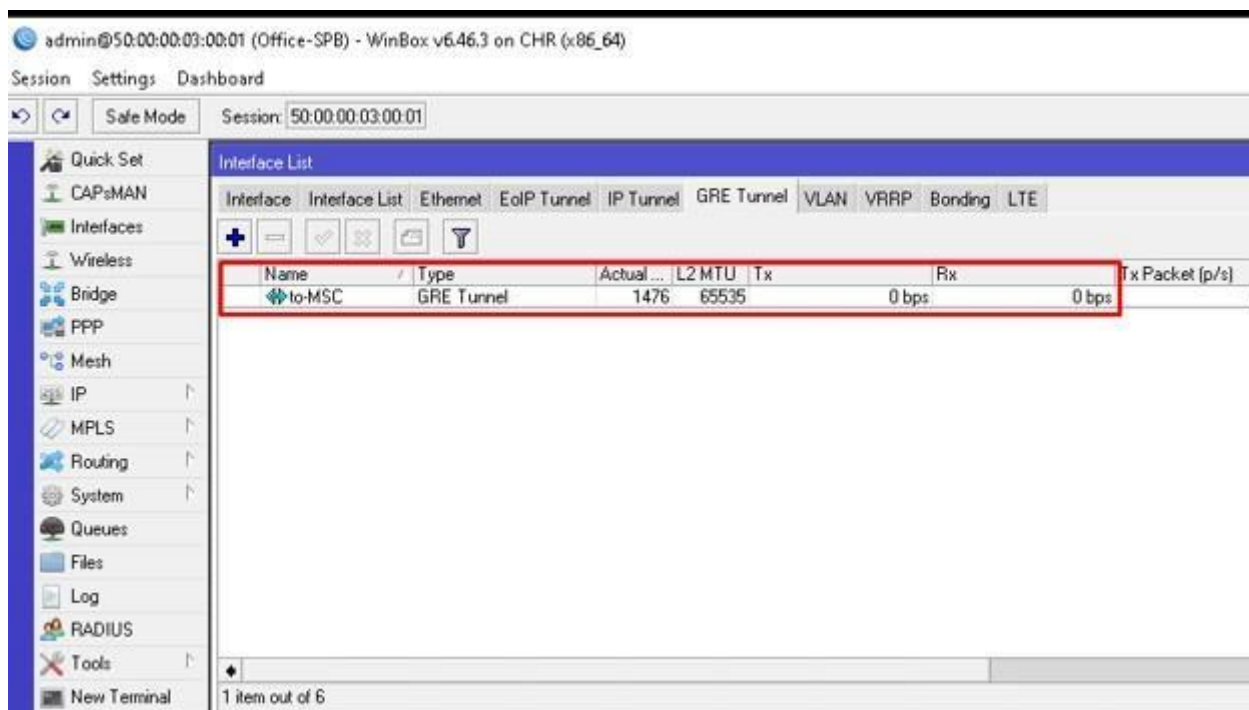
Если все проходит – конфигурация выполнена правильно. В целях демонстрации отключим интерфейс на первом роутере, после интервала в 100 секунд посмотрим на состояние туннеля.



Интересная ситуация, в одном офисе интерфейс активный, а в другом нет. Попробуем проверить связь.



Пинга нет, а туннель активен. Спустя какое-то время, второй Mikrotik понимает, что связи через gre туннель нет и меняет статус на интерфейсе.

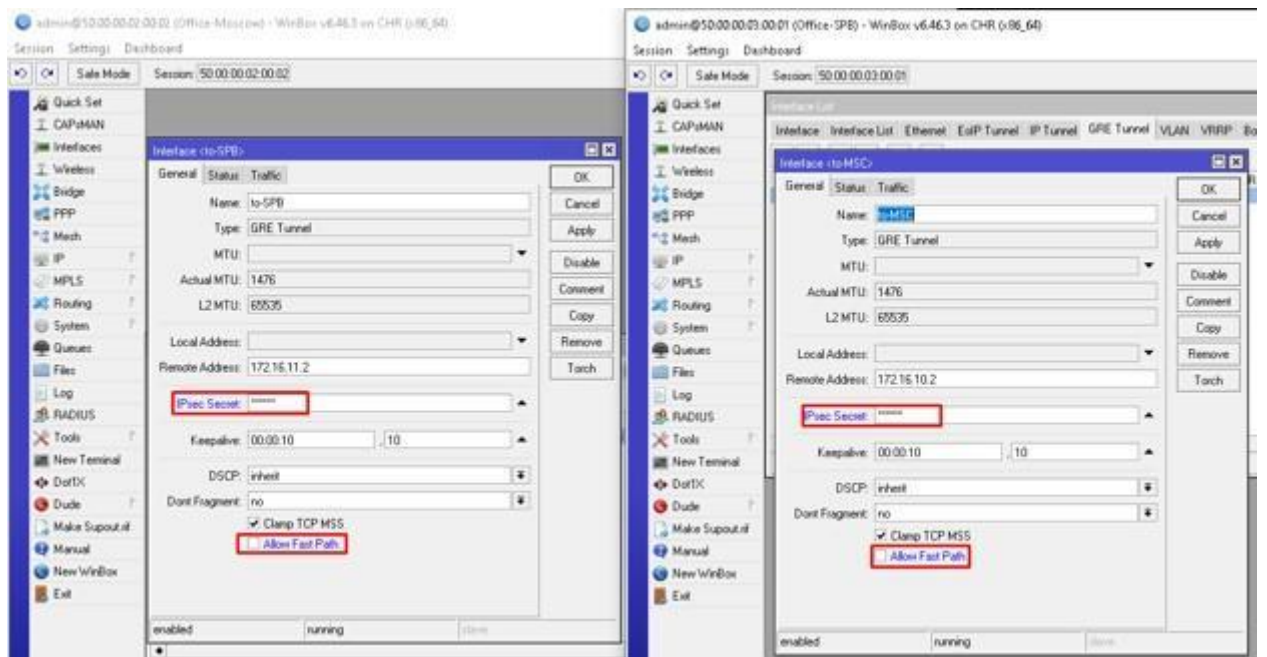


Тут-то и помогла доработка Mikrotik с keepalive. Маршруты в routes будут активны то время, которое вы указали в keepalive. Только по истечении этого времени маршруты и адрес станут не активны.

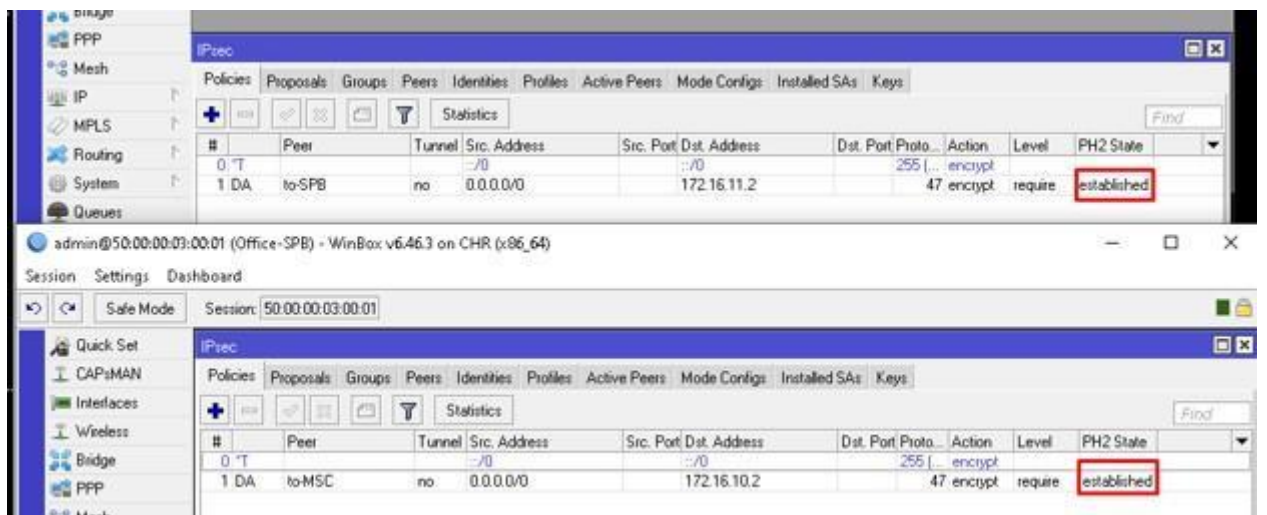
Настройка GRE IPSEC

Вернем активное состояние интерфейса на первом роутере. Далее переходим в настройки интерфейса и меняем следующие параметры:

- IPsec Secret – общий ключ (пароль);
- Allow Fast Path – снимаем галочку.



Сохраняем и проверяем.



Указав общий ключ, наши микротики согласуют стандартный IPSEC (не IKEv2) и инкапсулирует в него GRE. На этом настройка завершена.

Контрольные вопросы:

1. Назовите основные функции протокола GRE;
2. Укажите формат заголовка GRE;
3. От каких параметров зависит степень безопасности, обеспечиваемая VPN протоколами?

4. Назовите основные способы аутентификации пользователей в протоколах VPN на примере PPPoE;
5. Назовите основные функции протокола IPSec, способы аутентификации в IPSec.