Лабораторная работа: построение туннелей на маршрутизаторах Cisco с использованием технологии VRF

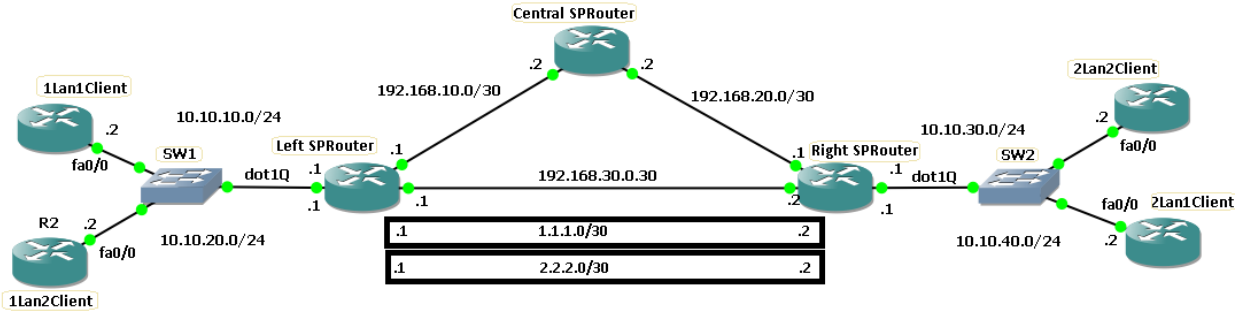
Описание задачи

Моделируется ситуация, когда двум клиентам провайдера необходимо логически соединить свои удалённые сети так, чтобы трафик одного клиента никак не пересекался с трафиком другого клиента. Для этого необходимо настроить виртуальные устройства на базе граничных маршрутизаторов провайдера, используя технологию VRF, и поднять GRE-туннели с концами на соответствующих виртуальных маршрутизаторах.

Описание решения

Для начала необходимо разобраться с понятием VRF. VRF (Virtual Routing and Forwarding) - механизм создания виртуальных маршрутизаторов на базе одного физического устройства. Из плюсов этого механизма можно отметить практически полную независимость таблиц маршрутизации и настроек разных виртуальных устройств. Отсюда и следует очевидное применение: если имеется большая операторская сеть, к которой нужно подключить некоторое количество новых клиентов, со специфичными настройками (новый DHCP сервер или шлюз по умолчанию) достаточно создать виртуальное устройство и настроить его соответствующим образом. В нашем случае VRF применяется для разделения трафика различных клиентов и построения необходимой логической топологии.

Из условия поставленной задачи необходимо реализовать следующую сеть:



Здесь маршрутизаторы *LeftSPRouter, CentralSPRouter, RightSPRouter* моделируют сеть провайдера, *1Lan1Client, 2Lan1Client* – удаленные сети первого клиента, *1Lan2Client, 2Lan2Client* – второго клиента. Для сети провайдера используем маршрутизаторы Cisco серии 7200, офисные сети реализуются маршрутизаторами Cisco серии 3600.

Моделирование

Добавим необходимые маршрутизаторы и коммутаторы. Настроим коммутаторы: на SW1 интерфейс 1 находится в режиме trunk (на коммутаторе в GNS3 такой режим называется dot1q), интерфейс 2 в VLAN 2 в режиме access, интерфейс 3 в VLAN 3 в режиме access; на

SW2 аналогичные настройки. Соединим устройства как на схеме, причем *1Lan1Client* подключим ко второму интерфейсу SW1, *1Lan2Client* – к третьему интерфейсу SW1; *2Lan2Client* – к третьему интерфейсу SW2, *2Lan1Client* – ко второму интерфейсу SW2. На этом моделирование закончено, переходим к настройке.

Настройка

Настроим предложенную схему для первого клиента. Для этого необходимо создать виртуальные устройства на маршрутизаторах, затем настроить все необходимые интерфейсы на устройствах, после чего поднять GRE-туннель на loopback-ах и, наконец, настроить динамическую маршрутизацию.

Сначала нужно настроить VRF-маршрутизаторы. На *LeftSPRouter* в режиме глобального конфигурирования добавим новый виртуальный маршрутизатор.

```
LeftSPRouter(config)# ip vrf Client1vrf
```

Присвоим ему уникальный идентификатор.

```
LeftSPRouter(config-vrf)# rd 1:1
```

Проделаем то же самое на *RightSPRouter*.

```
RightSPRouter(config)# ip vrf Client1vrf
RightSPRouter(config-vrf)#rd 1:2
```

Далее, настроим *1Lan1Client*. Перейдем в режим конфигурирования интерфейса в сторону коммутатора (в нашем случае он называется fa0/0) и присвоим ему IP-адрес с маской командой **ip address** *10.10.10.2 255.255.255.0*. Включим интерфейс командой **no shutdown**. Кроме того, создадим loopback-интерфейс, моделирующий сеть клиента.

```
1Lan1Client(config)# int loopback 1
1Lan1Client(config-if)# ip address 10.10.11.1 255.255.255.0
```

Переходим к настройке *LeftSPRouter*. В режиме конфигурирования интерфейса в сторону *CentralSPRouter* присвоим ему IP-адрес с маской командой **ip address** *192.168.10.1 255.255.255.252*. Поднимем интерфейс командой **no shutdown**.

То же самое проделаем для интерфейса в сторону *RightSPRouter*.

```
LeftSPRouter(config-if)# ip address 192.168.30.1 255.255.255.252
LeftSPRouter(config-if)# no shutdown
```

Теперь настроим интерфейс в сторону коммутатора SW1. Поднимем подынтерфейс, добавим его в наш виртуальный маршрутизатор, настроим инкапсуляцию, присвоим IP-адрес и маску.

```
LeftSPRouter(config)# int fa0/0.2
LeftSPRouter(config-if)# ip vrf forwarding Client1vrf
LeftSPRouter(config-if)# encapsulation dot1Q 2
LeftSPRouter(config)# ip address 10.10.10.1 255.255.255.0
```

Переходим к настройке *CentralSPRouter*.

```
CentralSPRouter(config)# int fa0/0
CentralSPRouter(config-if)# ip address 192.168.10.2 255.255.255.252
CentralSPRouter(config-if)# no shutdown
CentralSPRouter(config)# int fa0/1
CentralSPRouter(config-if)# ip address 192.168.20.2 255.255.255.252
CentralSPRouter(config-if)# no shutdown
```

Теперь *RightSPRouter*. Аналогичные настройки, что и для *LeftSPRouter*. Настроим и поднимем интерфейсы в сторону *LeftSPRouter* и *CentralSPRouter* командами **ip address** *192.168.20.1 255.255.255.252* и **ip address** *192.168.30.2 255.255.255.252* соответственно. Теперь настроим интерфейс в сторону коммутатора SW2. Поднимем подынтерфейс, добавим его в наш виртуальный маршрутизатор, настроим инкапсуляцию, присвоим IP-адрес и маску.

```
RightSPRouter(config)# int fa0/0.2
RightSPRouter(config-if)# ip vrf forwarding Client1vrf
RightSPRouter(config-if)# encapsulation dot1Q 2
RightSPRouter(config)# ip address 10.10.40.1 255.255.255.0
```

Осталось настроить *2Lan1Client*. Перейдем в режим конфигурирования интерфейса в сторону коммутатора и присвоим ему IP-адрес с маской командой **ip address** *10.10.40.2 255.255.255.0*. Поднимем интерфейс командой **no shutdown**. Кроме того, создадим loopback-интерфейс, моделирующий сеть клиента.

```
2Lan1Client(config)# int loopback 1
2Lan1Client(config-if)# ip address 10.10.41.1 255.255.255.0
```

На этом первый этап настройки завершен. Теперь необходимо настроить GRE-туннель на интерфейсах loopback между *RightSPRouter* и *LeftSPRouter*. В режиме глобальной конфигурации *LeftSPRouter* добавляем интерфейс loopback1.

```
LeftSPRouter(config)# int loopback1
LeftSPRouter(config-if)# ip address 1.1.3.1 255.255.255.252
```

Настройка туннеля: поднимаем туннельный интерфейс, добавляем его в наш виртуальный маршрутизатор, настраиваем туннель.

```
LeftSPRouter(config)# int tunnel1
LeftSPRouter(config-if)# ip vrf forwarding Client1vrf
LeftSPRouter(config-if)# ip address 1.1.1.1 255.255.255.252
LeftSPRouter(config-if)# tunnel source loopback1
LeftSPRouter(config-if)# tunnel destination 1.1.4.1
LeftSPRouter(config-if)# tunnel key 1
```

Предпоследняя команда указывает адрес другого конца туннеля, который мы настроим далее, а последняя команда необходима для идентификации туннеля.

Поясним назначение команды идентификации туннеля *tunnel key*. Часто туннели клиентов поднимаются на одних и тех же интерфейсах (в нашем случае это интерфейсы loopback) из-за чего возникает неоднозначность определения принадлежности входящего пакета тому или иному туннелю. В этом легко убедиться самостоятельно: без указания ключа функционировать будет только один туннель (тот, что был настроен последним), т.е. в нашем случае после настройки туннеля второго клиента на тех же loopback-интерфесах туннель первого клиента отключится. Решить проблему неоднозначности можно, если

настроить туннели разных клиентов на разных интерфейсах, что ресурсоемко. Проще указать ключ идентификации туннеля, что и было сделано.

Теперь настроим *RightSPRouter*.

```
RightSPRouter(config)# int loopback1
RightSPRouter(config-if)# ip address 1.1.4.1 255.255.255.252
```

Настройка туннеля: поднимаем туннельный интерфейс, добавляем его в наш виртуальный маршрутизатор, настраиваем туннель.

```
RightSPRouter(config)# int tunnel1
RightSPRouter(config-if)# ip vrf forwarding Client1vrf
RightSPRouter(config-if)# ip address 1.1.1.2 255.255.255.252
RightSPRouter(config-if)# tunnel source loopback1
RightSPRouter(config-if)# tunnel destination 1.1.3.1
 RightSPRouter(config-if)# tunnel key 1
```

Туннель настроен, но использоваться он не будет, пока не будет настроена динамическая маршрутизация.

Перейдем к последнему этапу настройки. В качестве протокола динамической маршрутизации выберем OSPF.

В сети провайдера на *LeftSPRouter*.

```
LeftSPRouter(config)# router ospf 1
LeftSPRouter(config-router)# network 192.168.10.0 0.0.0.3 area 0
LeftSPRouter(config-router)# network 192.168.30.0 0.0.0.3 area 0
LeftSPRouter(config-router)# network 1.1.3.0 0 0.0.0.3 area 0
```

Теперь для маршрутизации в VRF *Client1vrf* на *LeftSPRouter* выполним нижеследующие команды.

```
LeftSPRouter(config)# router ospf 2 vrf Client1vrf
LeftSPRouter(config-router)# network 10.10.10.0 0.0.0.255 area 0
LeftSPRouter(config-router)# network 1.1.1.0 0.0.0.3 area 0
```

В сети провайдера на *RightSPRouter*.

```
RightSPRouter(config)# router ospf 1
RightSPRouter(config-router)# network 192.168.20.0 0.0.0.3 area 0
RightSPRouter(config-router)# network 192.168.30.0 0.0.0.3 area 0
RightSPRouter(config-router)# network 1.1.4.0 0 0.0.0.3 area 0
```

Теперь для маршрутизации в VRF *Client1vrf* на *RightSPRouter*.

```
RightSPRouter(config)# router ospf 2 vrf Client1vrf
RightSPRouter(config-router)# network 10.10.40.0 0.0.0.255 area 0
RightSPRouter(config-router)# network 1.1.1.0 0.0.0.3 area 0
```

В сети провайдера на *CentralSPRouter*.

```
CentralSPRouter(config)# router ospf 1
CentralSPRouter(config-router)# network 192.168.20.0 0.0.0.3 area 0
CentralSPRouter(config-router)# network 192.168.10.0 0.0.0.3 area 0
```

На *1Lan1Client* (номер OSPF-процесса – номер VLAN).

```
1Lan1Client(config)# router ospf 2
1Lan1Client(config-router)# network 10.10.10.0 0.0.0.255 area 0
1Lan1Client(config-router)# network 10.10.11.0 0.0.0.255 area 0
```

На *2Lan1Client* (номер OSPF-процесса – номер VLAN).

```
2Lan1Client(config)# router ospf 2
2Lan1Client(config-router)# network 10.10.40.0 0.0.0.255 area 0
2Lan1Client(config-router)# network 10.10.41.0 0.0.0.255 area 0
```

На этом настройка для первого клиента завершена.

Настройка для второго клиента почти ничем не отличается. Ниже приводятся настройки на каждом устройстве с пояснениями в трудных местах.

1Lan2Client

```
1Lan2Client(config)# int fa0/0
1Lan2Client(config-if)# ip address 10.10.20.2 255.255.255.0
1Lan2Client(config)# no shutdown
1Lan2Client(config)# int loopback 1
1Lan2Client(config-if)# ip address 10.10.21.1 255.255.255.0

1Lan2Client(config)# router ospf 3

1Lan2Client(config-router)# network 10.10.20.0 0.0.0.255 area 0
1Lan2Client(config-router)# network 10.10.21.0 0.0.0.255 area 0
```

LeftSPRouter

```
LeftSPRouter(config)# ip vrf Client2vrf\\новы й vrf-router
LeftSPRouter(config-vrf)# rd 2:1
LeftSPRouter(config)# int fa0/0.3\\настройка интерфейса в сторону сети клиента
LeftSPRouter(config-if)# ip vrf forwarding Client2vrf
LeftSPRouter(config-if)# encapsulation dot1Q 3
LeftSPRouter(config)# ip address 10.10.20.1 255.255.255.0
LeftSPRouter(config)# int tunnel2\\новый туннель
LeftSPRouter(config-if)# ip vrf forwarding Client2vrf
LeftSPRouter(config-if)# ip address 1.1.2.1 255.255.255.252
LeftSPRouter(config-if)# tunnel source loopback1\\туннель поднимаем на том же loopback, что
LeftSPRouter(config-if)# tunnel destination 1.1.4.1\\и первый
LeftSPRouter(config-if)# tunnel key 2\\здесь пригодится ключ идентификации
LeftSPRouter(config)# router ospf 3 vrf Client2vrf  \\добавляем 1Lan2Client в таблицу Client2vrf
LeftSPRouter(config-router)# network 10.10.20.0 0.0.0.255 area 0
LeftSPRouter(config-router)# network 1.1.2.0 0.0.0.3 area 0
```

RightSPRouter

```
RightSPRouter(config)# ip vrf Client2vrf
RightSPRouter(config-vrf)# rd 2:2
RightSPRouter(config)# int fa0/0.3
RightSPRouter(config-if)# ip vrf forwarding Client2vrf
RightSPRouter(config-if)# encapsulation dot1Q 3
RightSPRouter(config)# ip address 10.10.30.1 255.255.255.0
RightSPRouter(config)# int tunnel2
RightSPRouter(config-if)# ip vrf forwarding Client2vrf
RightSPRouter(config-if)# ip address 1.1.2.2 255.255.255.252
RightSPRouter(config-if)# tunnel source loopback1
RightSPRouter(config-if)# tunnel destination 1.1.3.1
RightSPRouter(config-if)# tunnel key 2
RightSPRouter(config)# router ospf 3 vrf Client2vrf
```

```
RightSPRouter(config-router)# network 10.10.30.0 0.0.0.255 area 0
RightSPRouter(config-router)# network 1.1.2.0 0.0.0.3 area 0
```

2Lan2Client

```
2Lan2Client(config)# int fa0/0
2Lan2Client(config-if)# ip address 10.10.30.2 255.255.255.0
2Lan2Client(config)# no shutdown
2Lan2Client(config)# int loopback 1
2Lan2Client(config-if)# ip address 10.10.31.1 255.255.255.0
2Lan2Client(config)# router ospf 3
2Lan2Client(config-router)# network 10.10.30.0 0.0.0.255 area 0
2Lan2Client(config-router)# network 10.10.31.0 0.0.0.255 area 0
```

CentralSPRouter в настройке не нуждается. На этом конфигурация заканчивается, перейдем к тестированию сети.

Тестирование

1) Для начала с помощью команд *ping 10.10.40.2 source 10.10.11.1* с *1Lan1Client* и *ping 10.10.30.2 source 10.10.41.1* с *1Lan2Client* убедимся, что пакеты успешно ходят по сети. Кроме того, используя *traceroute* с тех же устройств по тем же адресам, изучим пути следования этих пакетов.

2) Далее убедимся, что протокол OSPF функционирует, прописав *show ip protocols* и *show ip route* на всех устройствах сети. Подумайте, как указанные команды позволяют убедиться в правильной работе протокола OSPF?

3) На *LeftSPRouter* и *RightSPRouter* просмотрим результат работы команды *show ip protocols vrf Client1vrf (show ip protocols vrf Client2vrf)*. Проанализируйте полученные данные.

4) На *LeftSPRouter* и *RightSPRouter* изучим результат работы команды *show ip route vrf Client1vrf (show ip route vrf Client2vrf)*.

5) Используя Wireshark, перехватим пакеты на каналах между *LeftSPRouter* и *RightSPRouter*. Проанализируйте результаты перехвата.

6) Наконец, проверим отказоустойчивость сети: отключим канал между *LeftSPRouter* и *RightSPRouter* и убедимся, что сеть по-прежнему функционирует. Укажите, какие из устройств способны обнаружить изменения в сети провайдера.

7) Восстановить работоспособность отключенного в предыдущем пункте канала. Убедитесь в нормализации маршрутизации в сети оператора.

8) Предложите решение, позволяющее передавать пользовательский IPv6 трафик между сетями клиентов так, чтобы не потребовалась перенастройка операторской сети.

9) *Реализуйте предложенное в предыдущем пункте решение.