eucariot 15 октября 2013 в 12:00

Сети для Самых Маленьких. Микровыпуск №3. IBGP

Я пиарюсь

Tutorial

Все выпуски

Долго ли коротко ли длилась история linkmeup, но компания росла, развивалась. Счёт маршрутизаторов уже на десятки, свои опто-волоконные линии, развитая сеть по городу. И было принято решение оформлять компанию, как провайдера и предоставлять услуги доступа в Интернет для сторонних в том числе организаций.
Сама по себе задача административная — лицензии там, поиск клиентской базы, реклама, поставить СОРМ.
Разумеется, с технической стороны тоже нужны приготовления — просчитать ресурсы, мощности, порты, подготовить политику QoS. Но всё это (за исключением QoS) — рутина.

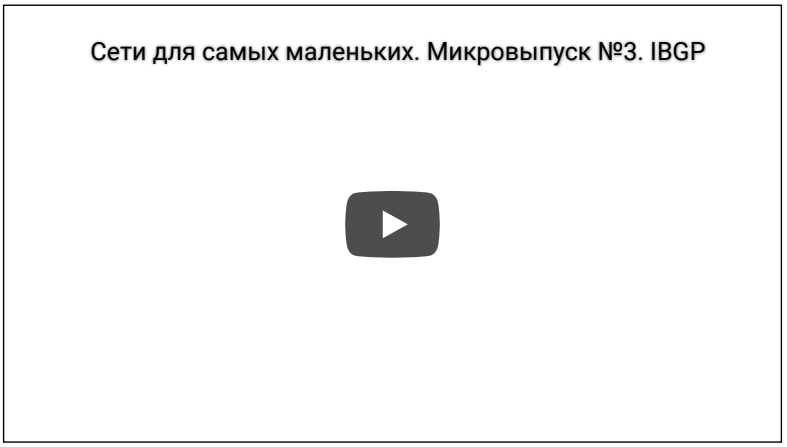
Мы же хотим поговорить о другом — IBGP. Возможно, тема покажется вам несколько притянутой за уши, мол, внутренний BGP — прерогатива достаточно крупных провайдеров.
Однако это не так, сейчас iBGP задействуется в ентерпрайзах чуть ли не чаще, чем в провайдерах. С целью исключительно внутренней маршрутизации. Например, ради VPN — очень популярное приложение на базе BGP в корпоративной среде. К примеру, возможность организовать периметры, изолированные на L3, на уже используемой инфраструктуре очень ценна. А префиксов-то может быть каких-то полсотни, а то и десяток. Вовсе никакой не Full View, однако все равно удобно.

Возможно, к нашей сети Linkmeup это не имеет по-прежнему отношения, но обойти стороной такую концепцию будет совершенно непростительно. Поэтому предположим, что сеть достаточно велика, и у нас есть необходимость в BGP в ядре.

Сегодня обсудим

- Когда нужен IBGP
- В чём отличия от EBGP
- Route Reflector’ы
- Конфедерации
- Нерассмотренные в основной статье атрибуты BGP

Традиционное видео

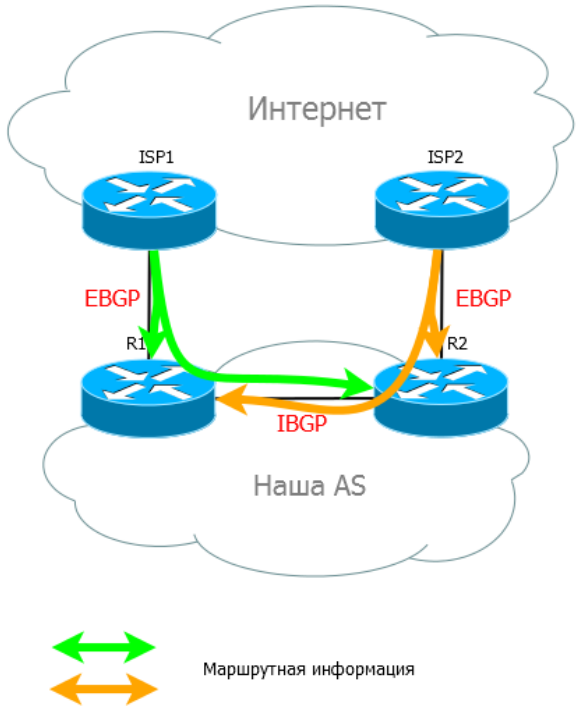


Задачки в этом выпуске не относятся напрямую к IBGP, это, скорее, по BGP в целом. Интересно будет как новичкам поломать голову, так и старожилам размяться

Что такое IBGP?

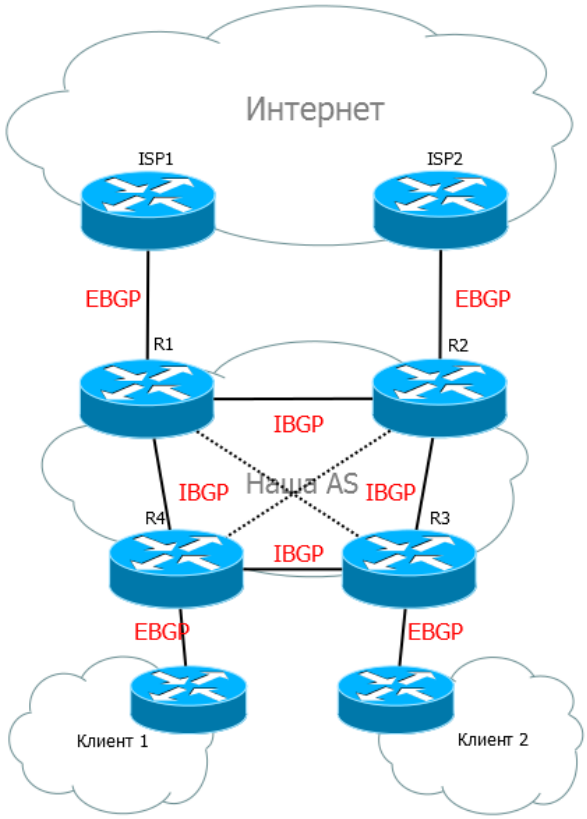
Начнём с того, что вообще такое Internal BGP. По сути это тот же самый BGP, но **внутри** AS. Он даже настраивается

практически так же.
Основных применения два:
Резервирование. Когда есть несколько линков к провайдерам и не хочется замыкать всё на одном своём граничном маршрутизаторе (т.н. боредре (от старославянского border — граница), ставится несколько маршрутизаторов, а между ними поднимается IBGP для того, чтобы на них всегда была актуальная информация обо всех маршрутах.



В случае проблем у провайдера ISP2 R2 будет знать о том, что те же самые сети доступны через ISP1. Об этом ему сообщит R1 по IBGP.

Подключение клиентов по BGP. Если стоит задача подключить клиента по BGP, при этом у вас больше, чем один маршрутизатор, без IBGP не обойтись.



Чтобы R4 передал Клиенту1 Full View, он должен получить его по IBGP от R1 или R2.

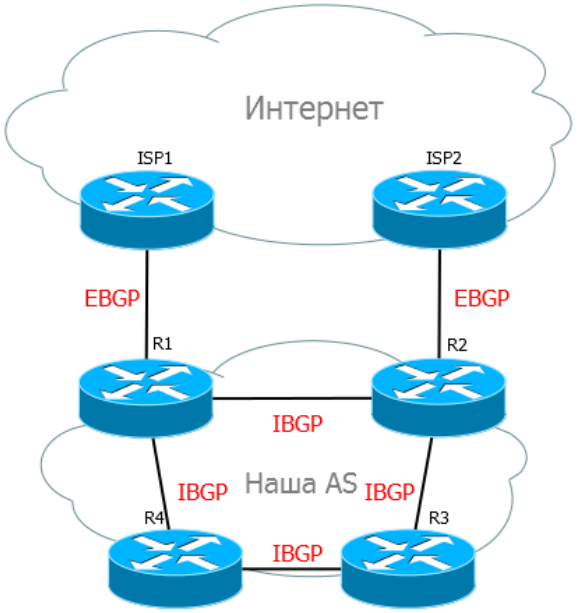
Повторимся, EBGP используется **между** Автомномными Системами, IBGP — **внутри**.

Различия IBGP и EBGP

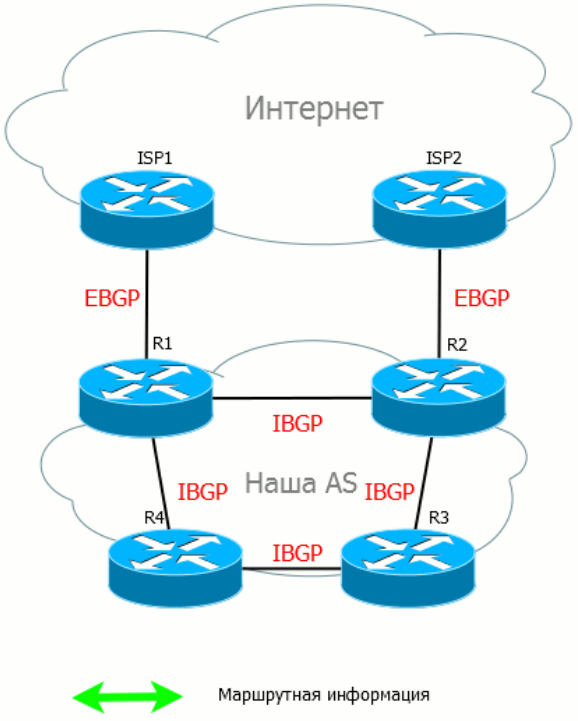
1) Главная тонкость, которая появляется при переходе внутрь Автономной Системы и откуда растут ноги почти всех отличий — петли. В EBGP мы с ними справлялись с помощью AS-Path. Если в списке уже был номер собственной AS, то такой маршрут отбрасывался.
Но, как вы помните, при передаче маршрута внутри Автономной Системы AS-Path не меняется. Вместо этого в IBGP прибегают к хитрости: используется **полносвязная топология** — все соседи имеют сессии со всеми — **Full Mesh**.
При этом маршрут, полученный от IBGP-соседа не анонсируется другим IBGP-соседям.
Это позволяет на всех маршрутизаторах иметь все маршруты и при этом не допустить петель.

Поясним на примерах.

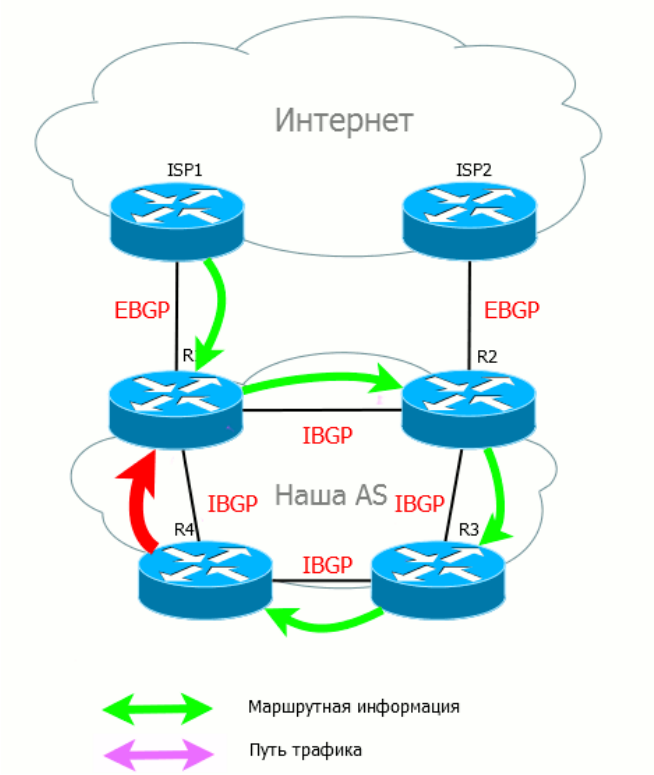
Как это могло бы быть в такой топологии, например, если не использовать технологию избежания петель:



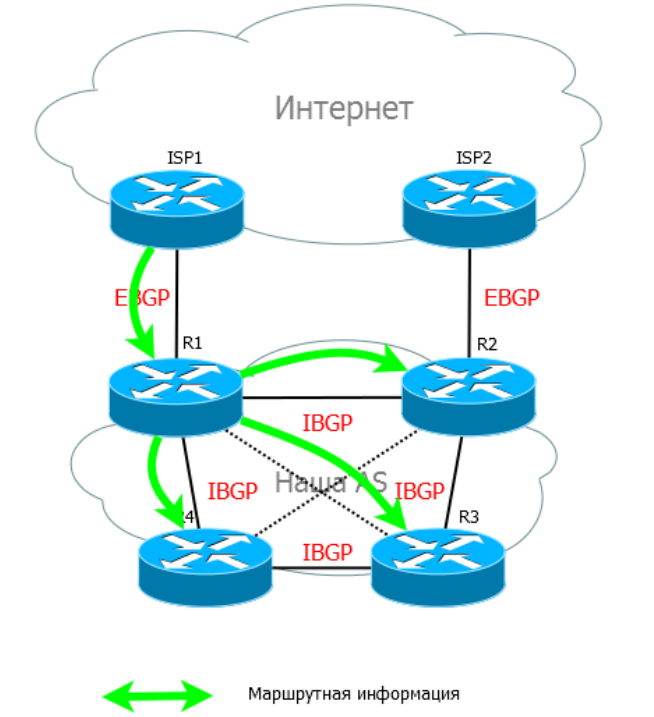
R1 получил анонс от EBGP-соседа, передал его R2, тот передал R3, R3 передал R4. Вроде, все молодцы, все знают, где находится сеть Интернет. Но R4 передаёт этот анонс обратно R1.



R1 получил маршрут от R4, и он по выгодности точно такой-же, как оригинальный от ISP — AS-Path-то не менялся. Поэтому в качестве приоритетного может выбраться даже новый маршрут от R4, что, естественно, неразумно: мало того, что маршруты будут изучены неверно, так и трафик в итоге может залоориться и не попадёт к точке назначения.



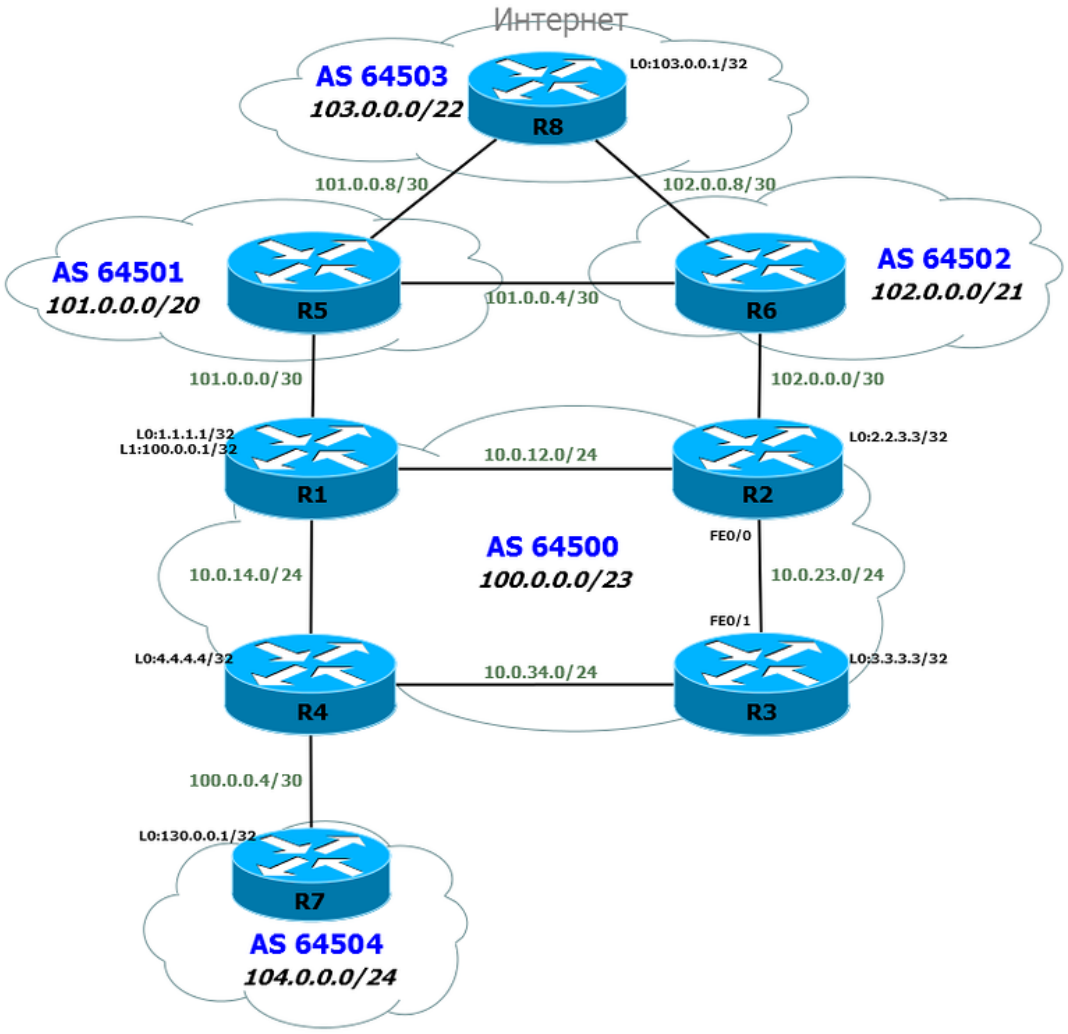
В случае же полносвязной топологии и правила Split Horizon, такая ситуация исключается. R1, получив анонс от ISP1, передаёт его сразу всем своим соседям: R2, R3, R4. А те в свою очередь эти анонсы сохраняют, но передают только EBGP-партнёрам, но не IBGP, именно потому, что получены от IBGP-партнёра. То есть все BGP-маршрутизаторы имеют актуальную информацию и исключены петли.



Причём, не имеет значение, подключены соседи напрямую или через промежуточные маршрутизаторы. Так, например, на вышеприведённой схеме R1 не имеет связи с R3 напрямую — они общаются через R2, однако это не мешает им установить TCP-сессию и поверх неё BGP.

> Понятие Split Horizon тут применяется в более широком смысле. Если в RIP это означало «не отсылать анонсы обратно в тот **интерфейс, откуда они пришли**», в IBGP это означает «не отсылать анонсы от IBGP-партнёров **другим IBGP-партнёрам**.»

2) Вторая тонкость — адрес Next Hop. В случае External BGP маршрутизатор при отправке анонса своему EBGP-соседу сначала меняет адрес Next-Hop на свой, а потом уже отсылает. Вполне логичное действие.



Вот как анонс сети 103.0.0.0/22 выглядит при передаче от R5 к R1:

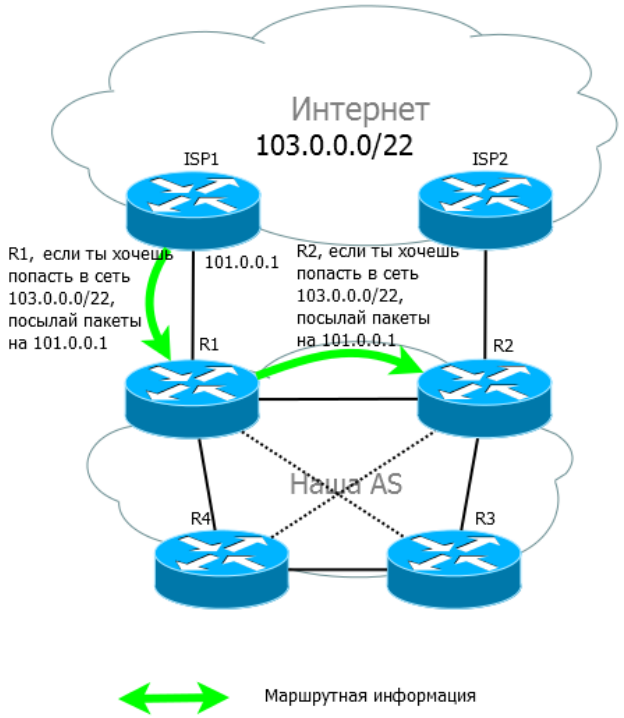
```
  275 18:52:11.903987    c0:04:24:64:00:10  c0:04:24:64:00:10 LOOP      60 Reply
  276 18:52:13.820987    101.0.0.2          101.0.0.1         BGP       77 ROUTE-REFRESH Message
  277 18:52:14.057987    101.0.0.1          101.0.0.2         BGP       247 UPDATE Message, UPDATE Message, UPDATE Message, UPDATE Message
  278 18:52:14.058987    101.0.0.2          101.0.0.1         BGP       153 UPDATE Message, UPDATE Message

⊞ Frame 277: 247 bytes on wire (1976 bits), 247 bytes captured (1976 bits)
⊞ Ethernet II, Src: c0:04:24:64:00:10 (c0:04:24:64:00:10), Dst: c0:01:27:c4:00:10 (c0:01:27:c4:00:10)
⊞ Internet Protocol Version 4, Src: 101.0.0.1 (101.0.0.1), Dst: 101.0.0.2 (101.0.0.2)
⊞ Transmission Control Protocol, Src Port: 38567 (38567), Dst Port: bgp (179), Seq: 672, Ack: 553, Len: 193
⊞ Border Gateway Protocol - UPDATE Message
⊞ Border Gateway Protocol - UPDATE Message
⊟ Border Gateway Protocol - UPDATE Message
    Marker: ffffffffffffffffffffffffffffffff
    Length: 47
    Type: UPDATE Message (2)
    Unfeasible routes length: 0 bytes
    Total path attribute length: 20 bytes
  ⊟ Path attributes
    ⊞ ORIGIN: IGP (4 bytes)
    ⊞ AS_PATH: 64501 64503 (9 bytes)
    ⊞ NEXT_HOP: 101.0.0.1 (7 bytes)
  ⊟ Network layer reachability information: 4 bytes
    ⊞ 103.0.0.0/22
⊞ Border Gateway Protocol - UPDATE Message
```

Если же маршрутизатор передаёт анонс IBGP-соседу, то адрес Next-Hop не меняется. Хм. Непонятно. Почему? Это расходится с привычным пониманием DV-протокола маршрутизации.

Вот тот же анонс при передаче от R1 r R2:

```
454 18:52:13.819987    1.1.1.1           2.2.2.2           BGP      77 ROUTE-REFRESH Message
455 18:52:13.910987    1.1.1.1           2.2.2.2           BGP      229 UPDATE Message, UPDATE Message, UPDATE Message

⊞ Frame 455: 229 bytes on wire (1832 bits), 229 bytes captured (1832 bits)
⊞ Ethernet II, Src: c0:01:27:c4:00:00 (c0:01:27:c4:00:00), Dst: c0:02:27:c4:00:01 (c0:02:27:c4:00:01)
⊞ Internet Protocol Version 4, Src: 1.1.1.1 (1.1.1.1), Dst: 2.2.2.2 (2.2.2.2)
⊞ Transmission Control Protocol, Src Port: 24420 (24420), Dst Port: bgp (179), Seq: 880, Ack: 636, Len: 175
⊞ Border Gateway Protocol - UPDATE Message
⊟ Border Gateway Protocol - UPDATE Message
    Marker: ffffffffffffffffffffffffffffffff
    Length: 61
    Type: UPDATE Message (2)
    Unfeasible routes length: 0 bytes
    Total path attribute length: 34 bytes
  ⊟ Path attributes
    ⊞ ORIGIN: IGP (4 bytes)
    ⊞ AS_PATH: 64501 64503 (9 bytes)
    ⊞ NEXT_HOP: 101.0.0.1 (7 bytes)
    ⊞ MULTI_EXIT_DISC: 0 (7 bytes)
    ⊞ LOCAL_PREF: 100 (7 bytes)
  ⊟ Network layer reachability information: 4 bytes
    ⊞ 103.0.0.0/22
⊞ Border Gateway Protocol - UPDATE Message
```

Дело в том, что здесь понятие Next-Hop отличается от того, которое используется в IGP. В IBGP он сообщает о точке выхода из локальной AS.



И тут возникает ещё один момент — важно, чтобы у получателя такого анонса был маршрут до Next-Hop — это первое, что проверяется при выборе лучшего маршрута. Если его не будет, то маршрут будет помещён в таблицу BGP, но его не будет в таблице маршрутизации.

Такой процесс называется рекурсивной маршрутизацией.

То есть, чтобы R2 мог отправлять пакеты ISP1 он должен знать, как добраться до адреса 101.0.0.1, который в данной схеме и является Next-Hop'ом для сети 103.0.0.0/22.

В принципе, практически всё оборудование даёт возможность менять адрес Next-Hop на свой при передаче маршрута IBGP-соседу.
На циске это делается командой "**neighbor XYZ Next-Hop-self**". Позже вы увидите, как это применяется на деле.

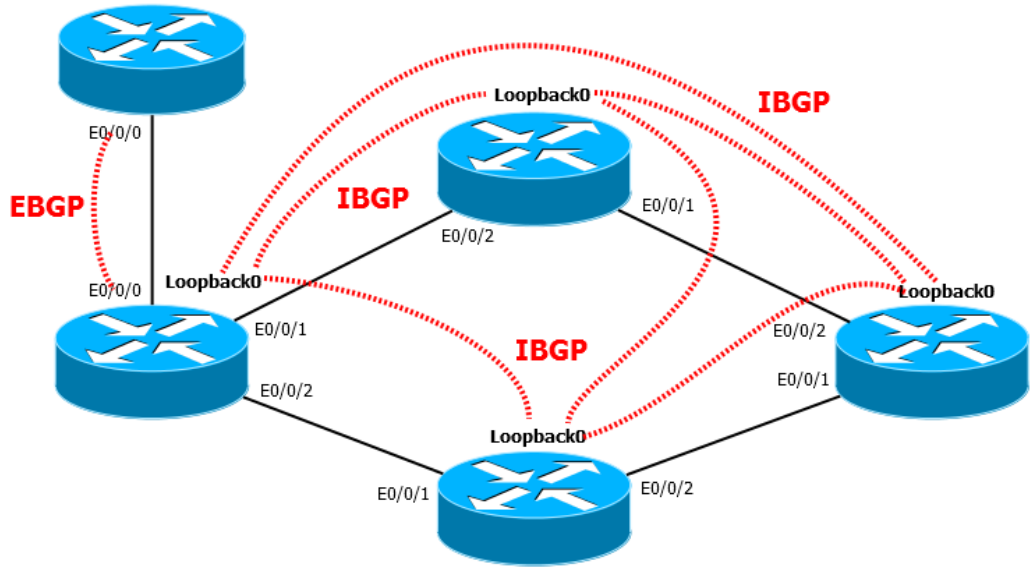
3) Третий момент: если в EBGP обычно подразумевается прямое подключение двух соседей друг к другу, то в Internal BGP соседи могут быть подключены через несколько промежуточных устройств.

> На самом деле в EBGP также можно настраивать соседей, которые находятся за несколько хопов друг от друга и это на самом деле практикуется, например, в случае настройки Inter-AS Option C. Называется это дело MultiHop BGP и включается командой "**neighbor XYZ ebgp-multihop**" в режиме конфигурации BGP.
> Но для IBGP это работает по умолчанию.

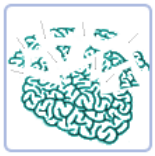
Это позволяет устанавливать IBGP-партнёрство между Loopback-адресами. Делается это для того, чтобы не привязываться к физическим интерфейсам — в случае падения основного линка, BGP-сессия не прервётся, потому что лупбэк будет доступен через резервный.
Это самая распространённая практика.

При этом EBGP однако обычно устанавливается на линковых адресах, потому что как правило имеется только одно подключение и в случае его падения, всё равно Loopback будет не доступен. Да и настраивать ещё какую-то дополнительную маршрутизацию с провайдером не очень-то хочется.

Пример конфигурации такого соседства:

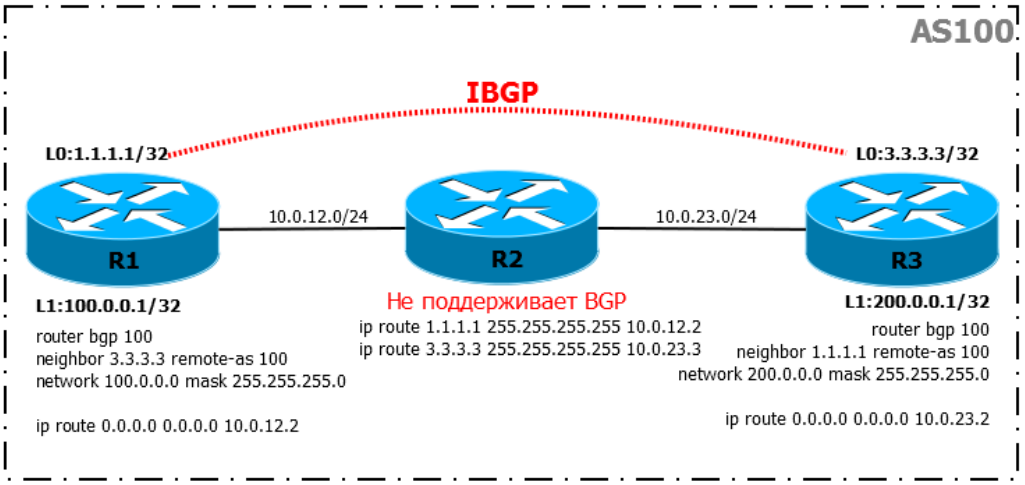


==========



Задача № 1

Схема:



В таком сценарии у нас два BGP-маршрутизатора R1 и R3, но они находятся в разных концах города и подключены через промежуточный маршрутизатор, на котором BGP не настроен.

Условие:
IBGP-сессия прекрасно установится, несмотря даже на то, что на промежуточном маршрутизаторе BGP не включен, и мы видим даже маршруты:

```
R1(config)#do sh ip bgp
BGP table version is 3, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 100.0.0.0/24     0.0.0.0                  0         32768 i
*>i200.0.0.0        3.3.3.3                  0    100      0 i
```

Но где пинг?

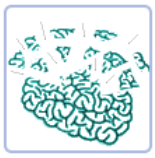
```
R1(config)#do ping 200.0.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 200.0.0.1, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)
```

Подробности задачи тут.
==========

Вам нужно стараться избегать таких ситуаций, когда между IBGP-соседями будут не-BGP маршрутизаторы.

> Вообще-то есть механизм, позволяющий если не исправить, то по крайней мере предупредить такую ситуацию, — IGP Synchronization. Он не позволит добавить маршрут в таблицу, если точно такой же маршрут не известен через IGP. Это в какой-то мере гарантирует, что на промежуточных устройствах, независимо от того, активирован на них BGP или нет, будут нужные маршруты.
> Но я не знаю тех десперадо, которые решились включить IGP Synchronization.
> Во-первых, каким образом такие маршруты попадут в IGP? Только редистрибуцией. Теперь представьте себе, как Full View медленно наполняет LSDB OSPF, проникая в отдалённые уголки памяти и заставляя процессор до изнеможения выискивать кратчайшие маршруты. Хотите ли вы этого?
> А, во-вторых, вытекающее из «во-первых», по умолчанию, IGP-synchronization выключен практически на всех современных маршрутизаторах.

==========



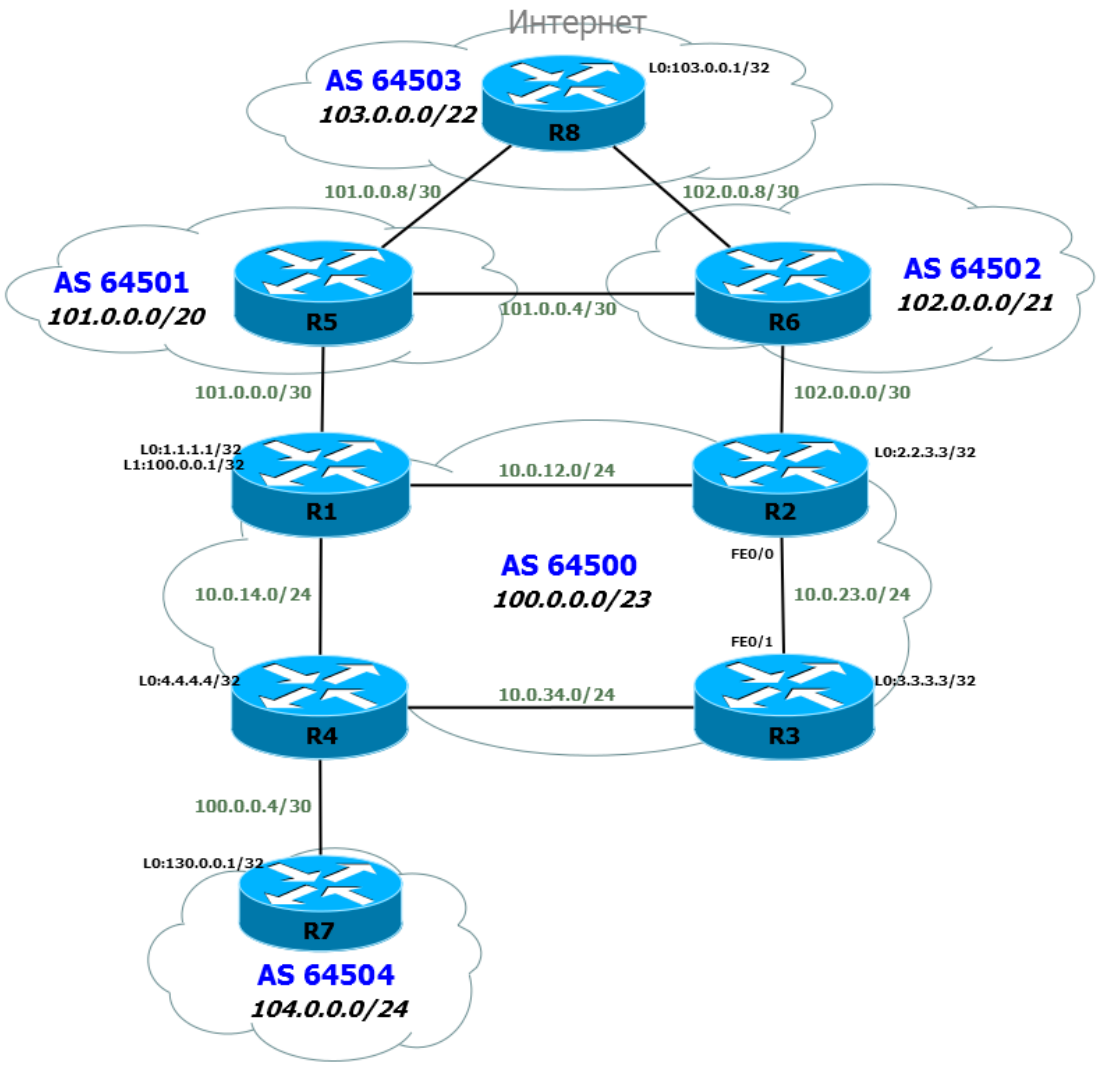
Задача № 2
Между AS64504 и AS64509 появился линк, который связывает их напрямую. Обе сети использовали OSPF и без проблем объединили сеть в одно целое. Но, после проверки, оказалось, что трафик ходит через AS64500, а не напрямую от AS64504 к AS64509, через OSPF.
Изменить конфигурацию BGP:
— R7 должен использовать OSPF, если трафик идет в сеть 109.0.0.0/24
— R9 должен использовать OSPF, если трафик идет в сеть 104.0.0.0/24

Подробности задачи тут
==========

Практика BGP

Давайте теперь вернёмся к сети linkmeup и попробуем запустить BGP в ней.

Схема будет следующей (по клику более подробная с интерфейсами и IP-адресами):



Как видите, мы её значительно видоизменили. Отказались от именования устройств по геолокации и функциям, изменили адресацию в угоду удобства запоминания и понимания.

Маршрутизаторы будем называть R*X*, ликновая подсеть между маршрутизаторами R*X* и R*Y* назначается следующим образом: 10.0.*XY*.0/24. Адреса соответственно 10.0.*XY*.***X*** у R*X* и 10.0.*XY*.***Y*** у *RY*.

Сверху всё осталось также, как было в основной статье по BGP.
А снизу добавился наш первый коммерческий клиент, который подключился по BGP.

==========

Задача № 3

Наш новый клиент AS 64504 подключен к нашей сети. И в дальнейшем планирует подключение к другому провайдеру и у него уже есть свой PI-блок адресов. Но на данном этапе, есть подключение только к нашей AS и поэтому клиент может использовать приватный номер AS.
Задание: При анонсе сети клиента вышестоящим провайдерам, удалить номер приватной AS64504.
Конфигурация и схема: базовые.

Подробности задачи тут.
==========

EBGP

Я думаю, не стоит останавливаться на настройке и работе EBGP — мы сделали это в прошлой статье.
Просто для примера приведём конфигурацию сессии с новым клиентом:

R4

```
interface FastEthernet1/0
 ip address 100.0.0.5 255.255.255.252
```

```
router bgp 64500
 network 100.0.0.0 mask 255.255.254.0
 neighbor 100.0.0.6 remote-as 64504
```

R7

```
interface Loopback1
 ip address 130.0.0.1 255.255.255.255
!
interface FastEthernet0/0
 ip address 100.0.0.6 255.255.255.252
!
router bgp 64504
 network 130.0.0.0 mask 255.255.255.0
 neighbor 100.0.0.5 remote-as 64500
```

Тут всё просто и понятно, после настройки всех внешних соседей мы будем иметь такую ситуацию:

```
R1#sh ip bgp
BGP table version is 7, local router ID is 100.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 100.0.0.0/23     0.0.0.0                  0         32768 i
*> 101.0.0.0/20     101.0.0.1                0             0 64501 i
*> 102.0.0.0/21     101.0.0.1                              0 64501 64502 i
*> 103.0.0.0/22     101.0.0.1                              0 64501 64503 i
R1#sh ip bgp summary
BGP router identifier 100.0.0.1, local AS number 64500
BGP table version is 7, main routing table version 7
4 network entries using 468 bytes of memory
4 path entries using 208 bytes of memory
5/4 BGP path/bestpath attribute entries using 620 bytes of memory
3 BGP AS-PATH entries using 72 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1368 total bytes of memory
BGP activity 5/1 prefixes, 9/5 paths, scan interval 60 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
101.0.0.1       4 64501      18      13        7    0    0 00:07:30        3
R1#
```

На остальных устройствах

Каждый BGP маршрутизатор знает только о тех сетях, которые получены им непосредственно от EBGP-соседа.

IBGP

Теперь обратимся к настройке маршрутизаторов нашей AS с точки зрения IBGP.

Во-первых, как мы говорили ранее, IBGP обычно устанавливается между Loopback-интерфейсами для повышения доступности, поэтому в первую очередь создадим их:

На всех маршрутизаторах на интерфейсе Loopback0 настраиваем IP-адрес X.X.X.X, где X — номер маршрутизатора (это исключительно для примера и не вздумайте такое делать на реальной сети):

R1

```
interface Loopback0
 ip address 1.1.1.1 255.255.255.255
```

R2

```
 interface Loopback0
 ip address 2.2.2.2 255.255.255.255
```

R3

```
interface Loopback0
 ip address 3.3.3.3 255.255.255.255
```

R4

```
interface Loopback0
 ip address 4.4.4.4 255.255.255.255
```

Они станут Router ID и для OSPF и для BGP.

Кстати, об OSPF. Как правило, IBGP «натягивается» поверх существующего на сети IGP. IGP обеспечивает связность всех маршрутизаторов между собой по IP, быструю реакцию на изменения в топологии и перенос маршрутной информации о внутренних сетях.

Настройка внутренней маршрутизации. OSPF

Собственно к этому и перейдём.

Наша задача, чтобы все знали обо всех линковых подсетях, адресах Loopback-интерфейсов и, естественно, о наших белых адресах.

Конфигурация OSPF:

R1

```
router ospf 1
 network 1.1.1.1 0.0.0.0 area 0
 network 10.0.0.0 0.255.255.255 area 0
 network 100.0.0.0 0.0.1.255 area 0
```

R2

```
router ospf 1
 network 2.2.2.2 0.0.0.0 area 0
 network 10.0.0.0 0.255.255.255 area 0
 network 100.0.0.0 0.0.1.255 area 0
```

R3

```
router ospf 1
 network 3.3.3.3 0.0.0.0 area 0
 network 10.0.0.0 0.255.255.255 area 0
 network 100.0.0.0 0.0.1.255 area 0
```

R4

```
router ospf 1
 network 4.4.4.4 0.0.0.0 area 0
 network 10.0.0.0 0.255.255.255 area 0
 network 100.0.0.0 0.0.1.255 area 0
```

После этого появляется связность со всеми Loopback-адресами.

```
R1#sh ip route ospf
     2.0.0.0/32 is subnetted, 1 subnets
O       2.2.2.2 [110/11] via 10.0.12.2, 00:03:57, FastEthernet0/0
     100.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
O       100.0.0.4/30 [110/11] via 10.0.14.4, 00:03:57, FastEthernet0/1
     3.0.0.0/32 is subnetted, 1 subnets
O       3.3.3.3 [110/21] via 10.0.14.4, 00:03:57, FastEthernet0/1
                [110/21] via 10.0.12.2, 00:03:57, FastEthernet0/0
     4.0.0.0/32 is subnetted, 1 subnets
O       4.4.4.4 [110/11] via 10.0.14.4, 00:03:57, FastEthernet0/1
     10.0.0.0/24 is subnetted, 4 subnets
O       10.0.23.0 [110/20] via 10.0.12.2, 00:03:57, FastEthernet0/0
O       10.0.34.0 [110/20] via 10.0.14.4, 00:03:57, FastEthernet0/1
```

Настраиваем BGP

На каждом узле нужно настроить всех соседей вручную:

R1

```
router bgp 64500
 network 100.0.0.0 mask 255.255.254.0
 neighbor 2.2.2.2 remote-as 64500
 neighbor 2.2.2.2 update-source Loopback0
 neighbor 3.3.3.3 remote-as 64500
 neighbor 3.3.3.3 update-source Loopback0
 neighbor 4.4.4.4 remote-as 64500
 neighbor 4.4.4.4 update-source Loopback0
```

Команда вида **neighbor 2.2.2.2 remote-as 64500** объявляет соседа и сообщает, что он находится в AS 64500, BGP понимает, что это та же AS, в которой он сам работает и далее считает 2.2.2.2 своим IBGP-партнёром.

Команда вида **neighbor 2.2.2.2 update-source Loopback0** сообщает, что соединение будет устанавливаться с адреса интерфейса Loopback. Дело в том, что на другой стороне (на 2.2.2.2) сосед настроен, как 1.1.1.1 и именно с этого адреса ждёт все BGP-сообщения.

Такую настройку применяем на всех узлах нашей AS:

R2

```
router bgp 64500
 network 100.0.0.0 mask 255.255.254.0
 neighbor 1.1.1.1 remote-as 64500
 neighbor 1.1.1.1 update-source Loopback0
 neighbor 3.3.3.3 remote-as 64500
 neighbor 3.3.3.3 update-source Loopback0
 neighbor 4.4.4.4 remote-as 64500
 neighbor 4.4.4.4 update-source Loopback0
```

R3

```
router bgp 64500
 network 100.0.0.0 mask 255.255.254.0
 neighbor 1.1.1.1 remote-as 64500
 neighbor 1.1.1.1 update-source Loopback0
 neighbor 2.2.2.2 remote-as 64500
 neighbor 2.2.2.2 update-source Loopback0
 neighbor 4.4.4.4 remote-as 64500
 neighbor 4.4.4.4 update-source Loopback0
```

R4

```
router bgp 64500
 network 100.0.0.0 mask 255.255.254.0
 neighbor 1.1.1.1 remote-as 64500
 neighbor 1.1.1.1 update-source Loopback0
 neighbor 2.2.2.2 remote-as 64500
 neighbor 2.2.2.2 update-source Loopback0
 neighbor 3.3.3.3 remote-as 64500
 neighbor 3.3.3.3 update-source Loopback0
```

Сейчас мы можем проверить, что отношения соседства установились благополучно

```
R1#sh ip bgp summary
BGP router identifier 100.0.0.1, local AS number 64500
BGP table version is 17, main routing table version 17
5 network entries using 585 bytes of memory
10 path entries using 520 bytes of memory
10/5 BGP path/bestpath attribute entries using 1240 bytes of memory
7 BGP AS-PATH entries using 168 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 2513 total bytes of memory
BGP activity 5/0 prefixes, 17/7 paths, scan interval 60 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2.2.2.2         4 64500      24      22       17    0    0 00:04:12        3
3.3.3.3         4 64500       7      21       17    0    0 00:03:41        1
4.4.4.4         4 64500       9      22       17    0    0 00:04:16        2
101.0.0.1       4 64501      15      11       17    0    0 00:04:28        3
```

Все маршруты есть в нашей таблице BGP.

Сеть 130.0.0.0/24 видно на R1:

```
R1#sh ip bgp
BGP table version is 25, local router ID is 100.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i100.0.0.0/23     3.3.3.3                  0    100      0 i
* i                 4.4.4.4                  0    100      0 i
* i                 2.2.2.2                  0    100      0 i
*>                  0.0.0.0                  0         32768 i
* i101.0.0.0/20     102.0.0.1                0    100      0 64502 64501 i
*>                  101.0.0.1                0             0 64501 i
*>i102.0.0.0/21     102.0.0.1                0    100      0 64502 i
*                   101.0.0.1                             0 64501 64502 i
* i103.0.0.0/22     102.0.0.1                0    100      0 64502 64503 i
*>                  101.0.0.1                             0 64501 64503 i
*>i130.0.0.0/24     100.0.0.6                0    100      0 64504 i
```

Сеть 103.0.0.0/22 видно на R4:

```
R4#sh ip bgp
BGP table version is 3, local router ID is 4.4.4.4
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i100.0.0.0/23     3.3.3.3                  0    100      0 i
* i                 1.1.1.1                  0    100      0 i
* i                 2.2.2.2                  0    100      0 i
*>                  0.0.0.0                  0         32768 i
* i101.0.0.0/20     102.0.0.1                0    100      0 64502 64501 i
* i                 101.0.0.1                0    100      0 64501 i
* i102.0.0.0/21     101.0.0.1                0    100      0 64501 64502 i
* i                 102.0.0.1                0    100      0 64502 i
* i103.0.0.0/22     101.0.0.1                0    100      0 64501 64503 i
* i                 102.0.0.1                0    100      0 64502 64503 i
*> i130.0.0.0/24     100.0.0.6                0             0 64504 i
```

Пора проверить сквозной пинг с R7 (нашего клиента) в Интернет (103.0.0.1)?

```
R7#ping 103.0.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 103.0.0.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Приехали.
Не будем долго мучить читателя и сразу посмотрим в таблицу маршрутизации, R4.

```
R4#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     1.0.0.0/32 is subnetted, 1 subnets
O       1.1.1.1 [110/11] via 10.0.14.1, 00:11:45, FastEthernet0/0
     2.0.0.0/32 is subnetted, 1 subnets
O       2.2.2.2 [110/21] via 10.0.34.3, 00:11:45, FastEthernet0/1
                [110/21] via 10.0.14.1, 00:11:45, FastEthernet0/0
     100.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
C       100.0.0.4/30 is directly connected, FastEthernet1/0
S       100.0.0.0/23 is directly connected, Null0
O       100.0.0.1/32 [110/11] via 10.0.14.1, 00:11:45, FastEthernet0/0
     3.0.0.0/32 is subnetted, 1 subnets
O       3.3.3.3 [110/11] via 10.0.34.3, 00:12:06, FastEthernet0/1
     4.0.0.0/32 is subnetted, 1 subnets
C       4.4.4.4 is directly connected, Loopback0
     130.0.0.0/24 is subnetted, 1 subnets
B       130.0.0.0 [20/0] via 100.0.0.6, 00:09:52
     10.0.0.0/24 is subnetted, 4 subnets
C       10.0.14.0 is directly connected, FastEthernet0/0
O       10.0.12.0 [110/20] via 10.0.14.1, 00:12:31, FastEthernet0/0
O       10.0.23.0 [110/20] via 10.0.34.3, 00:12:31, FastEthernet0/1
C       10.0.34.0 is directly connected, FastEthernet0/1
```

А на R7 при этом:

```
R7#sh ip rout
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     100.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       100.0.0.4/30 is directly connected, FastEthernet0/0
B       100.0.0.0/23 [20/0] via 100.0.0.5, 00:00:14
     130.0.0.0/16 is variably subnetted, 2 subnets, 2 masks
S       130.0.0.0/24 is directly connected, Null0
C       130.0.0.1/32 is directly connected, Loopback1
```

А? Где мои маршруты? Где все мои маршруты? R4 ничего не знает про сети Балаган-Телекома, Филькина Сертификата, Интернета, соответственно нет их и на R7.

Помните, мы выше говорили про Next-Hop? Мол, он не меняется при передаче по IBGP?

Обратите внимание на Next-Hop полученных R4 маршрутов:

```
R4#sh ip bgp
BGP table version is 3, local router ID is 4.4.4.4
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i100.0.0.0/23     3.3.3.3                  0    100      0 i
* i                 1.1.1.1                  0    100      0 i
* i                 2.2.2.2                  0    100      0 i
*>                  0.0.0.0                  0         32768 i
* i101.0.0.0/20     102.0.0.1                0    100      0 64502 64501 i
* i                 101.0.0.1                0    100      0 64501 i
* i102.0.0.0/21     101.0.0.1                0    100      0 64501 64502 i
* i                 102.0.0.1                0    100      0 64502 i
* i103.0.0.0/22     101.0.0.1                0    100      0 64501 64503 i
* i                 102.0.0.1                0    100      0 64502 64503 i
*> 130.0.0.0/24     100.0.0.6                0             0 64504 i
```

Несмотря на то, что они пришли на R4 от R1 и R2, адреса Next-Hop на них стоят R5 и R6 — то есть не менялись.

Это значит, что трафик в сеть 103.0.0.0/22 R4 должен отправить на адрес 101.0.0.1, ну, либо на 102.0.0.1. Где они в таблице маршрутизации? Нету их в таблице маршрутизации. Ну, и это естественно — откуда им там взяться.

Для решения этой проблемы у нас есть 3 пути:
1) Настроить статические маршруты до этих адресов — то ещё удовольствие, даже если это шлюз последней надежды.
2) Добавить эти интерфейсы (в сторону провайдеров) в домен IGP-маршрутизации. Тоже вариант, но, как известно, внешние сети не рекомендуется добавлять в IGP.
3) Менять адрес Next-Hop при передаче IBGP-соседям. Красиво и масштабируемо. А ситуации, которая нам помешает это реализовать, просто не может быть.

В итоге добавляем в BGP ещё такую команду: **neghbor 2.2.2.2 Next-Hop-self**. Для каждого соседа, на каждом узле.
После этого мы видим следующую ситуацию,

```
R4#sh ip bgp
BGP table version is 6, local router ID is 4.4.4.4
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i100.0.0.0/23     3.3.3.3                  0    100      0 i
* i                 1.1.1.1                  0    100      0 i
* i                 2.2.2.2                  0    100      0 i
*>                  0.0.0.0                  0         32768 i
*>i101.0.0.0/20     1.1.1.1                  0    100      0 64501 i
*>i102.0.0.0/21     2.2.2.2                  0    100      0 64502 i
*>i103.0.0.0/22     1.1.1.1                  0    100      0 64501 64503 i
* i                 2.2.2.2                  0    100      0 64502 64503 i
*> 130.0.0.0/24     100.0.0.6                0             0 64504 i
```

А уж, как добраться до адреса 1.1.1.1 — мы знаем благодаря OSPF:

```
R4#sh ip route  1.1.1.1
Routing entry for 1.1.1.1/32
  Known via "ospf 1", distance 110, metric 11, type intra area
  Last update from 10.0.14.1 on FastEthernet0/0, 00:21:46 ago
  Routing Descriptor Blocks:
  * 10.0.14.1, from 100.0.0.1, 00:21:46 ago, via FastEthernet0/0
      Route metric is 11, traffic share count is 1
```

Как видите в таблице R7 уже появилась все интересные нам сети.

```
R7#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     102.0.0.0/21 is subnetted, 1 subnets
B       102.0.0.0 [20/0] via 100.0.0.5, 00:04:09
     103.0.0.0/22 is subnetted, 1 subnets
B       103.0.0.0 [20/0] via 100.0.0.5, 00:04:40
     100.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       100.0.0.4/30 is directly connected, FastEthernet0/0
B       100.0.0.0/23 [20/0] via 100.0.0.5, 00:21:43
     101.0.0.0/20 is subnetted, 1 subnets
B       101.0.0.0 [20/0] via 100.0.0.5, 00:04:40
     130.0.0.0/16 is variably subnetted, 2 subnets, 2 masks
S       130.0.0.0/24 is directly connected, Null0
C       130.0.0.1/32 is directly connected, Loopback1
```

Теперь пинг успешной проходит:

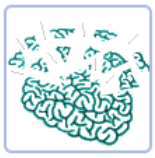
```
R7#ping 103.0.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 103.0.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1560/1744/1924 ms
R7#traceroute 103.0.0.1

Type escape sequence to abort.
Tracing the route to 103.0.0.1

  1 100.0.0.5 [AS 64500] 384 msec 508 msec 376 msec
  2 10.0.14.1 852 msec 760 msec 908 msec
  3 101.0.0.1 [AS 64501] 1196 msec 1336 msec 1220 msec
  4 101.0.0.10 [AS 64501] 1820 msec 2164 msec 2100 msec
```

Очень простой вопрос: откуда такие гигантские задержки в трассировке? А ещё часто и такая ситуация бывает:

```
R7#ping 103.0.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 103.0.0.1, timeout is 2 seconds:
!!..!
Success rate is 60 percent (3/5), round-trip min/avg/max = 1708/1777/1892 ms
```

Конфигурация устройств

==========



Задача № 4

Необходимо настроить такие правила работы с соседними AS:

— от всех соседних AS принимаются префиксы только если в них количество автономных систем в пути не более 10 (в реальной жизни порядок этого значения может быть около 100).

— все префиксы, которые принимаются от клиентов, должны быть с маской не более 24 бит.

Конфигурация и схема: базовые.

Подробности задачи тут.

==========

Что мы можем улучшить?

Разумеется, процесс настройки BGP. Всё-таки это трудозатраты — сделать весьма похожие настройки на каждом узле. Для упрощения вводится понятие peer-group, которая исходя из названия позволяет объединять соседей в группы и одной командой задавать нужные параметры сразу всем.
Дабы не быть голословными, внедрим это на нашей сети:

R1

```
router bgp 64500
 network 100.0.0.0 mask 255.255.254.0
 neighbor AS64500 peer-group
 neighbor AS64500 remote-as 64500
 neighbor AS64500 update-source Loopback0
 neighbor AS64500 Next-Hop-self
 neighbor 2.2.2.2 peer-group AS64500
 neighbor 3.3.3.3 peer-group AS64500
 neighbor 4.4.4.4 peer-group AS64500
```

Команда **neighbor AS64500 peer-group** создаёт группу соседей AS64500.
Команда **neighbor AS64500 remote-as 64500** сообщает, что все соседи находятся в AS 64500.
Команда **neighbor AS64500 update-source Loopback0** указывает, что со всеми соседями соединение будет устанавливаться с адреса Loopback-интрефейса.
Команда **neighbor AS64500 Next-Hop-self** заставляет маршрутизатор менять адрес Next-Hop на свой при передаче анонсов всем соседям.
Дальше, собственно, мы добавляем соседей в эту группу.

Причём мы можем запросто копировать команды конфигурации группы соседей на другие маршрутизаторы, меняя только адреса соседей.

Пара замечаний по Peer-group:
1) Для всех участников группы политики должны быть идентичны.
2) На самом деле cisco уже давно использует динамические Update-группы. Это позволяет сэкономить ресурсы

процессора, так как обработка проводится не по разу на каждого члена группы, а один раз на всю группу. Фактические Peer-группы только облегчают конфигурацию, а оптимизация отдана на откуп Update-групп.

Наверняка, у молодых зелёных инженеров возник вопрос: почему нельзя информацию про публичные адреса передавать по IBGP? Он же, вроде бы, для этого и предназначен? И даже более общий вопрос, почему нельзя обойтись вообще одним BGP, без OSPF или IS-IS, например? (Нет, серьёзно, на форумах иногда вскипают холивары на тему BGP vs OSPF). Ну, по сути ведь тоже протокол маршрутизации — какая разница, передавать информацию между AS или между маршрутизаторами — есть же Internal BGP.
На это я хочу сказать, что достаточно вам будет поработать немного с BGP на реальной сети, чтобы понять всю безумность такой затеи.

Самое главное препятствие — Full Mesh. Придётся устанавливать соседство со всеми всеми маршрутизаторами вручную. OMG, мне дороги моя жизнь и здоровье. (Да, даже не смотря на наличие Route Reflector'ов и скриптов — это лишние операции)
Другая проблема — медленная реакция и Дистанционно-Векторный подход к распространению маршрутной информации.
Да и тут можно резонно возразить, что, дескать, существует BFD. Однако он уменьшит время обнаружение проблемы, но сходимость/восстановление связности всё равно будет медленным.
Третий тонкий момент — отсутствие возможности автоматического изучения соседей. Что ведёт к ручной их конфигурации.
Из всего вышеуказанного вытекают проблемы масштабируемости и обслуживания.

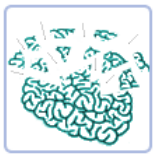
Просто попробуйте сами использовать BGP вместо IGP на сети из 10 маршрутизаторов, и всё станет ясно.

То же самое касается и распространения белых адресов — IBGP с этим справится, но на каждом маршрутизаторе придётся вручную прописывать все подсети.
Ну например, наша сеть 100.0.0.0/23. Допустим, к маршрутизатору R3 подключены 3 клиента по линковым адресам: 100.0.0.8/30, 100.0.0.12/30 и 100.0.0.16/0.
Так вот эти 3 подсети вам нужно будет ввести в BGP тремя командами **network**, в то время как в IGP достаточно активировать протокол на интерфейсе.
Можно, конечно, прибегнуть к хитрой редистрибуции маршрутов из IGP, но это попахивает уже костылями и ещё менее прозрачной конфигурацией.

К чему всё это мы ведём? eBGP — протокол маршрутизации, без дураков. В то же время iBGP — не совсем. Он больше похож на приложение верхнего уровня, организующее распространение маршрутной информации по всей сети. В неизменном виде, а не сообщая при каждой итерации соседу «вон туда через меня». У IGP такое поведение тоже иногда встречается, но там это исключение, а тут — норма.
Я хочу подчеркнуть ещё раз, IGP и IBGP работают в паре, в связке, каждый из них выполняет свою работу.
IGP обеспечивает внутреннюю IP-связность, быструю (читай мгновенную) реакцию на изменения в сети, оповещение всех узлов об этом как можно быстрее. Он же знает о публичных адресах **нашей** AS.
IBGP занимается обработкой Интернетных маршрутов в нашей AS и их транзитом от Uplink'а к клиентам и обратно. Обычно он ничего не знает о структуре внутренней сети.

Если вам пришёл в голову вопрос «что лучше BGP или IS-IS?» — это хорошо, значит у вас пытливый ум, но вы должны отчётливо понимать, что верный ответ тут только один — это принципиально разные вещи, их нельзя сравнивать и выбирать мисс "технология маршрутизации 2013". **IBGP работает поверх IGP**.

==========



Задача № 5
Вышестоящая AS 604503 агрегирует несколько сетей, в том числе и нашу, в один диапазон 100.0.0.0/6. Но этот суммарный префикс вернулся в нашу автономную систему, хотя и не должен был.
Настроить R8 так, чтобы агрегированный префикс не попадал в таблицу BGP маршрутизаторов, которые анонсируют подсети этого префикса. Не использовать фильтрацию для этого.

Конфигурация и схема: базовые.

Подробности задачи тут.
==========

Проблема Эн квадрат

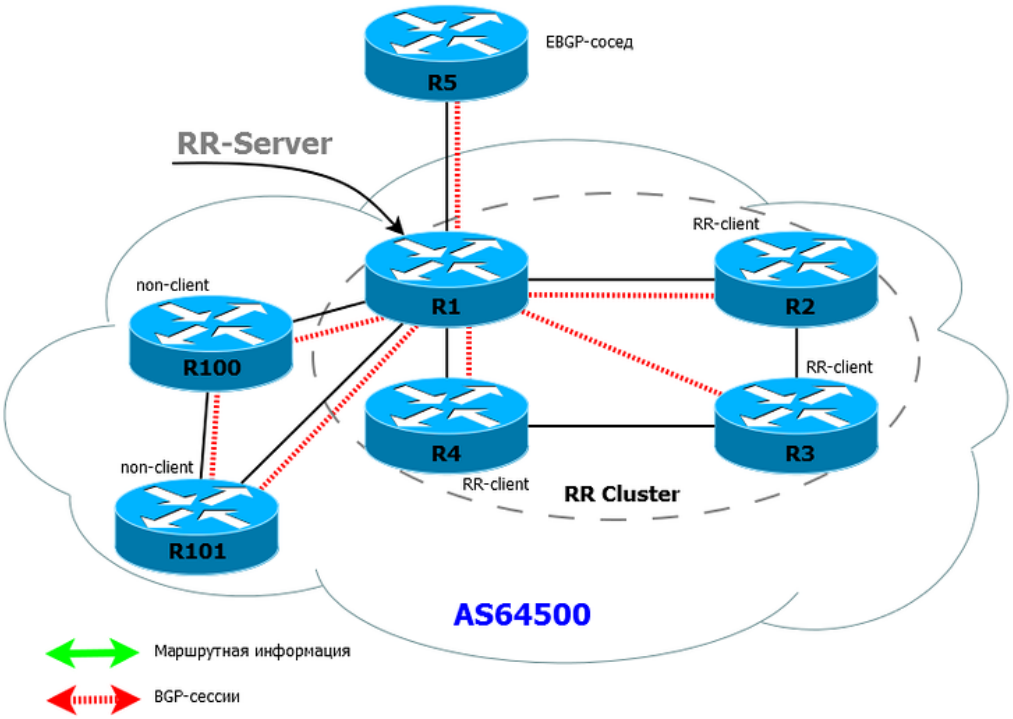
На этом месте тему IBGP можно было бы закрыть, если бы не одно «НО» — Full Mesh. Мы говорили о проблемах полносвязной топологии, когда обсуждали OSPF. Там выходом являлись DR — Designated Router, позволяющие сократить количество связей между маршрутизаторам с n*(n-1)/2 до n-1. Но, если в случае OSPF такая топология была, скорее, исключением, потому что больше 2-3 маршрутизаторов в одном L2-сегменте бывает довольно редко, то для IBGP — это самая обычная практика. У «больших» счёт BGP-маршрутизаторов внутри AS идёт на десятки. А уже для 10 устройств на **каждом** узле нужно будет прописать 9 соседей, то есть всего 45 связей и 90 команд **neighbor** как минимум. Не хило так.
Итак, мы подошли к таким понятиям, как *Route Reflector* и *Confederation*. Уж не знаю почему, но эта тема меня всегда пугала какой-то надуманной сложностью.

Route Reflectror

В чём суть понятия Route Reflector? Это специальный IBGP-маршрутизатор, который, исходя из дословного его перевода, выполняет функцию отражения маршрутов — ему присылает маршрут один сосед, а он рассылает его всем другим. То есть фактически на IBGP-маршрутизаторах вам нужно настроить сессию только с одним соседом — с Route Reflector'ом, а не с девятью. Всё довольно просто и тут прямая аналогия с тем самым DR OSPF.

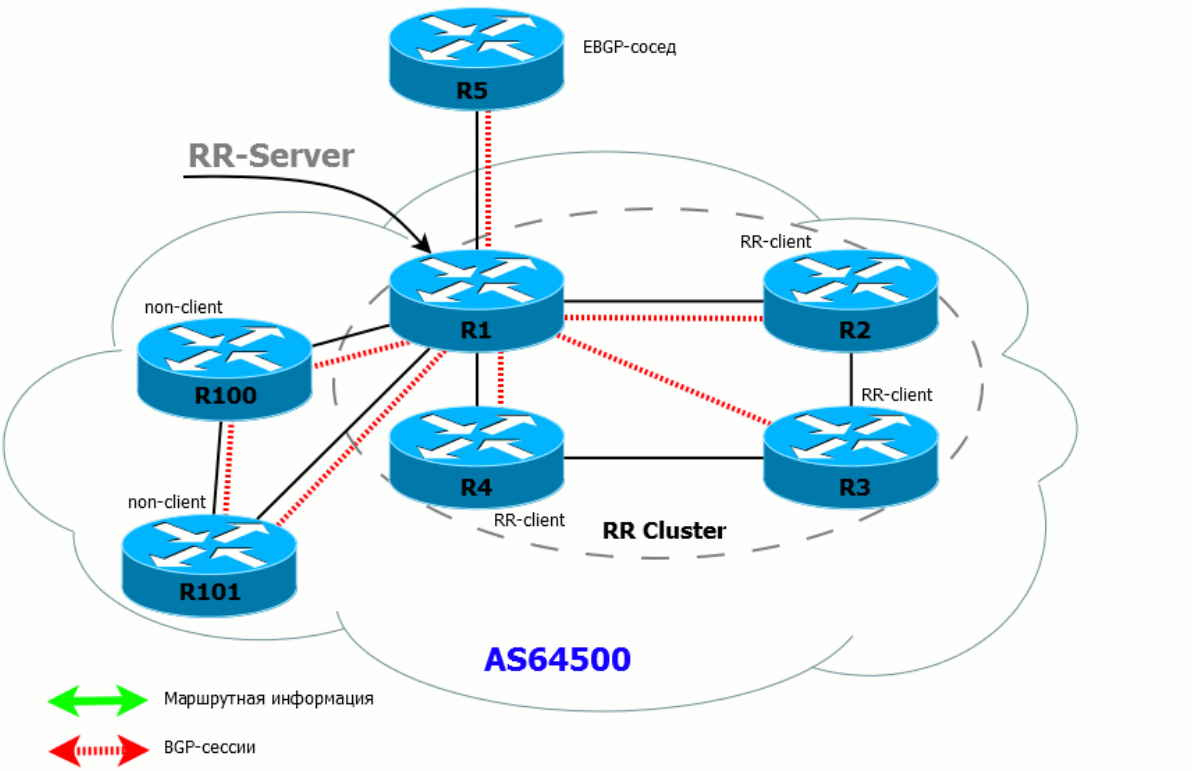
Чуть больше о правилах работы RR.
Во-первых введём понятия *клиент RR* и *не-клиент RR*.

Для данного маршрутизатора клиент — iBGP сосед, который специально объявлен, как RR client, и для которого действуют особые правила. Не-клиент — iBGP сосед, который не объявлен, как RR client
RR серверов может быть (и должно быть в плане отказоустойчивости) несколько. И понятия клиент/не-клиент строго локальны для каждого RR-сервера.
RR-сервер (или несколько) в совокупности с со своими клиентами формируют *кластер*.

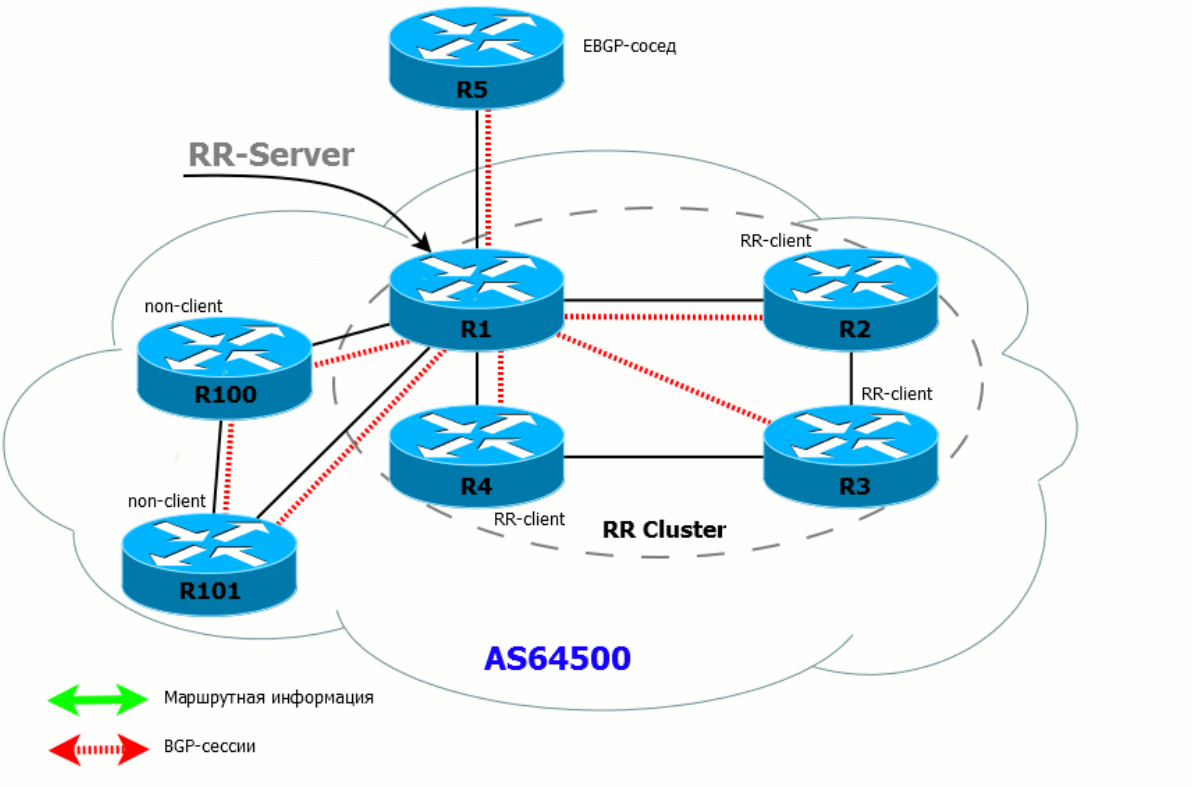


Правила работы RR

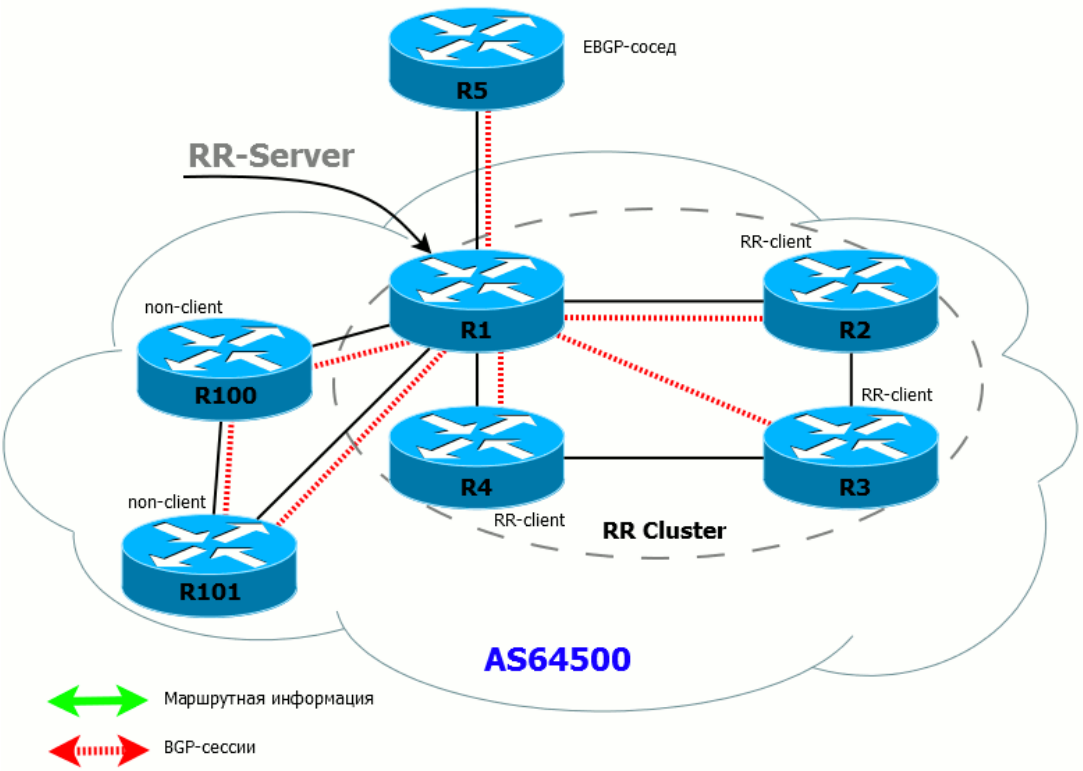
- Если RR получил маршрут от клиента, он отправляет его всем своим клиентам, не-клиентам-соседям и внешним (EBGP) соседям.



- Если RR получил маршрут от не-клиента, он отправляет его всем клиентам и EBGP-соседям. Не-клиентам маршруты НЕ отправляются (потому что они эти маршруты уже получили напрямую от исходного маршрутизатора).



- Если RR получил маршрут от EBGP-соседа, он отправляет его всем своим клиентам, не-клиентам-соседям и внешним соседям.



- Если клиент получил маршрут от RR, он его может отправить только EBGP-соседу.

Как мы сказали выше, в сети может быть несколько Route-reflector'ов. Это нормально, это не вызовет образование петли, потому что существует атрибут Originator ID — как только RR получит маршрут, где указан он сам, как отправитель этого маршрута, он его отбросит. Каждый RR в таком случае будет иметь таблицу маршрутов BGP точно такую же, как у других. Это вынужденная избыточность, позволяющая значительно увеличить стабильность, но при этом у вас должна быть достаточная производительность самих устройств, чтобы, например, поддерживать по паре Full View на каждом.
Но несколько RR могут собираться в кластеры и ~~разрушать деревни~~ обеспечивать экономию ресурсов — таблица BGP будет делиться между несколькими RR.
Принадлежность к одному кластеру настраивается на каждом RR и определяется атрибутом Cluster ID.

> И вот тут тонкий момент — считается, что Best Practice — это настройка одинакового Cluster-ID на всех RR, но на самом деле это не всегда так. Выбирать нужно, исходя из дизайна вашей сети. Более того, часто рекомендуют даже намеренно разделять Route Reflector'ы — как ни странно, это увеличивает стабильность сети.
> Дабы не растекаться мысью по древу, просто дам ссылку на материал об этом.

Вот так выглядит обычная схема с RR:

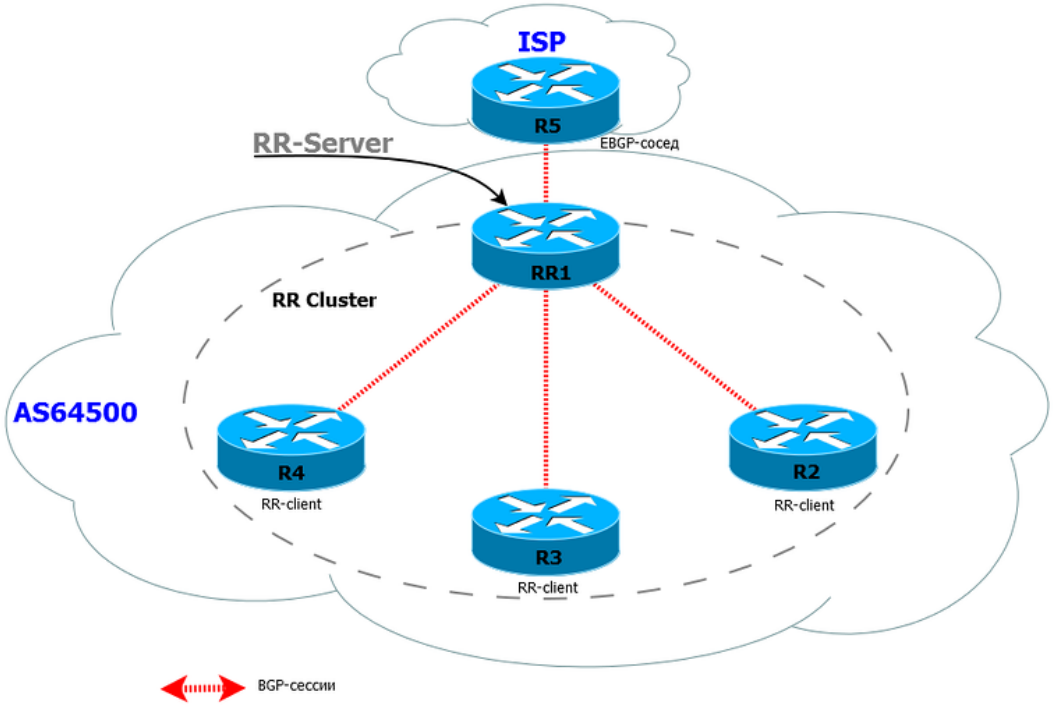
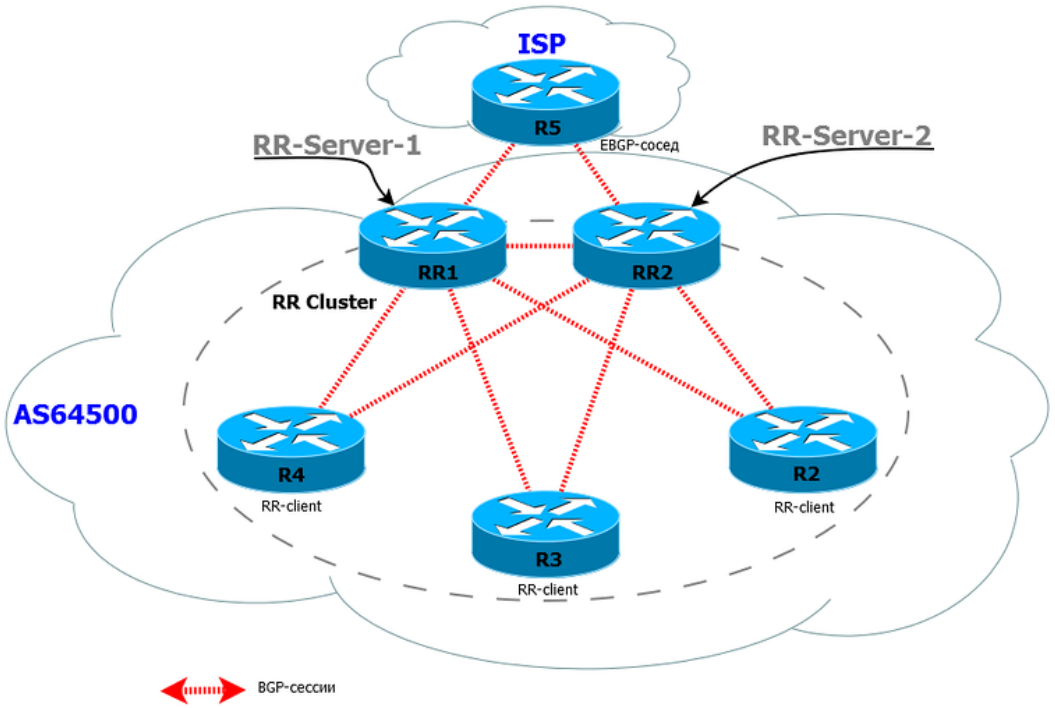
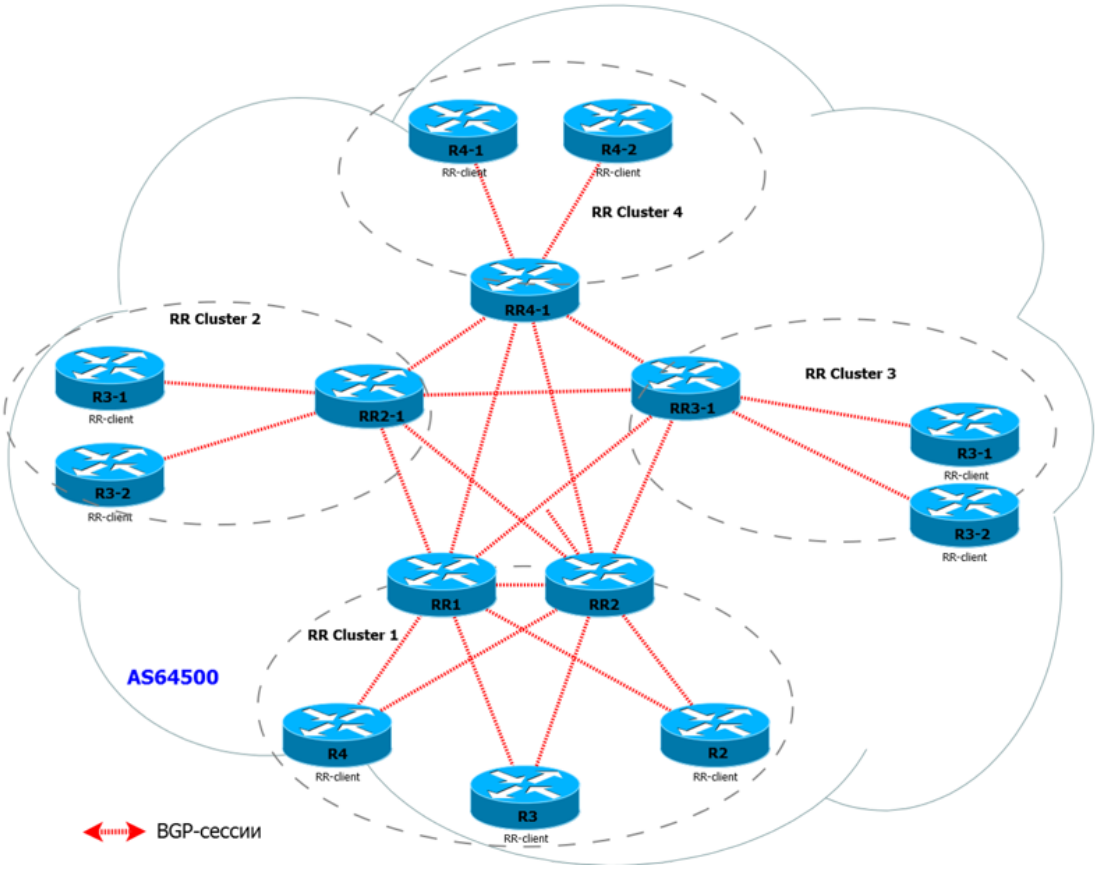


Схема с основным и резервным RR:



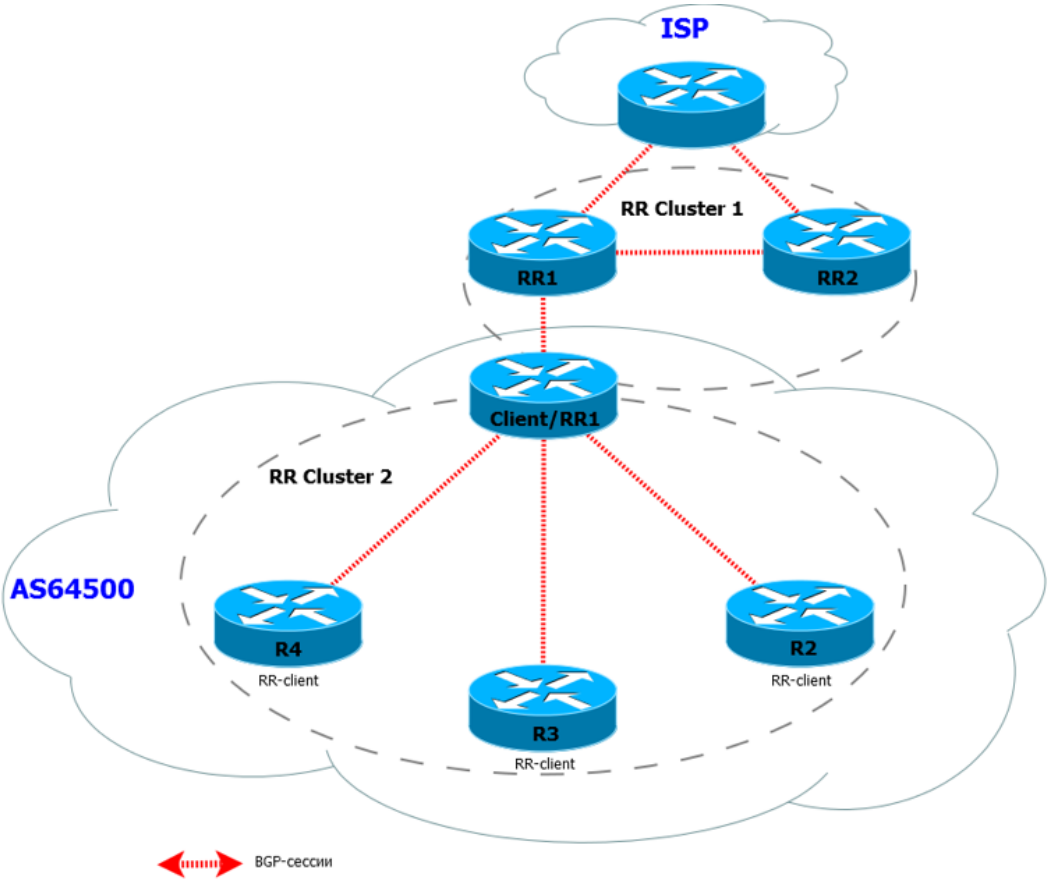
Внутри кластера между всему RR должна быть полная связность.

Кластеров может быть несколько и между ними также следует создавать Full-Mesh сеть:



Повторимся, что кластер: это Рут-рефлектор (один или несколько) вместе со всеми своими клиентами.

Кроме того, часто практикуют иерархические RR. Например, так:



RR1 получает маршруты от удалённой AS и раздаёт их своим дочерним RR (Client/RR1), которые в свою очередь раздают их клиентам.
Это имеет смысл только в достаточно крупных сетях.

Относительно Route Reflector'ов важно понимать, что сам маршрутизатор, выполняющий функции RR не обязательно

участвует в передаче данных. Более того, часто RR специально выносят за пределы пути передачи трафика, чтобы он выполнял исключительно обязанности по передаче маршрутов, чтобы не увеличивать нагрузку на него.

Практика RR

Для примера предположим, что в нашей сети в качестве RR будет выступать R1.
Вот конфигурация самого простого случая RR — одинокого, без кластера.

R1

```
router bgp 64500
 network 100.0.0.0 mask 255.255.254.0
 neighbor AS64500 peer-group
 neighbor AS64500 remote-as 64500
 neighbor AS64500 update-source Loopback0
 <b>neighbor AS64500 route-reflector-client</b>
 neighbor AS64500 Next-Hop-self
 neighbor 2.2.2.2 peer-group AS64500
 neighbor 3.3.3.3 peer-group AS64500
 neighbor 4.4.4.4 peer-group AS64500
 neighbor 101.0.0.1 remote-as 64501
```

R2

```
router bgp 64500
 network 100.0.0.0 mask 255.255.254.0
 neighbor 1.1.1.1 remote-as 64500
 neighbor 1.1.1.1 update-source Loopback0
 neighbor 1.1.1.1 Next-Hop-self
 neighbor 102.0.0.1 remote-as 64502
```

R3

```
router bgp 64500
 network 100.0.0.0 mask 255.255.254.0
 neighbor 1.1.1.1 remote-as 64500
 neighbor 1.1.1.1 update-source Loopback0
 neighbor 1.1.1.1 Next-Hop-self
```

R4

```
router bgp 64500
 network 100.0.0.0 mask 255.255.254.0
 neighbor 1.1.1.1 remote-as 64500
 neighbor 1.1.1.1 update-source Loopback0
 neighbor 1.1.1.1 Next-Hop-self
 neighbor 100.0.0.6 remote-as 64504
```

Обратите внимание на команду "**neighbor AS64500 route-reflector-client**", добавившуюся в настройку R1 и то, что конфигурация BGP на всех других устройствах полностью идентична, за исключением внешних соседей (102.0.0.1 для R2 и 100.0.0.6 для R4).

В общем-то внешне ничего не поменяется. R4, например, всё будет видеть точно также, за исключением количества соседей:

```
R4#sh ip bgp
BGP table version is 43, local router ID is 4.4.4.4
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i100.0.0.0/23     1.1.1.1                  0    100      0 i
*>                  0.0.0.0                  0         32768 i
*>i101.0.0.0/20     1.1.1.1                  0    100      0 64501 i
*>i102.0.0.0/21     2.2.2.2                  0    100      0 64502 i
*>i103.0.0.0/22     1.1.1.1                  0    100      0 64501 64503 i
*> 130.0.0.0/24     100.0.0.6                0             0 64504 i
R4#sh ip bgp summary
BGP router identifier 4.4.4.4, local AS number 64500
BGP table version is 43, main routing table version 43
5 network entries using 585 bytes of memory
6 path entries using 312 bytes of memory
7/5 BGP path/bestpath attribute entries using 868 bytes of memory
1 BGP rrinfo entries using 24 bytes of memory
4 BGP AS-PATH entries using 96 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1885 total bytes of memory
BGP activity 5/0 prefixes, 6/0 paths, scan interval 60 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
1.1.1.1         4 64500      66      33       43    0    0 00:25:27        4
100.0.0.6       4 64504      29      45       43    0    0 00:24:07        1
```

Обратите внимание на то, что Route Reflector не меняет Next-Hop отражённых маршрутов на свой, несмотря на наличие параметра *Next-Hop-self*.

На самом Route Reflector'e отличие будет выглядеть так:

```
R1#sh ip bgp update-group
BGP version 4 update-group 1, external, Address Family: IPv4 Unicast
  BGP Update version : 38/0, messages 0
  Update messages formatted 14, replicated 0
  Number of NLRIs in the update sent: max 1, min 0
  Minimum time between advertisement runs is 30 seconds
  Has 1 member (* indicates the members currently being sent updates):
   101.0.0.1

BGP version 4 update-group 2, internal, Address Family: IPv4 Unicast
  BGP Update version : 38/0, messages 0
  Route-Reflector Client
  NEXT_HOP is always this router
  Update messages formatted 45, replicated 66
  Number of NLRIs in the update sent: max 1, min 1
  Minimum time between advertisement runs is 0 seconds
  Has 3 members (* indicates the members currently being sent updates):
   2.2.2.2          3.3.3.3          4.4.4.4
```

Если смотреть по конкретным маршрутам:

```
R1#sh ip bgp 103.0.0.0
BGP routing table entry for 103.0.0.0/22, version 5
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to update-groups:
     2
  64502 64503, (Received from a RR-client)
    2.2.2.2 (metric 11) from 2.2.2.2 (2.2.2.2)
      Origin IGP, metric 0, localpref 100, valid, internal
  64501 64503
    101.0.0.1 from 101.0.0.1 (5.5.5.5)
      Origin IGP, localpref 100, valid, external, best
```

Здесь видно полную подсеть, количество путей до неё, какой из них лучший, в какую таблицу он добавлен, куда передаётся (update-group 2 — как раз наш кластер).
Далее перечисляются все эти пути, содержащие такие важные параметры, как AS-Path, Next-Hop, Origin итд, а также информацию о том, что например, первый маршрут было получен от RR-клиента.

Эту информацию можно успешно использовать для траблшутинга. Вот так, например выглядит её вывод, когда не настроен Next-Hop-self:

```
R4#sh ip bgp 103.0.0.0
BGP routing table entry for 103.0.0.0/22, version 47
Paths: (1 available, no best path)
Flag: 0x820
  Advertised to update-groups:
     2
  64501 64503
    101.0.0.1 (inaccessible) from 1.1.1.1 (100.0.0.1)
      Origin IGP, metric 0, localpref 100, valid, internal
```

Конфигурация устройств.

Проблема резервирования

Какая сейчас с рут-рефлектором есть проблема? У всех маршрутизаторов связи установлены только с ним. И если R1 вдруг выйдет из строя, пиши пропало — сеть ляжет.
Для этих целей, давайте настроим кластер и в качестве второго RR выберем R2.
То есть теперь на R3 и R4 нужно поднимать соседства не только с R1, но и с R2.

Теперь sh ip bgp update-group выглядит так:

```
R1#sh ip bgp update-group
BGP version 4 update-group 1, external, Address Family: IPv4 Unicast
  BGP Update version : 40/0, messages 0
  Update messages formatted 16, replicated 0
  Number of NLRIs in the update sent: max 1, min 0
  Minimum time between advertisement runs is 30 seconds
  Has 1 member (* indicates the members currently being sent updates):
   101.0.0.1

BGP version 4 update-group 2, internal, Address Family: IPv4 Unicast
  BGP Update version : 40/0, messages 0
  Route-Reflector Client
  NEXT_HOP is always this router
  Update messages formatted 12, replicated 17
  Number of NLRIs in the update sent: max 1, min 1
  Minimum time between advertisement runs is 0 seconds
  Has 2 members (* indicates the members currently being sent updates):
   3.3.3.3          4.4.4.4

BGP version 4 update-group 3, internal, Address Family: IPv4 Unicast
  BGP Update version : 40/0, messages 0
  NEXT_HOP is always this router
  Update messages formatted 10, replicated 0
  Number of NLRIs in the update sent: max 1, min 0
  Minimum time between advertisement runs is 0 seconds
  Has 1 member (* indicates the members currently being sent updates):
   2.2.2.2
```

Один внешний, один внутренний — не RR-клиент и два внутренних RR-клиента.
Аналогично на R2:

```
R2(config-router)#do sh ip bgp upd
BGP version 4 update-group 1, internal, Address Family: IPv4 Unicast
  BGP Update version : 42/0, messages 0
  NEXT_HOP is always this router
  Update messages formatted 41, replicated 0
  Number of NLRIs in the update sent: max 1, min 0
  Minimum time between advertisement runs is 0 seconds
  Has 1 member (* indicates the members currently being sent updates):
   1.1.1.1

BGP version 4 update-group 2, external, Address Family: IPv4 Unicast
  BGP Update version : 42/0, messages 0
  Update messages formatted 20, replicated 0
  Number of NLRIs in the update sent: max 1, min 0
  Minimum time between advertisement runs is 30 seconds
  Has 1 member (* indicates the members currently being sent updates):
   102.0.0.1

BGP version 4 update-group 3, internal, Address Family: IPv4 Unicast
  BGP Update version : 42/0, messages 0
  Route-Reflector Client
  NEXT_HOP is always this router
  Update messages formatted 11, replicated 0
  Number of NLRIs in the update sent: max 1, min 1
```

На клиентах у нас теперь два соединения с RR:

```
R4#sh ip bgp summary
BGP router identifier 4.4.4.4, local AS number 64500
BGP table version is 53, main routing table version 53
5 network entries using 585 bytes of memory
10 path entries using 520 bytes of memory
8/5 BGP path/bestpath attribute entries using 992 bytes of memory
2 BGP rrinfo entries using 48 bytes of memory
5 BGP AS-PATH entries using 120 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 2265 total bytes of memory
BGP activity 5/0 prefixes, 10/0 paths, scan interval 60 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
1.1.1.1         4 64500     184     135       53    0    0 02:05:26        4
2.2.2.2         4 64500      15      11       53    0    0 00:05:01        4
100.0.0.6       4 64504     129     151       53    0    0 02:04:06        1
```

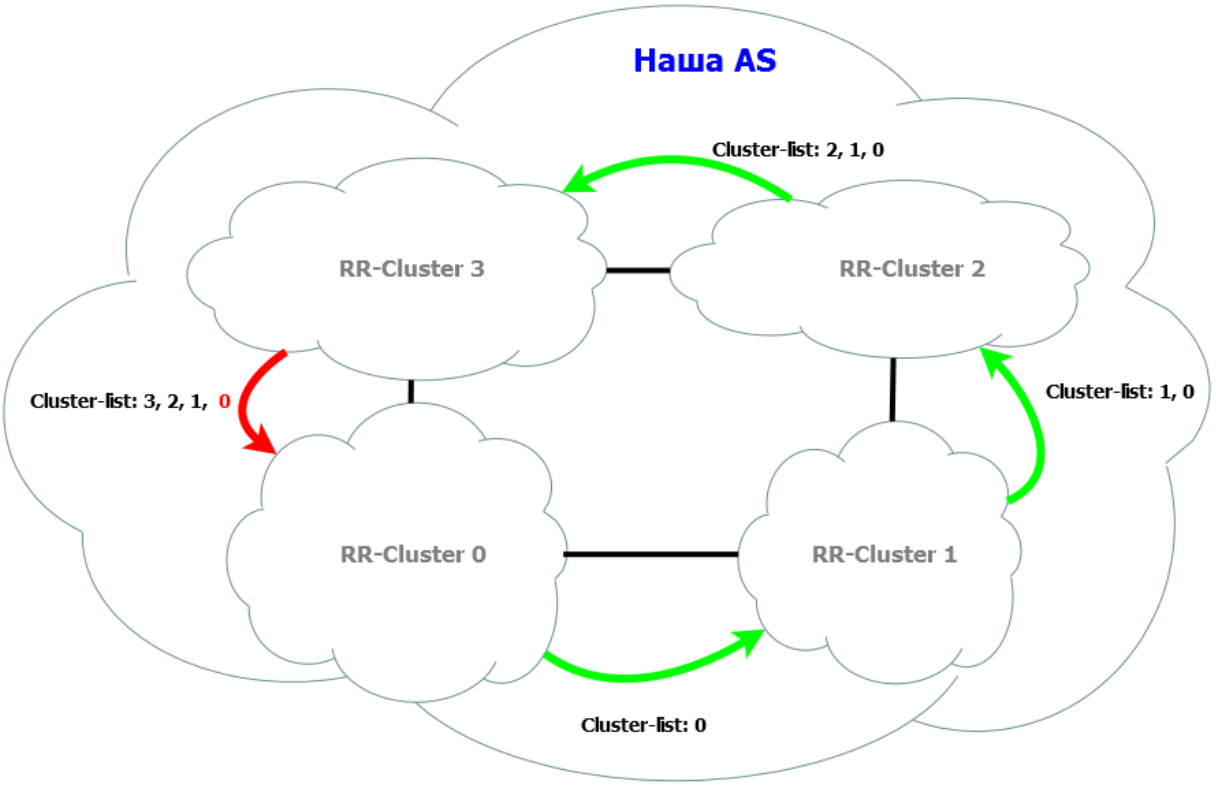
Обратите внимание, в сообщениях Update теперь появились два новых атрибута: Cluster-List и Originator-ID. Исходя из названия, они несут в себе номер RR-кластера и идентификатор отправителя анонса:

R1R2

```
    33 16:03:52.985577    1.1.1.1         3.3.3.3         BGP      375 UPDATE Message, UPDATE Message, UPDATE Message, UPDATE Message,
    34 16:03:53.125577    1.1.1.1         2.2.2.2         BGP      302 UPDATE Message, UPDATE Message, UPDATE Message, UPDATE Message
    35 16:03:53.250577    1.1.1.1         2.2.2.2         TCP       60 bgp > 50836 [ACK] Seq=291 Ack=268 Win=15854 Len=0
⊞ Ethernet II, Src: c0:01:3a:08:00:00 (c0:01:3a:08:00:00), Dst: c0:02:3a:08:00:01 (c0:02:3a:08:00:01)
⊞ Internet Protocol Version 4, Src: 1.1.1.1 (1.1.1.1), Dst: 2.2.2.2 (2.2.2.2)
⊞ Transmission Control Protocol, Src Port: bgp (179), Dst Port: 50836 (50836), Seq: 43, Ack: 20, Len: 248
⊞ Border Gateway Protocol - UPDATE Message
⊞ Border Gateway Protocol - UPDATE Message
⊟ Border Gateway Protocol - UPDATE Message
    Marker: ffffffffffffffffffffffffffffffff
    Length: 59
    Type: UPDATE Message (2)
    Unfeasible routes length: 0 bytes
    Total path attribute length: 32 bytes
  ⊟ Path attributes
    ⊞ ORIGIN: IGP (4 bytes)
    ⊞ AS_PATH: 64501 (7 bytes)
    ⊞ NEXT_HOP: 1.1.1.1 (7 bytes)
    ⊞ MULTI_EXIT_DISC: 0 (7 bytes)
    ⊞ LOCAL_PREF: 100 (7 bytes)
  ⊟ Network layer reachability information: 4 bytes
    ⊞ 101.0.0.0/20
⊟ Border Gateway Protocol - UPDATE Message
    Marker: ffffffffffffffffffffffffffffffff
    Length: 73
    Type: UPDATE Message (2)
    Unfeasible routes length: 0 bytes
    Total path attribute length: 46 bytes
  ⊟ Path attributes
    ⊞ ORIGIN: IGP (4 bytes)
    ⊞ AS_PATH: 64504 (7 bytes)
    ⊞ NEXT_HOP: 4.4.4.4 (7 bytes)
    ⊞ MULTI_EXIT_DISC: 0 (7 bytes)
    ⊞ LOCAL_PREF: 100 (7 bytes)
    ⊞ CLUSTER_LIST: 0.0.0.1 (7 bytes)
    ⊞ ORIGINATOR_ID: 4.4.4.4 (7 bytes)
  ⊟ Network layer reachability information: 4 bytes
    ⊞ 130.0.0.0/24
```

Эти параметры добавляются только маршрутам, передающимся по IBGP.

Они необходимо для того, чтобы избежать образования петель. Если, например, маршрут прошёл несколько кластеров и вернулся в исходный, то в параметре Cluster-List среди всех прочих, маршрутизатор увидит номер своего кластера, и после этого удалит маршрут.



Попробуйте ответить на вопрос, зачем нужен атрибут Originator-ID? Разве Cluster-List не исчерпывает все варианты?

Если сейчас даже сжечь R1, то связь частично ляжет только на время обнаружения проблемы и перестроения таблиц маршрутизации (в худшем случае это 3 минуты ожидания Keepalive сообщения BGP и ещё какое-то время на изучение новых маршрутов).
Но, если дизайн сети у вас предполагал, что RR — это самостоятельные железки, и через них не ходил трафик (то есть они занимались исключительно распространением маршрутов), то, вполне вероятно, что перерыва трафика не будет вовсе. Во-первых, отправитель только через 3 минуты заметит, что что-то не так с RR — в течение этого времени маршрут у него всё-

равно будет, а поскольку он ведёт не через бесславно погибший RR, трафик будет ходить вполне благополучно. По прошествии этих трёх минут отправитель переключится на резервный RR и получит от него новый актуальный маршрут. Таким образом связь не будет прервана.

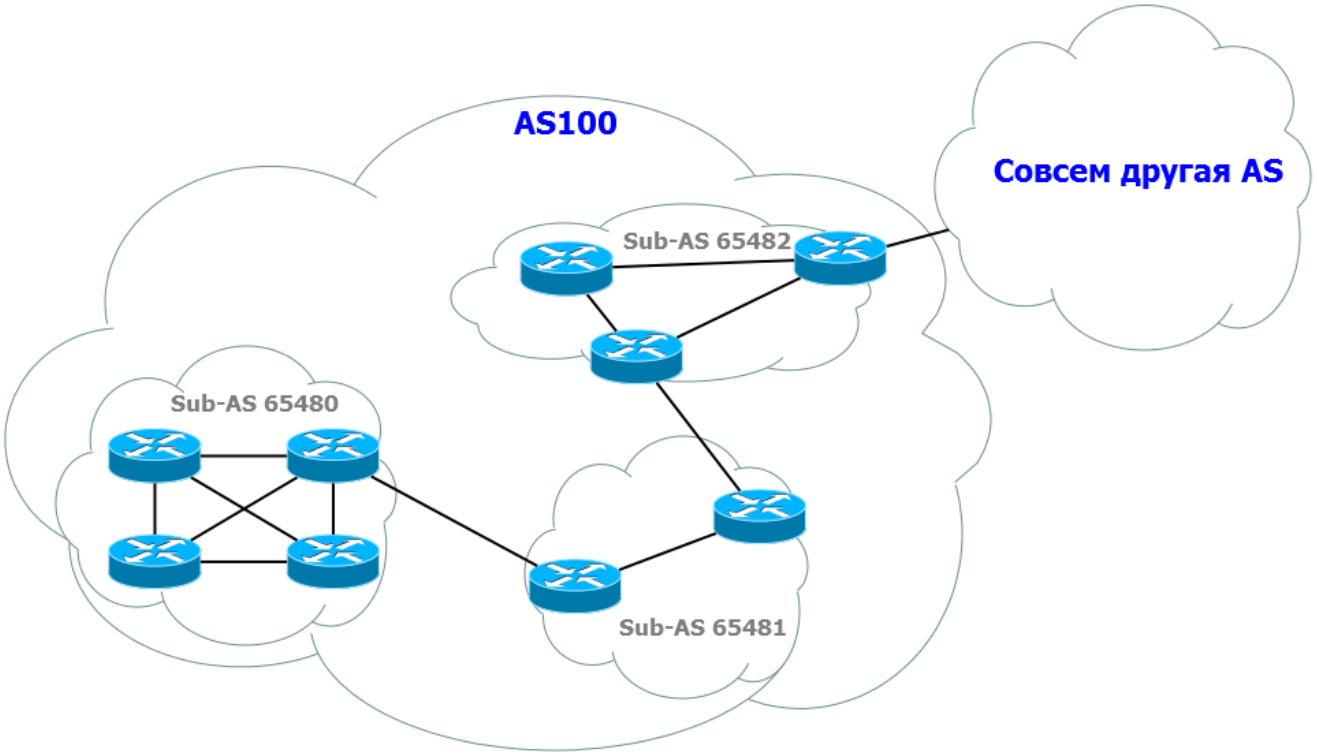
Суть иерархических рут-рефлекторов лишь в том, что один из них является клиентом другого. Это помогает выстроить более понятную и прозрачную схему работы, которую будет проще траблшутить далее.
На нашей сети это лишено какого бы то ни было смысла, поэтому данный случай рассматривать не будем.

Конфедерации

Другой способ решения проблемы Full-Mesh — это конфедерации или иначе их называют sub-AS, под-АС. По сути — это маленькая виртуальная АС внутри большой настоящей АС.

Каждая конфедерация ведёт себя как взрослая AS — внутри полная связность, снаружи, как Лейбниц на душу положит — IBGP работает тут по принципу EBGP (с некоторыми оговорками), граничные маршрутизаторы конфедераций, ведут себя как EBGP-соседи, должны быть подключены напрямую.

Пример топологии:



Когда маршруты передаются внутри АС между конфедерациями в их AS-Path добавляется номер конфедерации (сегменты AS_CONFED_SEQ и AS_CONFED_SET) для избежания петель. Как только маршрут покидает AS, удаляются все эти номера, чтобы внешний мир о них не знал.

Встречается он довольно редко из-за своей слабой масштабируемости и непрозрачности, поэтому рассматривать мы его не будем.
Более подробно можно почитать на xgu.ru.

Атрибуты BGP

Последняя тема, которую мы затронем касательно BGP — это его атрибуты. Мы их уже начали рассматривать в основной статье (AS-Path и Next-Hop, например). Теперь же имеющиеся знания систематизируем и дополним.

Они делятся на четыре типа:

- Хорошо известные обязательные (Well-known Mandatory)
- Хорошо известные необязательные (Well-known Discretionary)

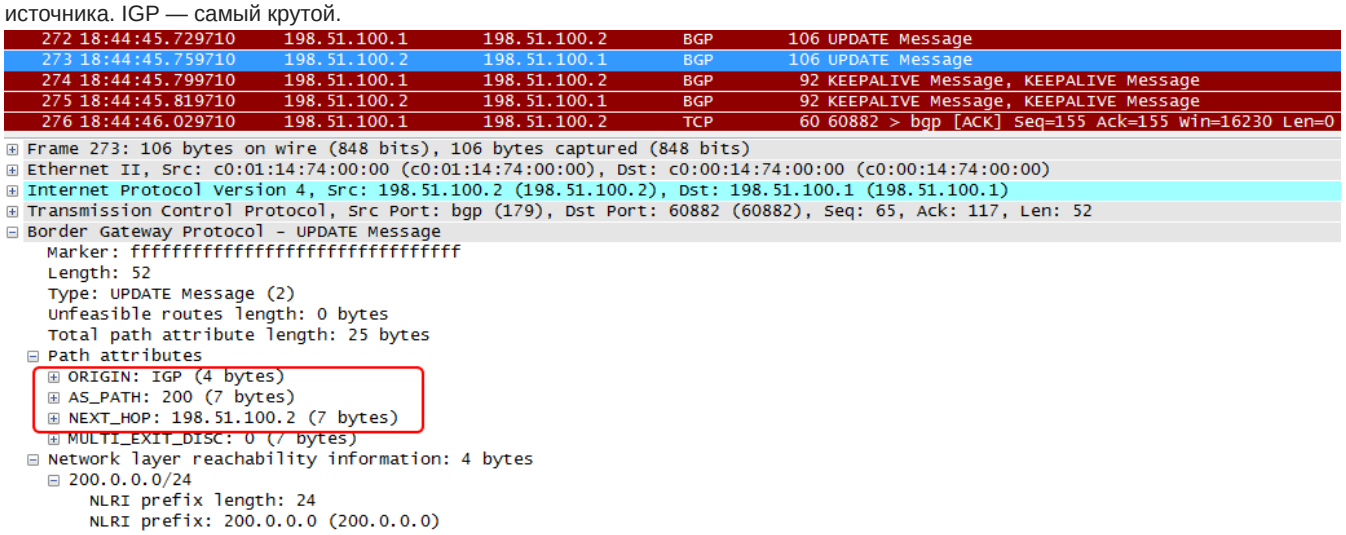
- Опциональные передаваемые/транзитивные (Optional Transitive)
- Опциональные непередаваемые/нетранзитивные (Optional Non-transitive)

Хорошо известные обязательные (Well-known Mandatory)

Это атрибуты, которые должны присутствовать в анонсах **всегда**, и **каждый** BGP-маршрутизатор должен их знать. Следующие три атрибута и только они принадлежат к этому типу.
Next-Hop говорит маршрутизатору, получающему анонс, о том, куда отправлять пакет.
При передаче маршрута между различными AS значение Next-Hop меняется на адрес отправляющего маршрутизатора. Внутри AS атрибут Next-Hop по умолчанию не меняется при передаче от одного IBGP-оратора другому. Выше мы уже рассматривали почему.

AS-path несёт в себе список всех Автономных Систем, которые нужно преодолеть для достижения цели. Используется для выбора лучшего пути и для исключения петель маршрутизации. Когда маршрут передаётся из одной AS в другую, в AS-path вставляется номер **отправляющей** AS. При передаче внутри AS параметр не меняется.

Origin сообщает, как маршрут зародился — командой network (IGP — значение 0) или редистрибуцией (Incomplete — значение 2). Значение 1 (EGP) — уже не встречается ввиду того, что протокол EGP не используется. Назначается единожды маршрутизатором-папой, сгенерировавшим маршрут, и более нигде не меняется. По сути означает степень надёжности источника. IGP — самый крутой.



Хорошо известные необязательные (Well-known Discretionary)

Эти атрибуты должны знать все BGP-маршрутизаторы, но их присутствие в анонсе не требуется. Хочешь — есть, не хочешь — не есть.

Примеры:
Local Preference помогает выбрать один из нескольких маршрутов в одну сеть. Данный атрибут может передаваться лишь в пределах одной AS. Если анонс с Local Preference приходит от EBGP-партнёра, атрибут просто игнорируется — мы не можем с помощью Local Preference управлять маршрутами чужой AS.

Atomic Aggregate говорит о том, что префикс был получен путём агрегирования более мелких.

Опциональные передаваемые/транзитивные (Optional Transitive)

Атрибуты, которые не обязательно знать всем. Кто знает — использует, кто не знает — передаёт их дальше.

Примеры:
Aggregator. Указывает на Router ID маршрутизатора, где произошло агрегирование.

Community. Про этот атрибут мы подробно поговорим далее, в заключительной части статьи.

Опциональные непередаваемые/нетранзитивные (Optional Non-transitive)

Атрибуты, которые не обязательно знать всем. Но маршрутизатор, который их не поддерживает, их отбрасывает и никуда дальше не передаёт.

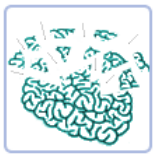
Пример:

MED — Multi-exit Descriminator. Этим атрибутом мы можем попытаться управлять приоритетами в чужой AS. Можем попытаться, но вряд ли что-то получится :) Часто этот атрибут фильтруется, он имеет значение только при наличии как минимум двух линков в одну AS, он проверяется после многих очень сильных атрибутов (Local Preference, AS-Path), да и разные вендоры могу по-разному трактовать MED.

```
   272 18:44:45.729710    198.51.100.1    198.51.100.2    BGP      106 UPDATE Message
   273 18:44:45.759710    198.51.100.2    198.51.100.1    BGP      106 UPDATE Message
   274 18:44:45.799710    198.51.100.1    198.51.100.2    BGP       92 KEEPALIVE Message, KEEPALIVE Message
   275 18:44:45.819710    198.51.100.2    198.51.100.1    BGP       92 KEEPALIVE Message, KEEPALIVE Message
   276 18:44:46.029710    198.51.100.1    198.51.100.2    TCP       60 60882 > bgp [ACK] Seq=155 Ack=155 Win=16230 Len=0
⊞ Frame 273: 106 bytes on wire (848 bits), 106 bytes captured (848 bits)
⊞ Ethernet II, Src: c0:01:14:74:00:00 (c0:01:14:74:00:00), Dst: c0:00:14:74:00:00 (c0:00:14:74:00:00)
⊞ Internet Protocol Version 4, Src: 198.51.100.2 (198.51.100.2), Dst: 198.51.100.1 (198.51.100.1)
⊞ Transmission Control Protocol, Src Port: bgp (179), Dst Port: 60882 (60882), Seq: 65, Ack: 117, Len: 52
⊟ Border Gateway Protocol - UPDATE Message
    Marker: ffffffffffffffffffffffffffffffff
    Length: 52
    Type: UPDATE Message (2)
    Unfeasible routes length: 0 bytes
    Total path attribute length: 25 bytes
  ⊟ Path attributes
    ⊞ ORIGIN: IGP (4 bytes)
    ⊞ AS_PATH: 200 (7 bytes)
    ⊞ NEXT_HOP: 198.51.100.2 (7 bytes)
    ⊞ MULTI_EXIT_DISC: 0 (7 bytes)
  ⊟ Network layer reachability information: 4 bytes
    ⊟ 200.0.0.0/24
        NLRI prefix length: 24
        NLRI prefix: 200.0.0.0 (200.0.0.0)
```

Упомянутые прежде ***Cluster List*** и ***Originator-ID***. Естественно, они являются опциональными, и естественно, передавать их куда-то за пределы AS нет смысла, поэтому и непередаваемые.

==========



Задача № 6*

Необходимо изменить стандартную процедуру выбора лучшего маршрута на маршрутизаторах в AS64500:
— маршрутизаторы R1 и R2 должны выбирать маршруты eBGP, а не iBGP, независимо от длины AS path,
— маршрутизаторы R3 и R4 внутри автономной системы должны выбирать маршруты на основании метрики OSPF.

Конфигурация и схема: базовые.

Подробности задачи тут.
==========

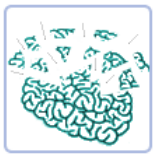
Community

Вот он — один из самых интересных аспектов BGP, вот где проявляется его гибкость — возможность помимо самих маршрутов, передавать дополнительную информацию.
С помощью атрибута Community можно из своей AS управлять поведением маршрутизаторов другой AS.
Я долгое время по непонятной сейчас для себя причине недооценивал мощь этого инструмента.
Управление своими анонсами в чужой AS с помощью community поддерживается подавляющим большинством вендоров. Но на самом деле говорить тут надо не о вендорах, а, скорее, о операторах/провайдерах — именно от них зависит, от их конфигурации, сможете ли вы управлять или нет.

Начнём с теории, Community, как было сказано выше, — это опциональный передаваемый атрибут (Optional Transitive) размером 4 байта. Он представляет из себя запись вида AA:NN, где AA — двухбайтовый номер AS, NN — номер коммьюнити (например, 64500:666).

Существует 4 так называемых **Well-Known community** (**хорошо известные**):
Internet — Нет никаких ограничений — передаётся всем.
No-export — Нельзя экспортировать маршрут в другие AS. При этом за пределы конфедерации передавать их можно.
No-export-subconfed (называется также ***Local AS***) — Как No-export, только добавляется ограничение и по конфедерациям — между ними уже тоже не передаётся.
No-advertise — Не передавать этот маршрут никому — только сосед будет знать о нём.

==========



Задача № 7

Наш новый клиент AS 64504 подключен к нашей сети. И пока что не планирует подключение к другому провайдеру. На данном этапе клиент может использовать номер автономной системы из приватного диапазона. Блок адресов, который использует клиент, будет частью нашего диапазона сетей.
Задание: Так как сеть клиента является частью нашего блока адресов, надо чтобы сеть клиента не анонсировалась соседним провайдерам.
Не использовать фильтрацию префиксов или фильтрацию по AS для решения этой задачи.

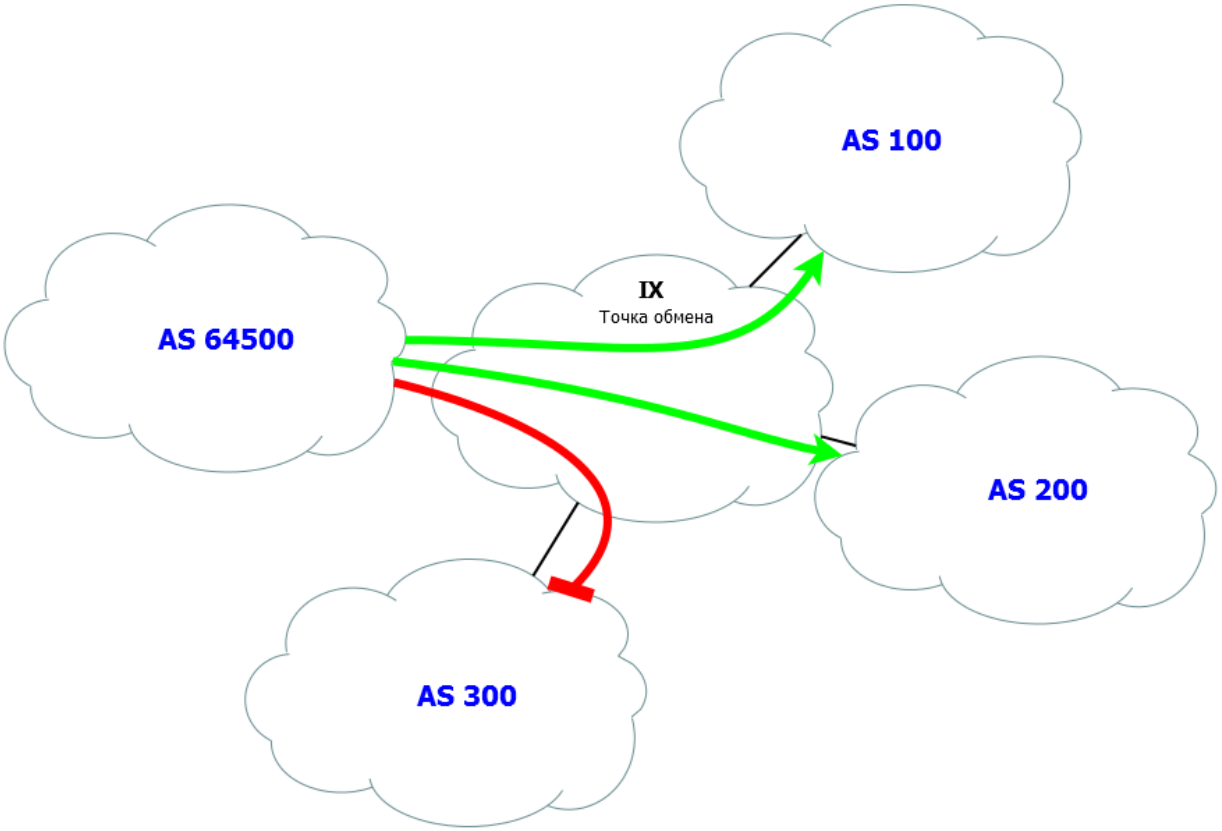
Конфигурация и схема: Community.
Отличия только в том, что сеть, которую анонсирует AS64504: 100.0.1.0/28, а не 130.0.0.0/24

Подробности задачи тут.
==========

В сети тысячи примеров настройки таких базовых коммьюнити и крайне мало примеров реального использования.
А меж тем одно из самых интересных применений этого атрибута — блэкхоулинг от старославянского black hole — способ борьбы с DoS-атаками. Очень подробно с примером настройки о нём уже было рассказано на хабре.
Суть в том, что когда началась атака на какой-то из адресов вашей AS, вы этот адрес передаёте вышестоящему провайдеру с комьюнити 666, и он отправляет такой маршрут в NULL — блэкхолит его. То есть до вас уже этот паразитный трафик не доходит. Провайдер может передать такой маршрут дальше, и так, шаг за шагом, трафик от злоумышленника или системы ботов будет отбрасываться уже на самых ранних этапах, не засоряя Интернет.
Достигается такой эффект расширяющейся чёрной дыры именно благодаря коммьюнити. То есть в обычном случае вы анонсируете этот адрес в составе большой сети /22, например, а в случае DoS'а передаёте самый специфичный маршрут /32, который будет, естественно, более приоритетным.

> О таких атаках вы, кстати, можете послушать в шестом выпуске нашего подкаста linkmeup.

Другие примеры — управление атрибутом Local Preference в чужой AS, сообщать ему, что анонсу нужно увеличить AS-path (AS-path prepending) или не передавать маршрут каким-либо соседям.
Насчёт последнего. Как, например, вы решите следующую задачу?
Имеется сеть, представленная на рисунке ниже. Вы хотите отдавать свои маршруты соседям из AS 100 и 200 и не хотите 300.

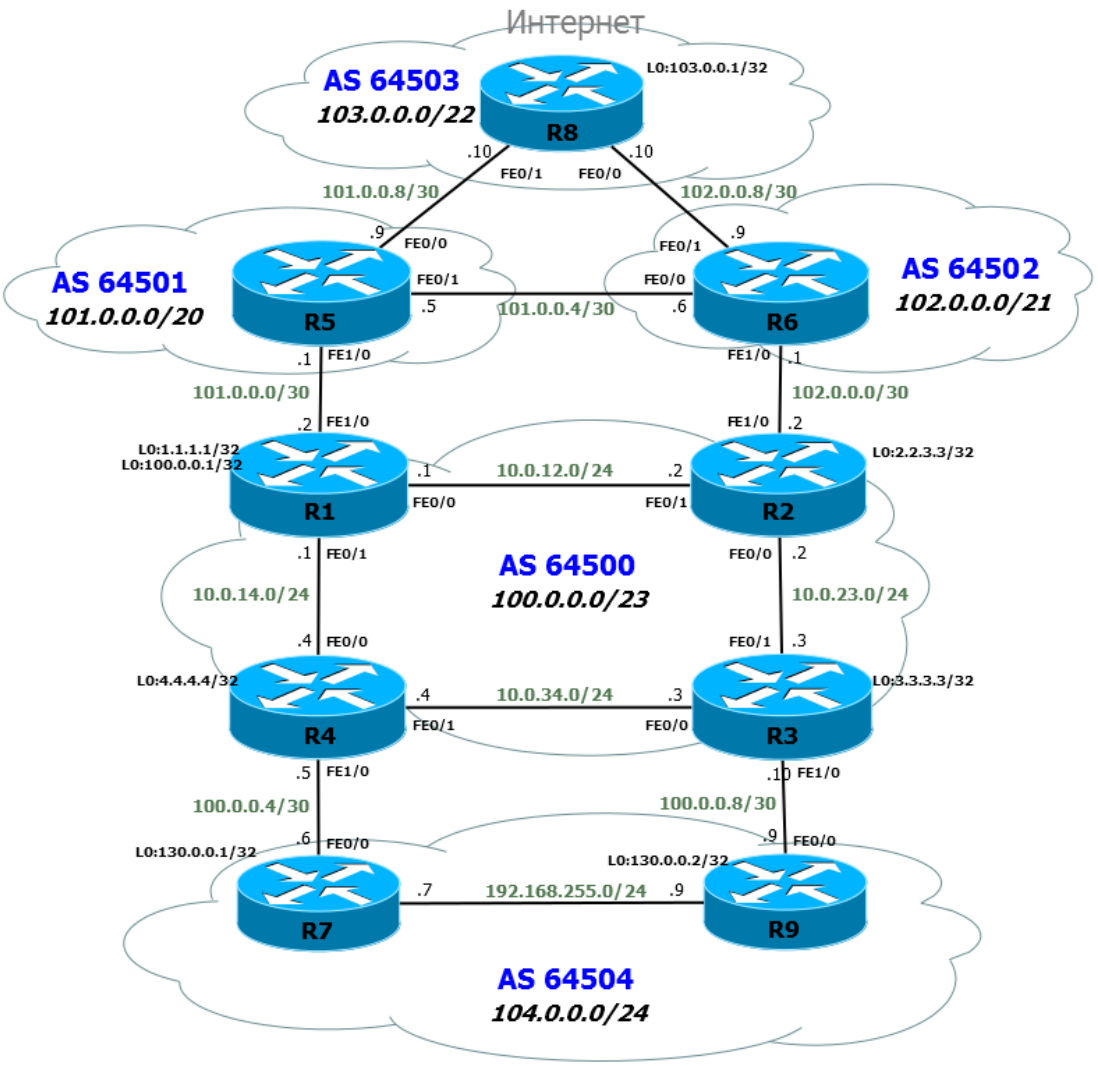


Без использования коммьюнити силами только своей AS сделать это не представляется возможным.
Кстати, как бы это ни было прискорбно, но такие ограничения реально используются в нашей жизни. Распространены ситуации, когда несколько провайдеров устанавливают между собой пиринговые отношения — трафик между их сетями не ходит через вышестоящих провайдеров, не даёт круг через пол-России, но кого-то не пускают — кому-то свои сети не анонсируют.
Интереснейшие статьи об Интернете и BGP и о пиринговых войнах.

Практика Community

Мы же в качестве примера рассмотрим следующую ситуацию.

Основная схема статьи дополняется ещё одним маршрутизатором клиента и двумя линками.



R7 и R9 разнесены территориально — так называемое георезервирование. Главным является его правый сайт, левый — резервный.
Внутри своей AS он без проблем настроил передачу исходящего трафика в нужном месте — через R3. А вот со входящим посложнее — MED не позволяет использовать религия, да и доверия ему нет.

Поэтому мы разработали схему взаимодействия, используя community. На самом деле она будет общая для всех. Например, ниже мы установим правило — если получили маршрут с коммьюнити 64500:150, увеличить Local Preference для него до 150. А потом такую политику применяем к нужным нам маршрутам.

На нашем оборудовании (на всём) мы определяем ip community-list:

```
ip community-list 1 permit 64504:150
```

Задаём правило обработки:

```
route-map LP150 permit 10
 match community 1
 set local-preference 150
```

Это общий блок, который будет одинаков для всех устройств. После этого клиент может сказать, что хочет воспользоваться этой функцией и мы применяем карту на BGP-соседа:
Применяем карту к BGP-соседу на R3:

```
router bgp 64500
 neighbor 100.0.0.10 remote-as 64504
 neighbor 100.0.0.10 route-map LP150 in
```

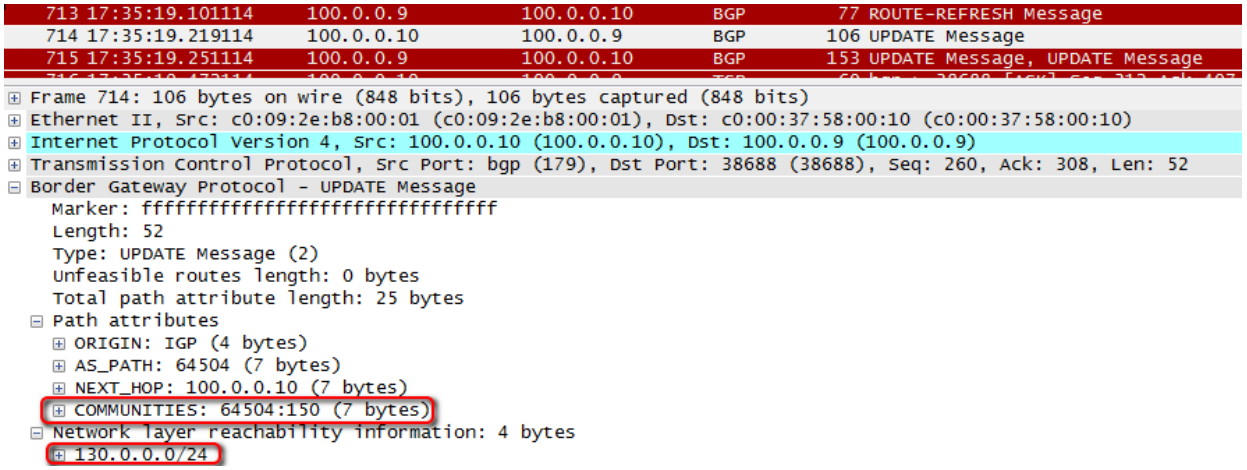
Итак, если в анонсе от соседа 100.0.0.10 community совпадает со значением в условии, установить Local Preference для этих маршрутов в 150.

> Часто такие политики (route-map) применяются по умолчанию на всех внешних соседей. Клиентам остаётся только настроить передачу нужной коммьюнити и даже не нужно просить о чём-то провайдера — всё сработает автоматически.

Это наша политика по использовании коммьюнити. О ней мы сообщаем клиенту, мол, хочешь Установить для своего маршрута Local Preference в 150 в нашей AS, используй community 64500:150
И вот он настраивает на R9:

```
router bgp 64504
 neighbor 100.0.0.9 remote-as 64500
 neighbor 100.0.0.9 route-map LP out
 neighbor 100.0.0.9 send-community

 route-map LP permit 10
 set community 64500:150
```

При необходимости то же самое он может настроить на R7.
После **clear ip bgp * soft** в отправляемых анонсах мы можем увидеть community:



В итоге R3 имеет маршрут с более высоким Local Preference:

```
R3#sh ip bgp
BGP table version is 6, local router ID is 3.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 100.0.0.0/23     0.0.0.0                  0         32768 i
* i                 1.1.1.1                  0    100      0 i
* i                 2.2.2.2                  0    100      0 i
*> 130.0.0.0/24     100.0.0.10               0    150      0 64504 i
```

Передаёт его рут-рефлектору (R1 и R2), который делает выбор и распространяет всем своим соседям:

```
R1#sh ip bgp
BGP table version is 27, local router ID is 100.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i100.0.0.0/23     3.3.3.3                  0    100      0 i
* i                 4.4.4.4                  0    100      0 i
* i                 2.2.2.2                  0    100      0 i
*>                  0.0.0.0                  0         32768 i
*>i130.0.0.0/24     3.3.3.3                  0    150      0 64504 i
* i                 3.3.3.3                  0    150      0 64504 i
```

И даже R4, которому рукой дотянуться до R7, будет отправлять трафик на R3:

```
R4#sh ip bgp 130.0.0.0
BGP routing table entry for 130.0.0.0/24, version 14
Paths: (3 available, best #1, table Default-IP-Routing-Table)
  Advertised to update-groups:
     1
  64504
    3.3.3.3 (metric 11) from 1.1.1.1 (100.0.0.1)
      Origin IGP, metric 0, localpref 150, valid, internal, best
      Originator: 3.3.3.3, Cluster list: 0.0.0.1
  64504
    3.3.3.3 (metric 11) from 2.2.2.2 (2.2.2.2)
      Origin IGP, metric 0, localpref 150, valid, internal
      Originator: 3.3.3.3, Cluster list: 2.2.2.2
  64504
    100.0.0.6 from 100.0.0.6 (130.0.0.1)
      Origin IGP, metric 0, localpref 100, valid, external
      Community: 4227334294
```

Трафик идёт именно тем путём, который мы выбрали.

```
R1#traceroute 130.0.0.2 source 100.0.0.1

Type escape sequence to abort.
Tracing the route to 130.0.0.2

  1 10.0.14.4 296 msec
    10.0.12.2 440 msec
    10.0.14.4 284 msec
  2 10.0.23.3 784 msec
    10.0.34.3 656 msec
    10.0.23.3 796 msec
  3 100.0.0.10 1152 msec 1468 msec 1112 msec
```

> Пусть вас не пугает по 3 записи для каждого хопа — это говорит о балансировке трафика между равноценными линками *R1R2R3* и *R1R4R3*. Просто один раз он идёт по одному пути, второй по другому. А вот вы лучше попробуйте ответить на вопрос, почему на первом хопе 1-я и 3-я попытки идут через R4, а вот на втором хопе на R3. Почему пакет “перепрыгивает”? Как так получается?

Кстати, не стоит забывать команду **ip bgp-community new-format**, а иначе вместо этого:

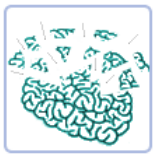
```
R3#sh ip community-list
Community standard list 1
    permit 64504:150
```

вы увидите это:

```
R3#sh ip community-list
Community standard list 1
    permit 4227334294
```

Отправляться будет то же самое, но в выводах show команд будет отображаться в удобном виде.

==========



Задача № 8

В нашей AS для настройки политик с клиентскими AS, используются community. Используются такие значения: 64500:150, 64500:100, 64500:50, 64500:1, 64500:2, 64500:3.
Кроме того, маршрутизаторы нашей AS также используют community для работы с соседними AS. Их формат: 64501:xxx, 64502:xxx.
Задание:
— все значения community приходящие от клиентов, которые не определены политикой, должны удаляться,
— значения community, которые проставлены клиентами, должны удаляться, при передаче префиксов соседним вышестоящим AS. При этом не должны удаляться другие значения, которые проставлены маршрутизаторами нашей AS.

Конфигурация и схема: базовые.

Подробности задачи тут.

==========

Конфигурация устройств

В приведённом примере видно, что коммьюнити позволяет работать не с отдельными анонсами и для каждого из них отдельно применять какие-то политики, а рассматривать их сразу как группу, что естественно, значительно упрощает обслуживание. Иными словами, коммьюнити — это группа анонсов с одинаковыми характеристиками.

При работе с community важно понимать, что настройка необходима с двух сторон — чтобы желаемое заработало, у провайдера тоже должна быть выполнена соответствующая конфигурация.

Часто у провайдеров бывает уже выработанная политика использования коммьюнити, и они просто дают вам те номера, которые необходимо использовать. То есть после того, как вы добавите к анонсу номер коммьюнити, провайдеру не придётся ничего делать руками — всё произойдёт автоматически.

Например, у Балаган-Телекома может быть такая политика:

Значение	Действие
64501:100X	При анонсировании маршрута соседу A добавить X препендов, где X от 1 до 6
64501:101X	При анонсировании маршрута соседу B добавить X препендов, где X от 1 до 6
64501:102X	При анонсировании маршрута соседу C добавить X препендов, где X от 1 до 6
64501:103X	При анонсировании маршрута в AS64503 добавить X препендов, где X от 1 до 6
64501:20050	Установить Local Preference для полученных маршрутов в 50
64501:20150	Установить Local Preference для полученных маршрутов в 150
64501:666	Установить Next-Hop для маршрут в Null — создать Black Hole
64501:3333	выполнить скрипт по уничтожению конфигурации BGP на всех маршрутизаторах AS

Исходя из этой таблички, которая опубликована на сайте Балаган-телекома, мы можем сами принять решение об управлении трафиком.

Как это реально может помочь нам?

У нас Dual-homing подключение к двум различным провайдерам — Балаган Телеком и Филькин Сертификат. У датацентра подключение также к обоим провайдерам. Он принадлежит какому-то контент-генератору, допустим это оператор потового видео.

По умолчанию, в нашу сеть всё ходит через Балаган-Телеком (AS64501). Канал там хоть и широкий, но его утилизация уже достаточно высока. Мы хотим продавать домашним клиентам услугу IPTV и прогнозируем значительный рост входящего трафика. Неплохо было бы его завернуть в Филькин Сертификат и не бояться о том, что основной канал забьётся. При этом, естественно, весь другой трафик переносить не нужно.

В таблице BGP проверяем, где находится сеть 103.0.0.0. Видим, что это AS64503, которая достижима через обоих провайдеров с равным числом AS в AS-Path.

```
R1#sh ip bgp 103.0.0.0
BGP routing table entry for 103.0.0.0/22, version 4
Paths: (2 available, best #2, table Default-IP-Routing-Table)
Flag: 0x820
  Advertised to update-groups:
     3
  64502 64503
    2.2.2.2 (metric 11) from 2.2.2.2 (2.2.2.2)
      Origin IGP, metric 0, localpref 100, valid, internal
  64501 64503
    101.0.0.1 from 101.0.0.1 (5.5.5.5)
      Origin IGP, localpref 100, valid, external, best
```

Вот как видит нас маршрутизатор из AS 64503:

```
R8#sh ip bgp 100.0.0.0
BGP routing table entry for 100.0.0.0/23, version 5
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to update-groups:
     1
  64502 64500
    102.0.0.9 from 102.0.0.9 (6.6.6.6)
      Origin IGP, localpref 100, valid, external
  64501 64500
    101.0.0.9 from 101.0.0.9 (5.5.5.5)
      Origin IGP, localpref 100, valid, external, best
```

Маршрут в Балаган-Телеком выбран предпочтительным

Какие мысли?
Анонсировать определённые сети в Филькин Сертификат, а остальные оставить в Балаган Телеком? Негибко, немасштабируемо.
Вешать препенды на маршруты, отдаваемые в Балаган Телеком? Тогда, скорее всего, куча другого трафика перетечёт на Филькин Сертификат.
Попросить инженера Балаган-Телекома вручную удлинить наши маршруты при передаче их в AS64503. Уже ближе к истине, и это даже сработает, но, скорее всего, инженер провайдера пошлёт вас… на сайт с табличкой, где прописана их политика Community.

Собственно, всё, что нужно сделать нам — на маршрутизаторе R1 применить route-map по добавлению коммьюнити 64500:1031 к соседу R5(напоминаем, что 103X — это для соседа из AS 64503). Дальше всё сделает автоматика.

Вот как R5 видит маршрут сам:

```
R5#sh ip bgp
BGP table version is 7, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 100.0.0.0/23     101.0.0.2                0             0 64500 i
*                   101.0.0.10                             0 64503 64502 64500 i
*                   101.0.0.6                              0 64502 64500 i
*> 101.0.0.0/20     0.0.0.0                  0         32768 i
*  102.0.0.0/21     101.0.0.2                              0 64500 64502 i
*                   101.0.0.10                             0 64503 64502 i
*>                  101.0.0.6                0             0 64502 i
*  103.0.0.0/22     101.0.0.6                              0 64502 64503 i
*>                  101.0.0.10               0             0 64503 i
```

Всё без изменений.

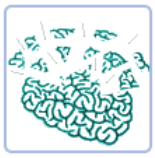
Вот как его видит R8:

```
R8#sh ip bg
BGP table version is 7, local router ID is 103.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 100.0.0.0/23     102.0.0.9                              0 64502 64500 i
*                   101.0.0.9                              0 64501 64501 64500 i
*> 101.0.0.0/20     102.0.0.9                              0 64502 64501 i
*> 102.0.0.0/21     102.0.0.9                0             0 64502 i
*> 103.0.0.0/22     0.0.0.0                  0         32768 i
```

Как видите, галочка стоит напротив более короткого пути через Филькин Сертификат, чего мы и добивались.

```
R8#traceroute 100.0.0.1

Type escape sequence to abort.
Tracing the route to 100.0.0.1

  1 102.0.0.9 [AS 64502] 440 msec 272 msec 344 msec
  2 102.0.0.2 [AS 64502] 832 msec 692 msec 700 msec
  3 10.0.12.1 1048 msec 1008 msec 1292 msec
```

==========



Задача № 9

Одним из наших клиентов стала крупная компания. Платят они нам довольно много, но тут возникла проблема с тем, что когда происходят какие-то проблемы с провайдером AS64501, то качество связи, которую обеспечивает линк с провайдером AS64502, не устраивает клиента. Главное для нашего клиента, хорошее качество связи к филиалам.

Так как клиент солидный, то пришлось установить пиринг с еще одним провайдером AS64513. Но он нам дорого обходится поэтому использовать его мы будем только когда провайдер AS64501 недоступен и только для этого важного клиента.

Задание:

Надо настроить работу сети таким образом, чтобы через провайдера AS64513 сеть клиента 150.0.0.0/24 анонсировалась только в том случае, если через провайдера AS64501 недоступна сеть 103.0.0.0/22 (она используется для проверки работы провайдера). Кроме того, от провайдера AS64513 нам надо принимать только сети филиалов клиента (50.1.1.0/24, 50.1.2.0/24, 50.1.3.0/24) и использовать их только если они недоступны через провайдера AS64501. Остальной трафик клиента будет ходить через AS64502.

Конфигурация: базовая.

Подробности задачи тут.

==========

Материалы выпуска

Повесть о настоящем Интернете

BGP Blackhole — эффективное средство борьбы с DDoS

Сравнение функций и мест использования EBGP и IBGP

Основы BGP

Конфигурация устройств: базовый IBGP, Route Reflectors, Community.

Послесловие

Вот на этом знакомство с BGP можно считать законченным. Теперь мы вернёмся к нему ни много ни мало при рассмотрении MPLS L3VPN.

Материал подготовил для вас eucariot.
За траблшутинг статьи спасибо JDIMA
Задачки предоставлены Наташей — автором лучшего викисайта по сетевым протоколам и технологиям — xgu.ru.

Как обычно минутка саморекламы: вы можете найти все статьи цикла на нашем сайте linkmeup.ru. Там же все выпуски первого подкаста для связистов linkmeup.

Теги: сети для самых маленьких, BGP, IBGP, RR

Хабы: Я пиарюсь

Марат @eucariot
Пользователь

Комментарии 17

sinist3r 15 октября 2013 в 12:21 +1

Как всегда глубоко, доступно и актуально.
Но естественно для полного понимания не помешает определенный бэкграунд.

Про пиринговые войны так же можно почитать в блоге Кипчатова

tomoto 15 октября 2013 в 14:08 0

ну что тут скажешь, супер…
~~наверное в visio~~ чем такие классные схемки рисуются и как анимацию накладываете? :)

eucariot 15 октября 2013 в 14:27 0

Dia. Весьма удобная и бесплатная)

Анимацию делал вручную в gimp покадрово.

rdntw 15 октября 2013 в 15:03 0

в первой задачи из-за недостатка eBGP multihop?

rdntw 15 октября 2013 в 15:10 0

во второй
из-за того что у EBGP-маршрута AD=20, а у OSPF 110, верно?

Nattali 15 октября 2013 в 15:46 0

Да. И суть задачи в том, чтобы не задевая все остальные маршруты (OSPF и BGP), исправить ситуацию так, чтобы трафик ходил через прямой линк, через OSPF.
(только не пишите, пожалуйста, ответ сюда. Пусть другие тоже подумают :))

PTPtupit 15 октября 2013 в 15:13 0

show ip route

Nattali 15 октября 2013 в 15:49 0

В этой задаче речь о iBGP. А эта команда используется для eBGP

PTPtupit 15 октября 2013 в 15:19 0

Статья неплохая… реквестую помимо картинок прикреплять pcap-файлы.

eucariot 15 октября 2013 в 16:26 0

Да тут в общем-то там ничего интересного. Но на заметку возьму себе.

MetallicAt 15 октября 2013 в 22:21 0

Очень добротно. Название блога перестаёт соответствовать наполнению :)

eucariot 16 октября 2013 в 05:17 0

Ирония обострится, когда под этим названием выйдет MPLS L3VPN на основе MP-BGP (или Inter-AS VPN):)

netvolart 15 октября 2013 в 23:05 +1

Очень классно! А MPLS для самых маленьких будет? :)

eucariot 16 октября 2013 в 05:14 0

Да, обязательно. Причём начиная с базового MPLS и заканчивая MPLS L2VPN и, возможно, даже TE. Но зарекаться не буду. — ещё несколько выпусков до этого.

gissarsky 16 октября 2013 в 16:35 0

Ждем. Отличные статьи, ссылаюсь на них, когда даю что-то студентам.

eucariot 16 октября 2013 в 17:20 0

Спасибо, Эмиль! Очень приятно слышать. Впереди ещё мультикаст до MPLS.

gissarsky 17 октября 2013 в 07:22 0

Здорово! Как-раз то, что надо.

САМОЕ ЧИТАЕМОЕ

Сутки | Неделя | Месяц

Как платить программистам

+10 13,8k 29 25

Не говорите «I feel myself», и другие правила английского языка, которые вгоняют в ступор

+32 18,8k 148 58

Инсайды от сотрудника Facebook: как попасть на стажировку, получить оффер и все о работе в компании

+34 14,2k 74 114

Умные пайетки

+45 8,4k 24 56

Визуальные и приборные правила полетов

+58 7,6k 31 26

/