Динамическая маршрутизация на коммутаторах SNR. Часть 2. BGP

В прошлой части статьи по динамической маршрутизации мы рассмотрели протокол OSPF. А теперь – обещанное нами продолжение о протоколе BGP. Как обычно, глубоко углубляться в теорию мы не будем, т.к. в интернете присутствует большое количество материалов, которые подробно описывают протокол BGP, к примеру здесь и здесь. Мы остановимся лишь на базовых понятиях и принципах работы.

BGP - протокол динамической маршрутизации, основная его часть описана в RFC 4271. Главная задача протокола состоит в том, чтобы обмениваться маршрутами между автономными системами (eBGP), а также в пределах одной системы (iBGP). BGP – один из самых главных протоколов маршрутизации в сети Internet, который поддерживается большинством L3-устройств, в том числе и L3-коммутаторами SNR. В качестве транспортного протокола BGP использует TCP (port 179).

Принцип работы протокола BGP довольно прост. Изначально между двумя маршрутизаторами, каждый из которых имеет свой уникальный номер автономномной системы, устанавливается TCP-соединение. Далее происходит обмен сообщениями для согласования и подтверждения параметров соединения. И затем, для поддержания соединения, происходит передача KEEPALIVE-сообщений. Обмен маршрутами между автономными системами происходит с помощью отправки UPDATE-сообщений, а сами маршруты хранятся в базах маршрутных данных RIB (routing information

Базовая настройка BGP

В качестве самого простого примера возьмем L3-коммутаторы SNR-S300X-24FQ, которые будут выступать в роли пограничных коммутаторов на сети оператора связи. Коммутатор SNR-S4650X-48FQ будет задействован как магистральный “Uplink”. Установим между устройствами 2 BGP-сессии.

[Изображение недоступно]

Конфигурация коммутаторов

```
R1
!
vlan 10
 name BGP_Test
!
interface Vlan10
 ip address 1.1.1.1 255.255.255.252
!
Interface Ethernet1/0/9
 switchport mode trunk
 switchport trunk allowed vlan 10
!
router bgp 10
 bgp router-id 1.1.1.1
 neighbor 1.1.1.2 remote-as 10
R2
!
vlan 10
 name BGP_Test
!
interface Vlan10
 ip address 1.1.1.2 255.255.255.252
!
vlan 20
 name EBGP_Test
!
interface Vlan20
 ip address 2.2.2.1 255.255.255.252
!
Interface Ethernet1/0/9
 switchport mode trunk
 switchport trunk allowed vlan 10
!
Interface Ethernet1/0/11
 switchport mode trunk
 switchport trunk allowed vlan 20
```

```
!
router bgp 10
 bgp router-id 1.1.1.2
 network 1.1.1.0/30
 network 2.2.2.0/30
 neighbor 1.1.1.1 remote-as 10
 neighbor 2.2.2.2 remote-as 20
R3
!
vlan 20
 name EBGP_Test
!
interface Vlan20
 ip address 2.2.2.2 255.255.255.252
!
Interface Ethernet1/0/11
 switchport mode trunk
 switchport trunk allowed vlan 20
!
router bgp 20
 bgp router-id 2.2.2.2
 neighbor 2.2.2.1 remote-as 10
```

Между R1 и R2-коммутаторами сети оператора связи будет установлена iBGP-сессия, а между коммутатором R2 и коммутатором R3, принадлежащему другой AS, условно, магистральным Uplink'ом будет eBGP-сессия. После применения конфигурации, приведенной выше, коммутаторы устанавливают BGP-соседство и анонсируют друг другу указанные маршруты. Процесс установления соединения происходит следующим образом:

1. начальное состояние - IDLE;
2. происходит установка TCP-сессии между устройствами, состояние сессии: ACTIVE;
3. сессия установлена, происходит обмен OPEN-сообщениями для согласования параметров;
4. после всех согласований сессия переходит в состояние ESTABLISHED. Это говорит нам о том, что BGP сессия успешно установлена;
5. далее, происходит обмен UPDATE-сообщениями, в которых устройства обмениваются известными им маршрутами, вследствие чего происходит заполнение таблицы маршрутизации BGP. Важно отметить, что именно сами маршруты содержатся в поле NLRI сообщения UPDATE. До отправки UPDATE- сообщений в таблицу маршрутизации будут добавлены только локальные маршруты.

Убедимся, что соседства между устройствами были установлены и посмотрим маршруты:

Выводы команд "show ip bgp summary" и "show ip route"

R1

```
Neighbor    V      AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
1.1.1.2     4      10       4       2       14    0    0 00:00:25       2
C   1.1.1.0/30 is directly connected, Vlan10  tag:0
B   2.2.2.0/30 [200/0] via 1.1.1.2, Vlan10, 00:01:43  tag:0
```

R2

```
Neighbor    V      AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
1.1.1.1     4      10       2       3       34    0    0 00:00:06       0
2.2.2.2     4      20       7       9       34    0    0 00:05:26       0
C   1.1.1.0/30 is directly connected, Vlan10  tag:0
C   2.2.2.0/30 is directly connected, Vlan20  tag:0
```

R3

```
Neighbor    V      AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2.2.2.1     4      10       8       7        9    0    0 00:05:05       1
C   2.2.2.0/30 is directly connected, Vlan20  tag:0
B   1.1.1.0/30 [200/0] via 2.2.2.1, Vlan20, 00:01:13  tag:0
```

BFD (Bidirectional Forwarding Detection)

Стоит учитывать, что при падении линка или потере связности между устройствами, BGP-сессия исчезает не сразу. Она остается активной еще некоторое время. Для быстрой сходимости сети в данном случае используют протокол BFD (RFC 5880). Данный протокол обнаруживает проблемы связности на IP-уровне и обеспечивает быструю сходимость сети. Рассмотрим ниже пример настройки. Обращаем внимание на то, что для каждого соседа он включается отдельно, при этом настройка таймеров производится на VLAN-интерфейсе:

Конфигурация коммутаторов

R1

```
!
vlan 10
 name BGP_Test
!
interface Vlan10
 ip address 1.1.1.1 255.255.255.252
!
Interface Ethernet1/0/9
 switchport mode trunk
 switchport trunk allowed vlan 10
!
router bgp 10
 bgp router-id 1.1.1.1
 neighbor 1.1.1.2 remote-as 10
 neighbor 1.1.1.2 bfd
```

R2

```
!
vlan 10
 name BGP_Test
!
interface Vlan10
 ip address 1.1.1.2 255.255.255.252
!
Interface Ethernet1/0/9
 switchport mode trunk
 switchport trunk allowed vlan 10
!
router bgp 10
 bgp router-id 1.1.1.2
 neighbor 1.1.1.1 remote-as 10
 neighbor 1.1.1.1 bfd
```

Вывод команды “show ip bgp summary” до использования протокола BFD

```
SNR-S300X-24FQ#sh ip bgp summary
Neighbor    V      AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
1.1.1.2     4      10       2       3       33    0    0 00:00:28       0
SNR-S300X-24FQ#sh ip bgp summary
Neighbor    V      AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
1.1.1.2     4      10       2       7       37    0    0 00:04:00       0
```

Теперь проверим то же самое, но уже после применения BFD. До применения команды сессия была активна еще порядка 4-х минут, хотя линк между коммутаторами уже отсутствовал. Только спустя это время, в логах коммутатора появляется информация о том, что не удалось установить соседство:

Логи коммутатора до и после применения протокола BFD

До использования протокола:

```
11:52:56 2021 %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet1/0/9, changed state to DOWN
11:52:57 2021 %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10,changed state to DOWN
Dec 14 11:56:42 2021 BGP: Get peer failed, please check the config.
```

После применения команд:

```
11:49:03 2021 %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet1/0/9, changed state to DOWN
11:49:04 2021 %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10,changed state to DOWN
11:49:11 2021 BGP: Get peer failed, please check the config.
```

```
SNR-S300X-24FQ#sh ip bgp summary
Neighbor    V      AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
1.1.1.2     4      10       0       0        0    0    0    never Active
```

Примеры управления BGP-маршрутизацией

Для управления политикой маршрутизации в BGP, существует несколько способов. В данном примере мы остановимся на таких атрибутах BGP как Weight, Community, MED. Для примера всех атрибутов, будет собрана одинаковая схема:

[Изображение недоступно]

MED (multi-exit discriminator)

Атрибут MED (RFC 4451) предназначен для определения наилучшего пути, по которому другие AS могут взаимодействовать с нашей AS. Данный атрибут не локальный, он передается между AS. При сравнении с другими атрибутами он будет менее весомым. Если вам требуется чтобы всегда происходило сравнение значения метрики между разными AS, то необходимо добавить в конфигурацию команду “***bgp always-compare-med*”.**

Конфигурация устройств

R1

```
vlan 1;40;50
!
Interface Ethernet1/0/9
 switchport mode trunk
 switchport trunk allowed vlan 40
!
Interface Ethernet1/0/11
 switchport mode trunk
 switchport trunk allowed vlan 50
!
interface Vlan40
 ip address 4.4.4.2 255.255.255.252
!
interface Vlan50
 ip address 5.5.5.1 255.255.255.252
!
!
router bgp 10
 bgp router-id 4.4.4.2
 bgp always-compare-med
 neighbor 4.4.4.1 remote-as 10
 neighbor 4.4.4.1 route-map metric out
 neighbor 5.5.5.2 remote-as 30
!
route-map metric permit 10
 set metric 50
```

R2

```
vlan 1;10;20;40
!
Interface Ethernet1/0/5
 shutdown
 switchport mode trunk
 switchport trunk allowed vlan 40
!
Interface Ethernet1/0/6
!
Interface Ethernet1/0/7
 switchport mode trunk
 switchport trunk allowed vlan 10
!
Interface Ethernet1/2/1
 interface mode cr4
 switchport mode trunk
 switchport trunk allowed vlan 20
!
```

```
interface Vlan10
 ip address 1.1.1.1 255.255.255.252
!
interface Vlan20
 ip address 2.2.2.1 255.255.255.252
!
interface Vlan40
 ip address 4.4.4.1 255.255.255.252
!
router bgp 10
 bgp router-id 1.1.1.1
 bgp always-compare-med
 network 1.1.1.0/30
 network 2.2.2.0/30
 network 4.4.4.0/30
 neighbor 1.1.1.2 remote-as 30
 neighbor 2.2.2.2 remote-as 20
 neighbor 4.4.4.2 remote-as 10
```

R3

```
vlan 1;20;60
!
Interface Ethernet1/0/15
 switchport mode trunk
 switchport trunk allowed vlan 60
!
Interface Ethernet1/1/1
 interface mode cr4
 switchport mode trunk
 switchport trunk allowed vlan 20
!
interface Vlan20
 ip address 2.2.2.2 255.255.255.252
!
interface Vlan60
 ip address 6.6.6.2 255.255.255.252
!
router bgp 20
 bgp router-id 2.2.2.2
 bgp always-compare-med
 neighbor 2.2.2.1 remote-as 10
 neighbor 2.2.2.1 route-map metric out
 neighbor 6.6.6.1 remote-as 30
 neighbor 6.6.6.1 route-map metric out
!
route-map metric permit 10
 set metric 150
```

R4

```
vlan 1;10;50;60
!
Interface Ethernet1/0/5
 switchport mode trunk
 switchport trunk allowed vlan 50
!
Interface Ethernet1/0/7
 switchport mode trunk
 switchport trunk allowed vlan 10
!
Interface Ethernet1/0/21
 switchport mode trunk
 switchport trunk allowed vlan 60
!
interface Vlan10
 ip address 1.1.1.2 255.255.255.252
!
interface Vlan50
```

```
 ip address 5.5.5.2 255.255.255.252
!
interface Vlan60
 ip address 6.6.6.1 255.255.255.252
!
router bgp 30
 bgp router-id 1.1.1.2
 bgp always-compare-med
 network 5.5.5.0/30
 network 6.6.6.0/30
 neighbor 1.1.1.1 remote-as 10
 neighbor 5.5.5.1 remote-as 10
 neighbor 5.5.5.1 route-map metric out
 neighbor 6.6.6.2 remote-as 20
!
route-map metric permit 10
 set metric 100
```

Вывод таблицы маршрутизации до добавления метрики

show ip bgp cidr-only

R1

```
   Network      Next Hop        Metric LocPrf Weight Path
*>i1.1.1.0/30   4.4.4.1               100      0 i
*>i2.2.2.0/30   4.4.4.1               100      0 i
*>i4.4.4.0/30   4.4.4.1               100      0 i
* i5.5.5.0/30   1.1.1.2               100      0 30 i
*>              5.5.5.2                  0 30 i
* i6.6.6.0/30   1.1.1.2               100      0 30 i
*>              5.5.5.2                  0 30 i
```

R2

```
   Network      Next Hop        Metric LocPrf Weight Path
*> 1.1.1.0/30   0.0.0.0                  32768 i
*> 2.2.2.0/30   0.0.0.0                  32768 i
*> 4.4.4.0/30   0.0.0.0                  32768 i
* i5.5.5.0/30   5.5.5.2               100      0 30 i
*               2.2.2.2                  0 20 30 i
*>              1.1.1.2                  0 30 i
* i6.6.6.0/30   5.5.5.2               100      0 30 i
*               2.2.2.2                  0 20 30 i
*>              1.1.1.2                  0 30 i
```

R3

```
   Network      Next Hop        Metric LocPrf Weight Path
*  1.1.1.0/30   6.6.6.1                     0 30 10 i
*>              2.2.2.1                  0 10 i
*  2.2.2.0/30   6.6.6.1                     0 30 10 i
*>              2.2.2.1                  0 10 i
*  4.4.4.0/30   6.6.6.1                     0 30 10 i
*>              2.2.2.1                  0 10 i
*  5.5.5.0/30   2.2.2.1                     0 10 30 i
*>              6.6.6.1                  0 30 i
*  6.6.6.0/30   2.2.2.1                     0 10 30 i
*>              6.6.6.1                  0 30 i
```

R4

```
   Network      Next Hop        Metric LocPrf Weight Path
*  1.1.1.0/30   5.5.5.1                     0 10 i
*               6.6.6.2                  0 20 10 i
*>              1.1.1.1                  0 10 i
*  2.2.2.0/30   5.5.5.1                     0 10 i
*               6.6.6.2                  0 20 10 i
*>              1.1.1.1                  0 10 i
*  4.4.4.0/30   5.5.5.1                     0 10 i
```

```
*            6.6.6.2                  0 20 10 i
*>            1.1.1.1                  0 10 i
*> 5.5.5.0/30   0.0.0.0                  32768 i
*> 6.6.6.0/30   0.0.0.0                  32768 i
```

Атрибут MED по умолчанию равен нулю. Изменим данное значение, создав route-map и добавив его к необходимым соседям. Сущность route-map - это набор правил, по которым маршрутизируется трафик. После применения данной политики правил, изменится Next Hop для подсетей на каждом устройстве. К примеру, ранее на R4 для подсети 1.1.1.0/30 Next Hop был 5.5.5.1, но после задания метрики он изменился на 6.6.6.2. Аналогично изменился Next Hop на других устройствах.

Посмотрим выводы таблицы маршрутизации:

R1

```
   Network      Next Hop       Metric LocPrf Weight Path
*>i1.1.1.0/30   4.4.4.1               100      0 i
*>i2.2.2.0/30   4.4.4.1               100      0 i
*>i4.4.4.0/30   4.4.4.1               100      0 i
*>i5.5.5.0/30   1.1.1.2               100      0 30 i
*            5.5.5.2             100      0 30 i
*>i6.6.6.0/30   1.1.1.2               100      0 30 i
*            5.5.5.2             100      0 30 i
```

R2

```
   Network      Next Hop       Metric LocPrf Weight Path
*> 1.1.1.0/30   0.0.0.0                  32768 i
*> 2.2.2.0/30   0.0.0.0                  32768 i
*> 4.4.4.0/30   0.0.0.0                  32768 i
*> 5.5.5.0/30   1.1.1.2                    0 30 i
*> 6.6.6.0/30   1.1.1.2                    0 30 i
```

R3

```
   Network      Next Hop       Metric LocPrf Weight Path
*> 1.1.1.0/30   2.2.2.1                    0 10 i
*            6.6.6.1                  0 30 10 i
*> 2.2.2.0/30   2.2.2.1                    0 10 i
*            6.6.6.1                  0 30 10 i
*> 4.4.4.0/30   2.2.2.1                    0 10 i
*            6.6.6.1                  0 30 10 i
*  5.5.5.0/30   2.2.2.1                   0 10 30 i
*>            6.6.6.1                  0 30 i
*  6.6.6.0/30   2.2.2.1                   0 10 30 i
*>            6.6.6.1                  0 30 i
```

R4

```
   Network      Next Hop       Metric LocPrf Weight Path
*  1.1.1.0/30   6.6.6.2         150          0 20 10 i
*            1.1.1.1                  0 10 i
*>            5.5.5.1                  0 10 i
*  2.2.2.0/30   6.6.6.2         150          0 20 10 i
*            1.1.1.1                  0 10 i
*>            5.5.5.1                  0 10 i
*  4.4.4.0/30   6.6.6.2         150          0 20 10 i
*            1.1.1.1                  0 10 i
*>            5.5.5.1                  0 10 i
*> 5.5.5.0/30   0.0.0.0                  32768 i
*> 6.6.6.0/30   0.0.0.0                  32768 i
```

BGP community

Атрибут BGP community (RFC 4360) предназначен для маркировки маршрутов с целью последующей их обработки по специальным правилам. У маршрутов может быть несколько community. К примеру, на основе данного атрибута можно фильтровать трафик. Также, данный атрибут применяется для балансировки и распределения нагрузки. Принцип его работы почти аналогичен работе ip access-list:

Конфигурация Community на R2

```
access-list 1 permit 2.0.0.0 0.255.255.255
access-list 2 permit any-source
!
router bgp 10
 neighbor 2.2.2.2 route-map set-community out
!
route-map set-community permit 10
 match ip address 1
 set community 1111
!
route-map set-community permit 20
 match ip address 2
```

Вывод команд

```
R3#sh ip bgp 2.2.2.0
BGP routing table entry for 2.2.2.0/30
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  10
2.2.2.1 from 2.2.2.1 (1.1.1.1)
    Origin IGP, localpref 100, valid, external, best
    Community: 0:1111
    Last update: 00:02:39
R3#sh ip bgp 4.4.4.0
BGP routing table entry for 4.4.4.0/30
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  10
2.2.2.1 from 2.2.2.1 (1.1.1.1)
    Origin IGP, localpref 100, valid, external, best
    Last update: 00:05:56
```

Как можно заметить из выводов команд выше, после настройки на R3 приходит атрибут Community 0:1111, а на остальные адреса данный атрибут не распространяется, т.к. они не подходят под созданный нами ACL. Маршрут может иметь и другие атрибуты Community, здесь разобран лишь пример самой простой настройки.

Weight

Атрибут Weight применяется в тех случаях, когда устройство имеет более одного выхода в другие AS. Логика работы данного атрибута очень проста - чем больше его численное значение, тем выше его приоритет. Важно отметить, что данный атрибут локальный, он не передается другим BGP-соседям.

Конфигурация R3

```
!
router bgp 20
 bgp router-id 2.2.2.2
 bgp bestpath as-path ignore
 neighbor 2.2.2.1 remote-as 10
 neighbor 2.2.2.1 weight 150
 neighbor 6.6.6.1 remote-as 30
 neighbor 6.6.6.1 weight 500
!
```

Атрибут можно указать непосредственно на соседе, а также это можно сделать с помощью route-map. Когда префикс генерируется локально, то по умолчанию его значение будет равно 32768.

BGP Confederation

Для масштабирования сети можно использовать функционал BGP Confederation (RFC 3065). Суть заключается в том, чтобы разделить большую автономную систему AS на несколько подсистем (подуровней AS). Это позволяет всем iBGP-соседям изучить все iBGP-маршруты в AS, не используя при этом полносвязную топологию между всеми L3-коммутаторами в сети. Для других AS конфедерация будет отображаться только под своим глобальным номером AS. В основном, BGP confederation используется на крупных сетях, где необходимо использовать крупную AS для захвата малых, более

управляемых sub-AS. Важно отметить, что при настройке “bgp confederation peers” внутри конфедерации, нужно указывать номер sub-AS соседа. Рассмотрим пример настройки:

[Изображение недоступно]

Конфигурация коммутаторов

R1

```
!
vlan 10
 name BGP_Test
!
interface Vlan10
 ip address 1.1.1.2 255.255.255.252
!
Interface Ethernet1/0/9
 switchport mode trunk
 switchport trunk allowed vlan 10
!
router bgp 10
 bgp confederation identifier 100
 bgp confederation peers 30
 neighbor 1.1.1.1 remote-as 30
```

R2

```
!
vlan 10
 name BGP_Test
!
interface Vlan10
 ip address 1.1.1.1 255.255.255.252
!
vlan 20
 name EBGP_Test
!
interface Vlan20
 ip address 2.2.2.1 255.255.255.252
!
Interface Ethernet1/0/9
 switchport mode trunk
 switchport trunk allowed vlan 10
!
Interface Ethernet1/0/11
 switchport mode trunk
 switchport trunk allowed vlan 20
!
router bgp 30
 bgp router-id 1.1.1.1
 bgp confederation identifier 100
 bgp confederation peers 10
 neighbor 1.1.1.2 remote-as 10
 neighbor 2.2.2.2 remote-as 20
```

R3

```
!
vlan 20
 name EBGP_Test
!
interface Vlan20
 ip address 2.2.2.2 255.255.255.252
!
Interface Ethernet1/0/11
 switchport mode trunk
 switchport trunk allowed vlan 20
!
router bgp 20
 bgp router-id 2.2.2.2
```

```
 neighbor 2.2.2.1 remote-as 100
```

Как можно заметить из вывода команд ниже, для R3 не будут видны номера sub-AS устройств в BGP confederation. Ему будет известна только непосредственно AS конфедерации, которая в нашем случае имеет идентификатор 100. R1, соответственно, видит в таблице маршрутизации только номер sub-AS, но если к нему присоединить еще одну BGP конфедерацию, то он также будет видеть только AS самой конфедерации.

Просмотр маршрутов и проверка связности

R1

```
sh ip bgp
   Network      Next Hop       Metric LocPrf Weight Path
*> 1.1.1.0/30   1.1.1.1              100      0 (30) i
*> 2.2.2.0/30   1.1.1.1              100      0 (30) i
```

R3

```
sh ip bgp
   Network      Next Hop       Metric LocPrf Weight Path
*> 1.1.1.0/30   2.2.2.1                   0 100 i
*> 2.2.2.0/30   2.2.2.1                   0 100 i
SNR-S4650X-48FQ#ping 1.1.1.1
Type ^c to abort.
Sending 5 56-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds.
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/6/16 ms
SNR-S4650X-48FQ#ping 1.1.1.2
Type ^c to abort.
Sending 5 56-byte ICMP Echos to 1.1.1.2, timeout is 2 seconds.
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/0 ms
```

Route Reflector

Еще одним способом масштабирования сети будет использование коммутатора как Route Reflector (RFC2796)**.** К примеру, если в AS будет использоваться большое количество устройств, то выглядит нелогично и неудобно прописывать множество соседей на каждом устройстве. Route Reflector будет выступать в роли "зеркала" и каждый коммутатор будет связан с ним только одним линком. При такой схеме мы получаем, что Route Reflector отправляет своим клиентам маршруты, а клиент не пересылает их дальше, т.к. является конечной точкой, ведь у него нет ни с кем более линков. RR-устройств может быть несколько, но в нашей схеме для примера мы будем использовать всего один. Логика работы этого устройства будет следующей:

- Если маршрут был получен от своего клиента, то маршрут будет отправлен всем.
- Если RR получил маршрут не от клиента, то такой маршрут будет отправлен только своим клиентам.
- Если RR получил маршрут от eBGP-соседа, то он пересылает этот маршрут всем.

[Изображение недоступно]

Конфигурация устройств

RR

```
vlan 10,20,333,50
!
Interface Ethernet1/0/5
 switchport mode trunk
 switchport trunk allowed vlan 50
!
Interface Ethernet1/0/6
!
Interface Ethernet1/0/7
 switchport mode trunk
 switchport trunk allowed vlan 10
!
Interface Ethernet1/0/8
 switchport mode trunk
```

```
 switchport trunk allowed vlan 333
!
Interface Ethernet1/2/1
 interface mode cr4
 switchport mode trunk
 switchport trunk allowed vlan 20
!
interface Vlan10
 ip address 1.1.1.1 255.255.255.252
!
interface Vlan20
 ip address 2.2.2.1 255.255.255.252
!
interface Vlan50
 ip address 5.5.5.1 255.255.255.252
!
interface Vlan333
 ip address 3.3.3.1 255.255.255.252
!
router bgp 10
 bgp router-id 1.1.1.1
 network 1.1.1.0/30
 network 2.2.2.0/30
 network 3.3.3.0/30
 network 5.5.5.0/30
 redistribute connected
 neighbor 1.1.1.2 remote-as 10
 neighbor 1.1.1.2 route-reflector-client
 neighbor 2.2.2.2 remote-as 20
 neighbor 3.3.3.2 remote-as 10
 neighbor 3.3.3.2 route-reflector-client
 neighbor 5.5.5.2 remote-as 10
 neighbor 5.5.5.2 route-reflector-client
!
```

SNR S4650X-48FQ

```
vlan 20
!
vlan 20
 name EBGP
!
Interface Ethernet1/1/1
 interface mode cr4
 switchport mode trunk
 switchport trunk allowed vlan 20
!
interface Vlan20
 ip address 2.2.2.2 255.255.255.252
!
router bgp 20
 bgp router-id 2.2.2.2
 neighbor 2.2.2.1 remote-as 10
```

R1

```
vlan 10
!
vlan 10
 name iBGP
!
Interface Ethernet1/0/7
 switchport mode trunk
 switchport trunk allowed vlan 10
!
interface Vlan10
 ip address 1.1.1.2 255.255.255.252
!
router bgp 10
```

```
 neighbor 1.1.1.1 remote-as 10
```

R2

```
vlan 333
!
vlan 333
 name RR_TEST
!
Interface Ethernet1/0/9
 switchport mode trunk
 switchport trunk allowed vlan 333
!
interface Vlan333
 ip address 3.3.3.2 255.255.255.252
!
router bgp 10
 bgp router-id 3.3.3.2
 neighbor 3.3.3.1 remote-as 10
```

R3

```
vlan 50
!
vlan 50
 name RR_TEST
!
Interface Ethernet1/0/12
 switchport mode trunk
 switchport trunk allowed vlan 50
!
interface Vlan50
 ip address 5.5.5.2 255.255.255.252
!
router bgp 10

 bgp router-id 5.5.5.2
 neighbor 5.5.5.1 remote-as 10
```

В качестве RR был выбран коммутатор SNR-S300X-24FQ, а клиентами были выбраны три других коммутатора SNR-S300X-24FQ и SNR-S300G-24FX . Как можно заметить, нам не требуется соединять каждое устройство с каждым, т.е использовать Full-Mesh-топологию. Все клиенты подключаются к центральной точке RR и получают всю информацию о маршрутах через него. Это сильно упрощает конфигурацию, особенно на сети оператора, где в схеме участвует гораздо больше устройств. Убедимся, что каждое устройство доступно друг для друга и посмотрим их таблицу маршрутизации:

Выводы команд

RR

```
sh ip bgp summary
Neighbor    V      AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
1.1.1.2     4      10      15      18      173    0    0 00:11:45      0
2.2.2.2     4      20     175     186      173    0    0 02:28:03      0
3.3.3.2     4      10     176     187      173    0    0 02:28:03      0
5.5.5.2     4      10       4       5      173    0    0 00:01:55      0
sh ip route
C   1.1.1.0/30 is directly connected, Vlan10  tag:0
C   2.2.2.0/30 is directly connected, Vlan20  tag:0
C   3.3.3.0/30 is directly connected, Vlan333  tag:0
C   5.5.5.0/30 is directly connected, Vlan50  tag:0
```

R1

```
sh ip bgp summary
Neighbor    V      AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
1.1.1.1     4      10      51      49     1663    0    0 00:40:41      5
sh ip route
C   1.1.1.0/30 is directly connected, Vlan10  tag:0
B   2.2.2.0/30 [200/0] via 1.1.1.1, Vlan10, 00:40:53  tag:0
```

```
B   3.3.3.0/30 [200/0] via 1.1.1.1, Vlan10, 00:40:53  tag:0
B   5.5.5.0/30 [200/0] via 1.1.1.1, Vlan10, 00:32:40  tag:0
sh ip route 5.5.5.2
Routing entry for 5.5.5.0/30
  Known via "bgp", distance 200, metric 0, best
  Last update 00:35:45 ago
  * 3.3.3.1, via Vlan333
sh ip bgp 5.5.5.2
BGP routing table entry for 5.5.5.0/30
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Local
3.3.3.1 from 3.3.3.1 (1.1.1.1)
   Origin incomplete, localpref 100, valid, internal, best
   Last update: 00:37:09
R1#ping 1.1.1.2
Type ^c to abort.
Sending 5 56-byte ICMP Echos to 1.1.1.2, timeout is 2 seconds.
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/3/16 ms
R1#ping 2.2.2.2
Type ^c to abort.
Sending 5 56-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds.
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/0 ms
R1#ping 3.3.3.2
Type ^c to abort.
Sending 5 56-byte ICMP Echos to 3.3.3.2, timeout is 2 seconds.
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/0 ms
R1#ping 5.5.5.2
Type ^c to abort.
Sending 5 56-byte ICMP Echos to 5.5.5.2, timeout is 2 seconds.
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/0 ms
```

На R2 и R3 таблицы маршрутизации выглядят аналогично, за исключением того, что будет другая подсеть directly connected.

BGP VPN

Часто для изоляции трафика используется VPN, а для их обслуживания создаются VRF. В данном разделе мы рассмотрим пример настройки MPLS VPN (RFC 7432). С принципом работы MPLS можно более подробно ознакомиться в нашем вебинаре. На схеме, разобранной нами ниже, будет рассмотрен только пример общей конфигурации MPLS-домена.

MPLS L3VPN-инфраструктура предполагает обеспечение изоляции клиентских IP-сетей в рамках VPN. VRF определяет принадлежность в VPN подсети за узлом клиентского оборудования, подключенного к граничному маршрутизатору. Интерфейсы граничных маршрутизаторов, обращенные к клиентскому оборудованию, логически связаны с персональными VRF. Для обмена маршрутной информацией между VRF разных граничных маршрутизаторов применяется протокол BGP, а сам BGP в свою очередь оперирует VPN-IPv4 маршрутами. Важно отметить, что VRF - локален для каждого маршрутизатора. Для определения VRF, в который требуется передать маршрут, используется уникальный идентификатор RT. Данный атрибут анонсируется в обновлениях BGP, как атрибут BGP extended community. А чтобы отличать принадлежность маршрута к определенному VPN, к префиксу подсети также добавляется идентификатор RD, который также передается в BGP extended community.

[Изображение недоступно]

Конфигурация R1 и R2

```
mpls enable
!
vlan 100
 name TEST_VRF
!
Interface Ethernet1/0/5
 switchport mode trunk
```

```
 switchport trunk allowed vlan 100
!
ip vrf VRF-A
 rd 100:10
 route-target both 100:10
!
ip vrf VRF-B
 rd 100:20
 route-target both 100:20
!
interface Vlan100
 mtu 2000
 label-switching
 ldp enable
 ip address 10.10.10.1 255.255.255.252
!
interface Loopback100
 ip address 1.1.1.1 255.255.255.255
!
interface Loopback101
 ip vrf forwarding VRF-A
 ip address 192.168.100.1 255.255.255.255
!
interface Loopback102
 ip vrf forwarding VRF-B
 ip address 192.168.120.1 255.255.255.255
!
router bgp 10
 redistribute connected
 redistribute static
 neighbor 2.2.2.2 remote-as 10
 neighbor 2.2.2.2 update-source 1.1.1.1
 address-family vpnv4 unicast
 neighbor 2.2.2.2 activate
 exit-address-family
 address-family ipv4 vrf VRF-A
 redistribute connected
 redistribute static
 exit-address-family
 address-family ipv4 vrf VRF-B
 redistribute connected
 redistribute static
 exit-address-family
!
router ldp
 router-id 1.1.1.1
 transport-address 1.1.1.1
```

R2

```
mpls enable
!
vlan 100
 name TEST_VRF
!
Interface Ethernet1/0/13
 switchport mode trunk
 switchport trunk allowed vlan 100
!
ip vrf VRF-A
 rd 100:10
 route-target both 100:10
!
ip vrf VRF-B
 rd 100:20
 route-target both 100:20
!
interface Vlan100
 mtu 2000
```

```
 label-switching
 ldp enable
 ip address 10.10.10.2 255.255.255.252
!
!
interface Loopback100
 ip address 2.2.2.2 255.255.255.255
!
interface Loopback101
 ip vrf forwarding VRF-A
 ip address 192.168.200.1 255.255.255.255
!
interface Loopback102
 ip vrf forwarding VRF-B
 ip address 192.168.220.1 255.255.255.255
!
router bgp 10
 redistribute connected
 redistribute static
 neighbor 1.1.1.1 remote-as 10
 neighbor 1.1.1.1 update-source 2.2.2.2
 address-family vpnv4 unicast
 neighbor 1.1.1.1 activate
 exit-address-family
 address-family ipv4 vrf VRF-A
 redistribute connected
 redistribute static
 exit-address-family
 address-family ipv4 vrf VRF-B
 redistribute connected
 redistribute static
 exit-address-family
!
router ldp
 router-id 2.2.2.2
 transport-address 2.2.2.2
```

Рассмотрим содержание таблицы маршрутизации на коммутаторах:

Вывод команды "show ip route"

R1

```
C   1.1.1.1/32 is directly connected, Loopback100  tag:0
O E2 2.2.2.2/32 [110/20] via 10.10.10.2, Vlan100, 01:35:58  tag:0
C   10.10.10.0/30 is directly connected, Vlan100  tag:0
```

R2

```
O E2 1.1.1.1/32 [110/20] via 10.10.10.1, Vlan100, 01:36:15  tag:0
C   2.2.2.2/32 is directly connected, Loopback100  tag:0
C   10.10.10.0/30 is directly connected, Vlan100  tag:0
```

Вывод команды "show ip route vrf VRF-A"

R1

```
C   192.168.100.1/32 is directly connected, Loopback101  tag:0
B   192.168.200.1/32 [200/0] via 2.2.2.2, 01:38:42  tag:0
```

R2

```
B   192.168.100.1/32 [200/0] via 1.1.1.1, 01:40:07  tag:0
C   192.168.200.1/32 is directly connected, Loopback101  tag:0
```

Вывод команды "show ip route vrf VRF-B"

R1

```
C   192.168.120.1/32 is directly connected, Loopback102  tag:0
B   192.168.220.1/32 [200/0] via 2.2.2.2, 01:38:47  tag:0
```

R2

```
B   192.168.120.1/32 [200/0] via 1.1.1.1, 01:40:23  tag:0
C   192.168.220.1/32 is directly connected, Loopback102  tag:0
```

Как можно заметить из вывода команд, мы изолировали данные маршруты между собой. Они будут недоступны для друг друга, т.к. находятся в разных VRF.

Подводя итог, BGP - основной связующий протокол в сети Internet. Де-факто он является стандартом для маршрутизации в сети Internet и, соответственно, необходим для большинства поставщиков Интернет-услуг. Кроме того, данный протокол может использоваться не только крупными операторами связи, но и большими частными IP-сетями. Есть множество способов и политик управления маршрутизацией, также изменяющих принцип работы протокола BGP. В данной статье мы затронули только основные принципы работы протокола BGP, его базовую настройку и примеры использования на коммутаторах SNR.