

# Практические примеры BGP

## Содержание

[Введение](#)

[Предварительные условия](#)

[Требования](#)

[Используемые компоненты](#)

[Условные обозначения](#)

[Практический пример использования BGP №1](#)

[Как работает BGP?](#)

[eBGP и iBGP](#)

[Включение маршрутизации BGP](#)

[Формирование соседей BGP](#)

[BGP и интерфейсы обратной связи](#)

[eBGP с несколькими переходами](#)

[eBGP с несколькими переходами \(распределение нагрузки\)](#)

[Карты маршрутов](#)

[команды настройки match и set](#)

[команда network](#)

[Перераспределение](#)

[Статические маршруты и перераспределение](#)

[iBGP](#)

[Алгоритм принятия решений в BGP](#)

[Примеры практического применения BGP 2](#)

[Атрибут AS\\_PATH](#)

[Атрибут "источник"](#)

[Атрибут BGP "следующий переход"](#)

[Запасной путь BGP](#)

[Synchronization](#)

[Атрибут "вес"](#)

[Атрибут "локальное предпочтение"](#)

[Атрибут "метрика"](#)

[Атрибут "сообщество"](#)

[Примеры практического применения BGP 3](#)

[Фильтрация BGP](#)

[Регулярные выражения AS](#)

[Соседи BGP и карты маршрутов](#)

[Примеры практического применения BGP 4](#)

[CIDR и агрегированные адреса](#)

[Конфедерация BGP](#)

[Рефлекторы маршрутов](#)

[Отключение переключающихся маршрутов](#)

[Выбор пути BGP](#)

[Примеры практического применения BGP 5](#)

[Практический пример построения](#)

[Дополнительные сведения](#)

## **Введение**

В данном документе приведено пять практических примеров использования протокола граничных шлюзов (BGP).

## **Предварительные условия**

### **Требования**

Для этого документа отсутствуют особые требования.

### **Используемые компоненты**

Настоящий документ не имеет жесткой привязки к каким-либо конкретным версиям программного обеспечения и оборудования.

### **Условные обозначения**

[Дополнительные сведения об условных обозначениях см. в документе Условные обозначения технических терминов Cisco.](#)

## **Практический пример использования BGP №1**

[Протокол BGP, определенный в RFC 1771, позволяет организовывать междоменную маршрутизацию без петель между автономными системами \(AS\).](#) AS – это совокупность маршрутизаторов под управлением единой службы технического администрирования. Для обмена информацией о маршрутах в пределах AS в маршрутизаторах могут использоваться различные внутренние протоколы шлюзов (IGP). Для маршрутизации пакетов за пределы AS маршрутизаторы могут использовать внешний протокол маршрутизации.

### **Как работает BGP?**

В качестве протокола транспортного уровня BGP использует TCP, порт 179. Два маршрутизатора BGP создают соединение TCP друг с другом. Эти маршрутизаторы являются равноправными. Равноправные маршрутизаторы обмениваются сообщениями для инициирования и подтверждения параметров соединения.

Маршрутизаторы BGP обмениваются информацией о доступности сетей. Эта информация главным образом состоит из указаний на полные пути, по которым должен пройти маршрут для достижения сети назначения. Пути представляют собой номер автономных систем BGP. Эта информация помогает в построении графа автономных систем, который не содержит петель. Граф также показывает, где нужно применить политики маршрутизации, чтобы наложить на поведение маршрутизаторов некоторые ограничения.

Любые два маршрутизатора, организующие TCP-соединение для обмена информацией о маршрутизации BGP являются "равноправными" или "соседями". Сначала равноправные узлы BGP обмениваются полными таблицами маршрутизации BGP. После этого обмена при изменении таблиц маршрутизации равноправные узлы рассылают обновления с

изменениями. BGP хранит номер версии таблицы BGP. Номер версии совпадает у всех одноранговых узлов BGP. Номер версии изменяется всякий раз, когда BGP обновляет таблицу при изменении информации маршрутизации. Передача пакетов поддержания сообщения позволяет убедиться в том, что соединение между равноправными узлами BGP действует. При возникновении или особых условий рассылаются пакеты уведомлений.

## eBGP и iBGP

Если в AS несколько адресантов BGP, AS может выполнять роль транзитной службы для других автономных систем. Как показано на схеме в данном разделе, AS200 является транзитной AS для AS100 и AS300.

Для передачи информации во внешние AS доступность сетей должна быть гарантирована. Чтобы гарантировать доступность сетей выполняются следующие процессы:

Обмен информацией внутренними равноправными узлами BGP (iBGP) между маршрутизаторами в системе AS

Перераспределение информации BGP в IGP, работающим в AS

Когда BGP работает между маршрутизаторами, принадлежащими двум разным AS, это называется внешним BGP (eBGP). Когда BGP работает между маршрутизаторами, принадлежащими одной AS, это называется iBGP.

## Включение маршрутизации BGP

Чтобы включить и настроить BGP, необходимо выполнить описанные ниже действия.

Предположим, что нам нужно два маршрутизатора, RTA и RTB, общающихся при помощи BGP. В первом примере RTA и RTB находятся в различных AS. Во втором примере оба маршрутизатора принадлежат одной AS.

Определите процесс маршрутизатора и номер AS, которой принадлежит маршрутизатор.

Для включения BGP в маршрутизаторе введите следующую команду:

```
router bgp autonomous-system RTA# router bgp 100 RTB# router bgp 200
```

Эти операторы указывают на то, что в RTA работает BGP и он принадлежит AS100. В RTB работает BGP и он принадлежит AS200.

Определение соседей BGP.

Формирование соседей BGP указывает маршрутизаторы, которые пытаются обмениваться информацией при помощи BGP. [Данный процесс описывается в разделе Формирование соседей BGP.](#)

## Формирование соседей BGP

Два маршрутизатора BGP становятся соседями после того, как маршрутизаторы установят TCP-соединение друг с другом. TCP-соединение необходимо для того, чтобы два равноправных маршрутизатора начали обмениваться информацией о маршрутах.

После организации TCP-соединения маршрутизаторы передают сообщения инициирования для обмена значениями. Значения, которыми обмениваются маршрутизаторы, содержат номер AS, версию BGP, которая работает в маршрутизаторе, идентификатор BGP маршрутизатора и интервал обмена сообщениями поддержания соединения. После подтверждения и принятия этих значений происходит установление соединения между соседями. , Established , BGP.

**Для установления TCP-соединения введите команду neighbor:**

```
neighbor ip-address remote-as number
```

**Номер** в команде является количеством AS маршрутизатора, с которым вы хотите соединиться с BGP. **IP-адрес** является адресом следующего узла с прямым подключением для eBGP. Для iBGP **IP-адресом** является любой IP-адрес на другом маршрутизаторе.

**Два IP-адреса, используемых в командах neighbor для равноправных маршрутизаторов, должны обладать возможностью доступа друг к другу.** Одним способом проверки доступности является использование расширенной команды ping между двумя IP-адресами. **Для расширенной команды ping маршрутизатор, на котором она запущена использует в качестве адреса источника IP-адрес, задаваемый командой neighbor.** Маршрутизатор должен использовать этот адрес, а не IP-адрес интерфейса, с которого отправляется пакет.

*При наличии каких-либо изменений конфигурации BGP необходимо сбросить соединение с соседом, чтобы новые параметры вступили в силу.*

```
clear ip bgp address
```

**Примечание:** *Адрес* является адресом соседа.

```
clear ip bgp *
```

Данная команда удаляет соединения со всеми соседями.

По умолчанию сессии BGP начинаются с использования 4 версии BGP и при необходимости выполняется согласование с предыдущими версиями. Можно отменить согласование и явно указать версию BGP, которую маршрутизаторы будут использовать для соединений с соседями. В режиме настройки маршрутизатора введите следующую команду:

```
neighbor {ip address | peer-group-name} version value
```

**Здесь приведен пример настройки с использованием команды neighbor:**

```
RTA#
router bgp 100
neighbor 129.213.1.1 remote-as 200
```

```
RTB#
router bgp 200
```

```
neighbor 129.213.1.2 remote-as 100
neighbor 175.220.1.2 remote-as 200
```

RTC#

```
router bgp 200
neighbor 175.220.212.1 remote-as 200
```

В данном примере в RTA и в RTB работает eBGP. В RTB и RTC работает iBGP. Номер удаленной AS не указывает либо на внешнюю, либо на внутреннюю AS, что указывает либо на eBGP, либо на iBGP. Кроме того, между равноправными узлами eBGP прямое соединение, а между равноправными узлами iBGP прямого соединения нет. маршрутизаторам iBGP прямое соединение не требуется. Однако должен быть одинаковый IGP, который позволит двум соседям обмениваться информацией друг с другом.

[В данном разделе приведен пример информации, отображаемой командой show ip bgp neighbors.](#)

**Примечание:** Обратите особое внимание состояние BGP. , Established, , .

**Примечание:** Кроме того, обратите внимание на следующее:

```
BGP version 4
```

```
remote router ID
```

Этот номер является самым большим значением IP-адреса маршрутизатора или самым большим адресом обратной связи интерфейса, если он есть.

```
table version
```

table version . Каждый раз при поступлении новой информации версия таблицы увеличивается. Постоянное увеличение версии указывает на наличие постоянного переключения маршрута, который приводит к постоянному обновлению маршрутов.

```
# show ip bgp neighbors BGP neighbor is 129.213.1.1, remote AS 200, external link BGP version 4,
remote router ID 175.220.12.1 BGP state = Established, table version = 3, up for 0:10:59 Last
read 0:00:29, hold time is 180, keepalive interval is 60 seconds Minimum time between
advertisement runs is 30 seconds Received 2828 messages, 0 notifications, 0 in queue Sent 2826
messages, 0 notifications, 0 in queue Connections established 11; dropped 10
```

## [BGP и интерфейсы обратной связи](#)

Практика использования интерфейса обратной связи для определения соседей распространена в iBGP, но не распространена в eBGP. Обычно интерфейс обратной связи используется для того, чтобы убедиться, что IP-адрес соседей доступен и не зависит от правильно работающего аппаратного обеспечения. В случае с eBGP между равноправными маршрутизаторами есть прямое соединение и обратная связь не применяется.

При использовании IP-адреса интерфейса обратной связи в команде neighbor требуется дополнительная настройка соседнего маршрутизатора. Соседний маршрутизатор должен сообщить BGP о том, что для инициирования TCP-соединения с соседом BGP он использует интерфейс обратной связи, а не физический интерфейс. Для указания на интерфейс обратной связи введите следующую команду:

```
neighbor ip-address update-source interface
```

В данном примере показано использование этой команды:

```
RTA#
router bgp 100
neighbor 190.225.11.1 remote-as 100
neighbor 190.225.11.1 update-source loopback 1
```

```
RTB#
router bgp 100
neighbor 150.212.1.1 remote-as 100
```

В данном примере в RTA и в RTB работает iBGP в AS100. В команде neighbor RTB использует интерфейс обратной связи RTA, 150.212.1.1. В данном случае RTA должен явно указать BGP на использование IP-адреса интерфейса обратной связи в качестве адреса источника в TCP-соединении с соседом. Чтобы сделать это, RTA добавляет update-source тип интерфейса номер интерфейса, чтобы команда выглядела так: neighbor 190.225.11.1 update-source loopback 1. Данный оператор явно указывает BGP на использование IP-адреса интерфейса обратной связи, когда BGP обменивается информацией с соседом 190.225.11.1.

**Примечание:** RTA использовал IP-адрес физического интерфейса RTB, 190.225.11.1, в качестве соседа. Использование этого IP-адреса позволяет не выполнять дополнительную настройку RTB. [Полный пример конфигурации для ситуации с сетью см. в статье Пример конфигурации для iBGP и eBGP с адресом обратной связи или без него.](#)

## [eBGP с несколькими переходами](#)

В некоторых случаях маршрутизаторы Cisco могут применять eBGP с маршрутизаторами других производителей, которые не поддерживают прямые соединения между двумя равноправными узлами. Для установления соединения можно использовать eBGP с несколькими переходами. EBGP с несколькими переходами позволяет устанавливать соединение между двумя равноправными узлами, между которыми нет прямого соединения. Функция поддержки нескольких переходов предназначена только для eBGP, но не для iBGP. В данном примере демонстрируется использование eBGP с несколькими переходами:

```
RTA#
router bgp 100
neighbor 180.225.11.1 remote-as 300
neighbor 180.225.11.1 ebgp-multihop
```

```
RTB#
router bgp 300
neighbor 129.213.1.2 remote-as 100
```

RTA указывает на внешнего соседа, с которым отсутствует прямое подключение. RTA должен указать на свое использование [команды neighbor ebgp-multihop](#). С другой стороны, RTB указывает на соседа, у которого есть прямое соединение, 129.213.1.2. Из-за этого прямого подключения RTB не нужна команда neighbor ebgp-multihop. Чтобы позволить соседям достигать друг друга без соединения необходимо также настроить IGP или статическую маршрутизацию.

[В примере раздела eBGP с несколькими переходами \(распределение нагрузки\) показано, как добиться распределения нагрузки при помощи BGP при использовании eBGP в параллельных подключениях.](#)

## eBGP с несколькими переходами (распределение нагрузки)

```
RTA#
int loopback 0
ip address 150.10.1.1 255.255.255.0
router bgp 100
neighbor 160.10.1.1 remote-as 200
neighbor 160.10.1.1 ebgp-multihop
neighbor 160.10.1.1 update-source loopback 0
network 150.10.0.0
```

```
ip route 160.10.0.0 255.255.0.0 1.1.1.2
ip route 160.10.0.0 255.255.0.0 2.2.2.2
```

```
RTB#
int loopback 0
ip address 160.10.1.1 255.255.255.0
router bgp 200
neighbor 150.10.1.1 remote-as 100
neighbor 150.10.1.1 update-source loopback 0
neighbor 150.10.1.1 ebgp-multihop
network 160.10.0.0
```

```
ip route 150.10.0.0 255.255.0.0 1.1.1.1
ip route 150.10.0.0 255.255.0.0 2.2.2.1
```

**В данном примере показано использование интерфейсов обратной связи, update-source и ebgp-multihop.** В примере описан обходной путь, позволяющий организовать распределение нагрузки между адресантами eBGP через параллельные последовательные соединения. В обычных ситуациях BGP пользуется для пересылки пакетов одним из соединений и распределение нагрузки не выполняется. При использовании интерфейсов обратной связи следующим переходом для eBGP является интерфейс обратной связи. Для добавления двух путей до точки назначения, обладающих одинаковой стоимостью, используются статические маршруты или IGP. У RTA есть выбор для следующего перехода к 160.10.1.1: один путь через 1.1.1.2, а другой путь через 2.2.2.2. У RTB есть такой же выбор.

## Карты маршрутов

В BGP карты маршрутов используются очень интенсивно. В контексте BGP карта маршрутов представляет собой способ контроля и изменения информации о маршрутах. Контроль и изменение информации о маршрутах выполняется при помощи определения условий перераспределения маршрутов от одного протокола маршрутизации другому. Контроль информации о маршрутах также может выполняться при включении в BGP и исключении из него. Формат карты маршрутов приведен ниже:

```
route-map map-tag [[permit | deny] | [sequence-number]]
```

Метка карты представляет собой простое имя, присваиваемое карте маршрутов. Можно определять несколько экземпляров одной и той же карты или одной метки имени. Порядковый номер просто указывает на положение, которое новая карта маршрутов будет занимать в списке карт маршрутов, уже настроенных с этим именем.

В данном примере определено два экземпляра карт маршрутов с именем MYMAP. У первого экземпляра порядковый номер 10, а у второго порядковый номер 20.

`route-map MYMAP permit 10` (Здесь идет первый набор условий.)

`route-map MYMAP permit 20` (Здесь идет второй набор условий.)

При применении карты маршрутов MYMAP к входящим или исходящим маршрутам первый набор условий применяется через экземпляр 10. Если первый набор условий не удовлетворяется, происходит переход к экземпляру карты маршрутов с более высоким номером.

### [команды настройки match и set](#)

Все карты маршрутов состоят из списка команд настройки `match` и `set`. Командой `match` задаются критерии соответствия, а командой `set` задаются действия, выполняемые при удовлетворении условий, задаваемых командой `match`.

Например, можно определить карту маршрутов, которая проверяет исходящие обновления. При наличии соответствия для IP-адреса 1.1.1.1, метрике этого обновления присваивается значение 5. Пример иллюстрируют следующие команды:

```
match ip address 1.1.1.1 set metric 5
```

В данном случае при удовлетворении критериям соответствия и наличии `permit` (разрешение), происходит перераспределение или контроль маршрутов, указанные заданным набором действий. После происходит выход из списка.

При удовлетворении критериям соответствия и наличии `deny` (запрет), перераспределение или контроль маршрутов не выполняются. После происходит выход из списка.

Если критерии соответствия не удовлетворяются и указано `permit or deny`, выполняется проверка следующего экземпляра карты маршрутов. Например, проверяется экземпляр 20. Данная проверка следующих экземпляров выполняется до ее прерывания или завершения со всеми экземплярами карты маршрутов. При завершении списка без соответствия маршрут не принимается и не пересылается.

*В ПО Cisco IOS® версий, предшествующих 11.2, при использовании карт маршрутов для фильтрации обновлений BGP, а не для перераспределения между протоколами, при использовании команды `match` для IP-адреса нельзя фильтровать входящие маршруты. Фильтр исходящих маршрутов разрешен. В ПО Cisco IOS начиная с версии 11.2 данное ограничение отсутствует.*

Команды, соответствующие команде `match`:

`match as-path`

`match community`

`match clns`



**match interface**

**match ip address**

**match ip next-hop**

**match ip route-source**

**match metric**

**match route-type**

**match tag**

**Команды, соответствующие команде set:**

**set as-path**

**set clns**

**set automatic-tag**

**set community**

**set interface**

**set default interface**

**set ip default next-hop**

**set level**

**set local-preference**

**set metric**

**set metric-type**

**set next-hop**

**set origin**

**set tag**

**set weight**

Вот несколько примеров карт маршрутов:

### [Пример 1](#)

Предположим, что RTA и RTB используют протокол информации о маршрутах (RIP) и RTA и RTC используют BGP. RTA получает обновления с использованием BGP и передает обновления в RIP. Предположим, что RTA хочет передать RTB маршруты 170.10.0.0 с метрикой 2 и все остальные маршруты с метрикой 5. В данном случае можно использовать следующую конфигурацию:

```
RTA#
router rip
network 3.0.0.0
network 2.0.0.0
network 150.10.0.0
passive-interface Serial0
redistribute bgp 100 route-map SETMETRIC

router bgp 100
neighbor 2.2.2.3 remote-as 300
network 150.10.0.0

route-map SETMETRIC permit 10
match ip-address 1
set metric 2

route-map SETMETRIC permit 20
set metric 5

access-list 1 permit 170.10.0.0 0.0.255.255
```

В данном примере, если маршрут соответствует IP-адресу 170.10.0.0, метрика маршрута равна 2. После происходит выход из списка карт маршрутов. При отсутствии соответствия происходит переход дальше по списку карт маршрутов, что означает использование для любого другого маршрута метрики 5.

**Примечание:** Всегда следует задавать вопрос: «что произойдет с маршрутами, которые не соответствуют ни одному из операторов?» По умолчанию эти маршруты отбрасываются.

### [Пример 2](#)

[Предположим, что в Примере 1 не нужно, чтобы AS100 принимала обновления о 170.10.0.0.](#) Когда за основу берется проверка соответствия IP-адреса, нельзя применять карту маршрутов к входящим маршрутам. Поэтому необходимо использовать карту исходящих маршрутов в RTC:

RTC#

```
router bgp 300
network 170.10.0.0
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-map STOPUPDATES out
```

```
route-map STOPUPDATES permit 10
match ip address 1
```

```
access-list 1 deny 170.10.0.0 0.0.255.255
access-list 1 permit 0.0.0.0 255.255.255.255
```

Теперь, зная, как запустить BGP и определить соседа, необходимо научиться запускать обмен информации о сетях.

Есть несколько способов передачи информации о сетях при помощи BGP. Способы описаны по порядку в следующих разделах:

[команда network](#)

[Перераспределение](#)

[Статические маршруты и перераспределение](#)

[команда network](#)

Команда `network` имеет следующий формат:

```
network network-number [mask network-mask]
```

Команда `network` контролирует сети, исходящие из данного устройства. Данный подход отличается от уже знакомой настройки при помощи внутреннего протокола маршрутизации шлюзов (IGRP) и RIP. С использованием данной команды не делается попыток запуска BGP на определенном интерфейсе. Вместо этого делается попытка указать BGP на то, какие сети протокол BGP должен начинать из данного устройства. В команде используется маска, поскольку BGP версии 4 (BGP4) может обрабатывать подсети и суперсети. **Допускается не более 200 записей команды `network`.**

Команда `network` работает в случае, если маршрутизатор знает сеть, попытка объявления которой предпринимается, будь то подключенная, статическая или динамически изучаемая сеть.

Пример использования команды `network`:

```
RTA#
router bgp 1
network 192.213.0.0 mask 255.255.0.0
ip route 192.213.0.0 255.255.0.0 null 0
```

В примере указывается на то, что маршрутизатор A создает запись сети для 192.213.0.0/16. /16 обозначает то, что используется адрес суперсети класса C и выполняется объявление первых двух октетов или первых 16 битов.

**Примечание:** Чтобы маршрутизатор создал 192.213.0.0, необходим статический маршрут, поскольку статический маршрут помещает соответствующую запись в таблицу маршрутизации.

## Перераспределение

Одним из способов объявления сетей через BGP является команда `network`. Другим способом является перераспределение IGP через BGP. Этим IGP может быть IGRP, протокол первоочередного открытия кратчайших маршрутов (OSPF), протокол улучшенной внутренней маршрутизации шлюзов (EIGRP) или другой протокол. Это перераспределение может показаться непривлекательным, поскольку теперь все внутренние маршруты попадают в BGP; некоторые из этих маршрутов уже могли быть изучены через BGP и в их повторной передаче нет необходимости. Необходимо проводить тщательный отбор, при котором будут переданы только предназначенные для объявления маршруты Интернет, а не все имеющиеся маршруты. Например:

RTA объявляет 129.213.1.0, а RTC объявляет 175.220.0.0. Вот конфигурация RTC:

При вводе команды `network` получаем следующее:

```
RTC#
router eigrp 10
network 175.220.0.0
redistribute bgp 200
default-metric 1000 100 250 100 1500

router bgp 200
neighbor 1.1.1.1 remote-as 300
network 175.220.0.0 mask 255.255.0.0
!--- This limits the networks that your AS originates to 175.220.0.0.
```

Если вместо этого использовать перераспределение, то получаем следующее:

```
RTC#
router eigrp 10
network 175.220.0.0
redistribute bgp 200
default-metric 1000 100 250 100 1500

router bgp 200
neighbor 1.1.1.1 remote-as 300
redistribute eigrp 10
!--- EIGRP injects 129.213.1.0 again into BGP.
```

Это перераспределение приводит к тому, что AS становится источником 129.213.1.0. Мы не являемся источником 129.213.1.0, источником является AS100. Поэтому необходимо использовать фильтры, которые не выпустят источник из этой сети в нашу AS. Правильной является следующая конфигурация:

```
RTC#
router eigrp 10
network 175.220.0.0
redistribute bgp 200
default-metric 1000 100 250 100 1500

router bgp 200
```

```
neighbor 1.1.1.1 remote-as 300
neighbor 1.1.1.1 distribute-list 1 out
redistribute eigrp 10

access-list 1 permit 175.220.0.0 0.0.255.255
```

**Для управления сетями, источником которых является AS200, служит команда access-list.**

Перераспределение OSPF в BGP немного отличается от перераспределения других IGP. Простое использование `redistribute ospf 1` в `router bgp` не работает. Для перераспределения соответствующих маршрутов требуются специальные ключевые слова, например, `internal`, `external` и `nssa-external`. [Дополнительные сведения см. в документе Общие сведения о перераспределении маршрутов OSPF в BGP.](#)

## Статические маршруты и перераспределение

Чтобы указать источник сети или подсети всегда можно использовать статические маршруты. Единственное отличие заключается в том, что BGP считает, что источник этих маршрутов является неполным или неизвестен. [Такого же результата, как и в примере раздела Перераспределение, можно достичь следующим образом:](#)

```
RTC#
router eigrp 10
network 175.220.0.0
redistribute bgp 200
default-metric 1000 100 250 100 1500

router bgp 200
neighbor 1.1.1.1 remote-as 300
redistribute static
...
ip route 175.220.0.0 255.255.255.0 null0
....
```

**Интерфейс null0 означает игнорирование пакета.** Поэтому при получении пакета и наличии более точного соответствия чем `175.220.0.0`, маршрутизатор пересылает пакет через это специальное соответствие. В противном случае маршрутизатор игнорирует пакет. Этот способ прекрасно подходит для объявления суперсети.

В данном документе обсуждались другие способы указания источников маршрутов, находящихся за пределами нашей AS. Следует помнить, что эти маршруты создаются дополнительно к другим маршрутам BGP, которые были изучены BGP через внутренних или внешних соседей. BGP передает другим равноправным узлам информацию, полученную от одного равноправного узла. **Отличие заключается в том, что маршруты, создаваемые с использованием команды `network`, перераспределения или статически указывают вашу AS в качестве источника этих сетей.**

Перераспределение всегда является способом внедрения BGP в IGP.

Например:

```
RTA#
router bgp 100
neighbor 150.10.20.2 remote-as 300
network 150.10.0.0
```

```
RTB#
router bgp 200
neighbor 160.10.20.2 remote-as 300
network 160.10.0.0
```

```
RTC#
router bgp 300
neighbor 150.10.20.1 remote-as 100
neighbor 160.10.20.1 remote-as 200
network 170.10.0.0
```

**Примечание:** Сеть 150.10.0.0 или 160.10.0.0 не нужны в RTC за исключением случаев, когда RTC должен создать эти сети и передать их по мере прихода из AS100 и AS200. Кроме того, различие заключается в том, что команда `network` добавляет дополнительное объявление для этих же сетей, что указывает на то, что AS300 также является источником этих маршрутов.

**Примечание:** Следует помнить, что BGP не принимает обновления, приходящие из собственной AS. Отказ от этой информации обеспечивает междоменную топологию без петель.

Например, предположим, что у AS200 из примера данного раздела есть прямое соединение BGP с AS100. RTA создает маршрут 150.10.0.0 и передает маршрут в AS300. После этого RTC передает данный маршрут в AS200 и хранит AS100 в виде источника. RTB передает 150.10.0.0 в AS100 с AS100 в качестве источника. RTA отмечает, что обновление пришло из собственной AS и игнорирует обновление.

## iBGP

iBGP используется в случаях, когда AS должна выступать в качестве транзитной к другим AS. Верно ли, что это можно сделать также при помощи обучения с использованием eBGP, перераспределения в IGP и повторного перераспределения в другие AS? Да, однако iBGP более гибкий и обеспечивает более эффективные способы обмена информацией в пределах AS. Например, iBGP обеспечивает способы управления лучшими точками выхода из AS с использованием локальных предпочтений. [В разделе Атрибут локального предпочтения приведены дополнительные сведения о локальном предпочтении.](#)

```
RTA#
router bgp 100
neighbor 190.10.50.1 remote-as 100
neighbor 170.10.20.2 remote-as 300
network 150.10.0.0
```

```
RTB#
router bgp 100
neighbor 150.10.30.1 remote-as 100
neighbor 175.10.40.1 remote-as 400
network 190.10.50.0
```

```
RTC#
router bgp 400
neighbor 175.10.40.2 remote-as 100
network 175.10.0.0
```

**Примечание:** Следует помнить, что когда адресант BGP получает обновление от других

адресантов BGP в собственной AS (iBGP), он не передает эту информацию другим адресантам BGP в собственной AS. Адресант BGP, получающий обновление, перераспределяет информацию другим адресантам BGP, находящимися за пределами его AS. При этом между адресантами iBGP в пределах AS поддерживается замкнутая сеть.

На схеме данного раздела в RTA и в RTB работает iBGP. В RTA и RTD также работает iBGP. Обновления BGP, поступающие из RTB в RTA передаются в RTE, находящийся за пределами AS. Обновления не передаются в RTD, который находится в AS. Поэтому чтобы прервать поток обновлений, следует организовать между RTB и RTD взаимодействие iBGP.

## Алгоритм принятия решений в BGP

Когда BGP получает обновления о различных адресах назначения из различных автономных систем, протокол должен выбирать пути для достижения определенных адресов назначения. Для достижения определенного адреса назначения BGP выбирает только один путь.

**BGP основывается на различных атрибутах, как, например, следующий переход, административный вес, локальное предпочтение, источник адреса, длина пути, код источника, метрика и на других атрибутах.**

BGP всегда распространяет наилучший путь до соседей. [Дополнительные сведения см. в "Алгоритм выбора наилучшего пути BGP"](#).

[Эти атрибуты и их использование описываются в разделе Практический пример использования BGP №2.](#)

## Примеры практического применения BGP 2

### Атрибут AS\_PATH

При каждом прохождении обновления маршрута через AS к этому обновлению добавляется номер AS. Фактически атрибут AS\_PATH представляет собой список номеров AS, через которые прошел маршрут, чтобы достичь адреса назначения. AS\_SET представляет собой упорядоченное математическое множество {} всех пройденных AS. [В разделе Пример CIDR №2 \(as-set\) данного документа приведен пример AS\\_SET.](#)

В примере данного раздела RTB объявляет сеть 190.10.0.0 в AS200. Когда этот маршрут проходит через AS300, RTC добавляет в сеть номер собственной AS. Поэтому когда 190.10.0.0 доходит до RTA, к сети присоединено два номера AS: сначала 200, а затем 300. Для RTA путь, по которому можно достичь 190.10.0.0, равен (300, 200).

Та же процедура применяется к 170.10.0.0 и 180.10.0.0. RTB должен принять путь (300, 100); чтобы достичь 170.10.0.0, RTB проходит AS300, а затем AS100. Чтобы достичь 190.10.0.0, RTC должен пройти путь (200), а чтобы достичь 170.10.0.0, необходимо пройти путь (100).

### Атрибут "источник"

Источник является обязательным атрибутом, определяющим источник информации о пути. Атрибут «источник» может принимать три значения:

IGP информация о доступности уровней сети (NLRI) является внутренней по отношению

к AS происхождения. Обычно это происходит при вводе команды `bgp network`. I BGP IGP.

EGPNLRI изучается через внешний протокол шлюза (EGP). E BGP EGP.

INCOMPLETENLRI отсутствует или получается другими способами. INCOMPLETE обычно возникает при перераспределении маршрутов из других протоколов маршрутизации в BGP, и когда источник маршрута является неполным. ? BGP INCOMPLETE.

```
RTA#
router bgp 100
neighbor 190.10.50.1 remote-as 100
neighbor 170.10.20.2 remote-as 300
network 150.10.0.0
redistribute static

ip route 190.10.0.0 255.255.0.0 null0
```

```
RTB#
router bgp 100
neighbor 150.10.30.1 remote-as 100
network 190.10.50.0
```

```
RTE#
router bgp 300
neighbor 170.10.20.1 remote-as 100
network 170.10.0.0
```

RTA достигает 170.10.0.0 через 300 i. "300 i" обозначает, что следующей AS пути является 300, а источником маршрута является IGP. RTA также достигает 190.10.50.0 через i. Данное "i" обозначает, что запись находится в той же AS, а источником является IGP. RTE достигает 150.10.0.0 через 100 i. "100 i" обозначает, что следующей AS является 100, а источником является IGP. RTE также достигает 190.10.0.0 через 100 ?. "100 ?" обозначает, что следующей AS является 100, а источник является неполным или получен из статического маршрута.

## [Атрибут BGP "следующий переход"](#)

Атрибут BGP "следующий переход" представляет собой IP-адрес следующего перехода, который необходимо использовать для достижения определенного адреса назначения.

**Для eBGP следующим переходом всегда является IP-адрес соседа, заданного командой `neighbor`.** В примере данного раздела RTC объявляет 170.10.0.0 RTA со следующим переходом 170.10.20.2. RTA объявляет 150.10.0.0 RTC со следующим переходом 170.10.20.1. Для iBGP протокол указывает, что следующий переход, объявляемый eBGP, должен сохраняться в iBGP. Следуя этому правилу, RTA объявляет 170.10.0.0 RTB, своему соседу по iBGP, со следующим переходом 170.10.20.2. Поэтому, следуя RTB, следующим переходом для достижения 170.10.0.0 будет 170.10.20.2, а не 150.10.30.1.

Необходимо убедиться в том, что RTB может достичь 170.10.20.2 через IGP. В противном



случае RTB будет отбрасывать пакеты с местом назначения 170.10.0.0, поскольку адрес следующего перехода будет недоступен. Например, если в RTB работает iGRP, можно также запустить iGRP в сети 170.10.0.0 RTA. Желательно сделать iGRP пассивным в соединении с RTC, чтобы обмен происходил только с использованием BGP.

```
RTA#
router bgp 100
neighbor 170.10.20.2 remote-as 300
neighbor 150.10.50.1 remote-as 100
network 150.10.0.0
RTB#
router bgp 100
neighbor 150.10.30.1 remote-as 100
RTC#
router bgp 300
neighbor 170.10.20.1 remote-as 100
network 170.10.0.0
```

**Примечание:** RTC объявляет 170.10.0.0 RTA со следующим переходом 170.10.20.2.

**Примечание:** RTA объявляет 170.10.0.0 RTB со следующим переходом 170.10.20.2. Следующий переход eBGP сохраняется в iBGP.

Следует уделять особое внимание при работе с сетями с множественным доступом и нешироковещательным множественным доступом (NBMA). [Дополнительные сведения содержатся в разделах Следующий переход BGP \(сети с множественным доступом\) и Следующий переход BGP \(NBMA\).](#)

### [Следующий переход BGP \(сети с множественным доступом\)](#)

В данном примере показано, как ведет себя атрибут "следующий переход" в сетях с множественным доступом, таких как Ethernet.

Предположим, что в RTC и RTD в AS300 работает OSPF. У RTC налажен обмен информацией с RTA с помощью BGP. RTC может достичь сети 180.20.0.0 через 170.10.20.3. Когда RTC пересылает RTA обновление BGP о 180.20.0.0, RTC использует следующий переход 170.10.20.3. RTC не использует собственный IP-адрес 170.10.20.2. RTC использует данный адрес, поскольку сеть между RTA, RTC и RTD является сетью с множественным доступом. Когда RTA использует RTD в качестве следующего перехода для достижения 180.20.0.0, это является более разумным, чем дополнительный переход через RTC.

**Примечание:** RTC объявляет 180.20.0.0 RTA со следующим переходом 170.10.20.3.

Если общая среда RTA, RTC и RTD является не средой с множественным доступом, а NBMA, возникают дополнительные сложности.

### [Следующий переход BGP \(NBMA\)](#)

На схеме общая среда отображена облаком. Если общая среда является облаком frame relay или любой NBMA, поведение будет таким же, как и при подключении через Ethernet. RTC объявляет 180.20.0.0 RTA со следующим переходом 170.10.20.3.

Проблема состоит в том, что у RTA нет прямого постоянного виртуального контура (PVC) к RTD и невозможно выполнить следующий переход. В данном случае маршрутизация не работает.

В данной ситуации помогает команда `next-hop-self`.

### команда `next-hop-self`

В ситуациях со следующим переходом как в примере Следующий переход BGP (NBMA) можно использовать команду `next-hop-self`. Синтаксис:

```
neighbor {ip-address | peer-group-name} next-hop-self
```

Команда `next-hop-self` позволяет явно указывать BGP на использование определенных IP-адресов в качестве следующих переходов.

Для примера Следующий переход BGP (NBMA) проблема решается при помощи следующей конфигурации:

```
RTC#  
router bgp 300  
neighbor 170.10.20.1 remote-as 100  
neighbor 170.10.20.1 next-hop-self  
RTC объявляет 180.20.0.0 со следующим переходом 170.10.20.2.
```

### Запасной путь BGP

На данной схеме в RTA и в RTC работает eBGP. В RTB и в RTC работает eBGP. В RTA и в RTB работает один из протоколов IGP, например RIP, IGRP или другой протокол. По определению у обновлений eBGP расстояние равно 20, что меньше расстояний IGP. По умолчанию расстояния равны:

120 для RIP

100 для IGRP

90 для EIGRP

110 для OSPF

RTA получает обновления о 160.10.0.0 через два протокола маршрутизации:

eBGP с расстоянием 20

IGP с расстоянием, превышающим 20

По умолчанию расстояния BGP равны:

Внешнее расстояние 20

Внутреннее расстояние 200

Внутреннее расстояние 200

Однако для изменения этих расстояний по умолчанию можно воспользоваться командой `distance`:

```
distance bgp external-distance internal-distance local-distance
```

RTA использует eBGP через RTC из-за более короткого расстояния.

Если необходимо, чтобы RTA запомнил информацию о 160.10.0.0 через RTB (IGP), есть два варианта:

Изменить внешнее расстояние eBGP или расстояние IGP.

**Примечание:** Вносить это изменение не рекомендуется.

Использование запасного пути BGP.

Запасной путь BGP делает маршрут IGP предпочтительным.

[Введите команду `network адрес backdoor`.](#)

Настроенная сеть является сетью, достигать которой необходимо с использованием IGP. Для BGP данная сеть обрабатывается так же, как локально назначенная сеть, за исключением того, что обновления BGP не объявляют данную сеть.

```
RTA#  
router eigrp 10  
  
network 150.10.0.0
```

```
router bgp 100  
neighbor 2.2.2.1 remote-as 300  
network 160.10.0.0 backdoor
```

Сеть 160.10.0.0 обрабатывается как локальная запись и не объявляется как обычная запись о сети.

RTA изучает 160.10.0.0 от RTB с использованием EIGRP с расстоянием 90. RTA также изучает адрес от RTC с использованием eBGP с расстоянием 20. Обычно eBGP является предпочтением, но из-за команды `network backdoor`, EIGRP является предпочтением.

## [Synchronization](#)

Перед тем, как рассматривать синхронизацию, рассмотрите следующую ситуацию. RTC в AS300 рассылает обновления о 170.10.0.0. В RTA и в RTB работает iBGP, поэтому RTB получает обновления и может достичь 170.10.0.0 через следующий переход 2.2.2.1. Следует помнить, что следующий переход сохраняется в iBGP. Чтобы достичь следующего перехода, RTB должен передавать трафик на RTE.

Предположим, что RTA не перераспределит сеть 170.10.0.0 в IGP. В этот момент времени RTE даже не подозревает о существовании 170.10.0.0.

Если RTB начинает объявлять AS400 о том, что RTB может достичь 170.10.0.0, трафик, поступающий из RTD в RTB с адресом назначения 170.10.0.0, поступает и отбрасывается в RTE.

Синхронизация указывает на то, что если ваша AS передает трафик из другой AS в третью, BGP не должен объявлять маршрут до тех пор, пока все маршрутизаторы в вашей AS не изучили маршрут с использованием IGP. BGP ожидает, пока IGP не распространит маршрут по AS. После BGP объявляет маршрут внешним равноправным узлам.

В примере данного раздела RTB ожидает получения информации о 170.10.0.0 с использованием IGP. После RTB начинает передачу обновления в RTD. Можно заставить RTB считать, что IGP распространил информацию, добавив в RTB статический маршрут, указывающий на 170.10.0.0. Необходимо убедиться в том, что другие маршрутизаторы могут достичь 170.10.0.0.

### Отключение синхронизации

В некоторых случаях синхронизация не требуется. Если трафик не передается из другой AS через вашу AS, синхронизацию можно отключить. Синхронизацию также можно отключить в случае, если на всех маршрутизаторах вашей AS работает BGP. Отключение данной функции может позволить передавать меньшее количество маршрутов в IGP и позволит BGP сходиться быстрее.

Отключение синхронизации не выполняется автоматически. Если во всех маршрутизаторах AS работает BGP и IGP не используется вообще, у маршрутизатора нет возможности узнать об этом. Маршрутизатор бесконечно долго ожидает обновления информации через IGP об определенном маршруте перед тем, как передать маршрут внешним равноправным узлам. Чтобы маршрутизация работала правильно, в данном случае необходимо вручную отключить синхронизацию:

```
router bgp 100
no synchronization
```

**Примечание:** *Следует убедиться в выполнении команды `clear ip bgp address` для сброса сессии.*

```
RTB#
router bgp 100
network 150.10.0.0
neighbor 1.1.1.2 remote-as 400
neighbor 3.3.3.3 remote-as 100
no synchronization
!--- RTB puts 170.10.0.0 in its IP routing table and advertises the network !--- to RTD, even if
RTB does not have an IGP path to 170.10.0.0. RTD# router bgp 400 neighbor 1.1.1.1 remote-as 100
network 175.10.0.0 RTA# router bgp 100 network 150.10.0.0 neighbor 3.3.3.4 remote-as 100
```

### Атрибут "вес"

Атрибут "вес" – это атрибут, добавленный Cisco. Данный атрибут использует вес для выбора лучшего маршрута. Вес назначается маршрутизатору локально. Данное значение имеет смысл только для определенного маршрутизатора. Значение не распространяется и не передается в обновлениях маршрутов. Вес может быть числом от 1 до 65 535. Путям,

начинающимся в маршрутизаторе, по умолчанию задан вес 32 768, а вес других путей равен 0.

При существовании нескольких маршрутов до одного адреса назначения предпочитают маршруты с более высоким весом. Взгляните на пример данного раздела. RTA узнал о сети 175.10.0.0 от AS4. RTA распространяет обновление RTC. RTB также узнал о сети 175.10.0.0 от AS4. RTB распространяет обновление RTC. У RTC теперь есть два пути для достижения 175.10.0.0 и он должен решить, какой путь ему использовать. Если задать вес обновлений в RTC, поступающих от RTA, таким образом, чтобы вес был больше веса обновлений, поступающих из RTB, RTC получает явное указание на использование RTA в качестве следующего перехода для достижения 175.10.0.0. Задать вес можно различными способами:

### Использование команды neighbor.

```
neighbor {ip-address | peer-group} weight weight
```

### Использование списков доступа AS\_PATH.

```
ip as-path access-list access-list-number {permit | deny} as-regular-expression neighbor ip-address filter-list access-list-number weight weight
```

### Использование карт маршрутов.

```
RTC#
router bgp 300
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 weight 200
!--- The route to 175.10.0.0 from RTA has a 200 weight.
neighbor 2.2.2.2 remote-as 200
neighbor 2.2.2.2 weight 100 !--- The route to 175.10.0.0 from RTB has a 100 weight.
```

RTA, у которого значение веса больше, является предпочтительным в качестве следующего перехода.

Такой же результат можно получить с использованием AS\_PATH и списков фильтров IP.

```
RTC#
router bgp 300
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 filter-list 5 weight 200
neighbor 2.2.2.2 remote-as 200
neighbor 2.2.2.2 filter-list 6 weight 100
...
ip as-path access-list 5 permit ^100$
!--- This only permits path 100. ip as-path access-list 6 permit ^200$ ...
```

То же можно получить с использованием карт маршрутов.

```
RTC#
router bgp 300
neighbor 1.1.1.1 remote-as 100
```

```
neighbor 1.1.1.1 route-map setweightin in
neighbor 2.2.2.2 remote-as 200
neighbor 2.2.2.2 route-map setweightin in
...
ip as-path access-list 5 permit ^100$
...

route-map setweightin permit 10
match as-path 5
set weight 200
!--- Anything that applies to access list 5, such as packets from AS100, has weight 200. route-
map setweightin permit 20 set weight 100 !--- Anything else has weight 100.
```

**Примечание:** Можно модифицировать вес для предпочтения пути BGP MPLS VPN с путем IGP как Резервная копия.

**Примечание:** Для получения дополнительной информации обратитесь к этому документу Сообщества Cisco Support, который описывает, как настроить маршрутизатор, чтобы иметь предпочтительный путь и на основном и на неисправные состояния и перенаправить на восстановлении основного пути: [Предпочтение Пути BGP MPLS VPN с Резервной копией IGP](#)

## Атрибут "локальное предпочтение"

Локальное предпочтение указывает на AS, которая является предпочтительной для выхода из AS в целях достижения определенной сети. Путь с более высоким локальным предпочтением является более предпочитаемым. Значение по умолчанию для локального предпочтения равно 100.

В отличие от атрибута "вес", который относится к локальному маршрутизатору, локальное предпочтение является атрибутом, которым обмениваются маршрутизаторы одной AS.

### Локальное предпочтение задается при помощи команды `bgp default local-preference value`.

Задать локальное предпочтение можно также при помощи карты маршрутов, что демонстрирует пример в данном разделе:

**Примечание:** Необходимо выполнить программный сброс (т.е. очистите процесс BGP на маршрутизаторе) для изменений, которые будут приняты к рассмотрению. Для очистки процесса BGP используйте `clear ip bgp [мягкая] [входящая/исходящая]` команда, где **мягкий** указывает, что программный сброс, не разрывая сеанс и **[входящий/исходящий]** задает входящий или исходящая конфигурация. Если **входящий/исходящий** не задан и не входящий и сеансы исходящего соединения, перезагружены.

**Команда `bgp default local-preference` задает локальное предпочтение в обновлениях, рассылаемым из маршрутизатора равноправным узлам той же AS.** На схеме в данном разделе, AS256 получает обновления о 170.10.0.0 с двух разных сторон организации. Локальное предпочтение помогает определить, каким путем следует выходить из AS256 для достижения этой сети. Предположим, что RTD является предпочтительной точкой выхода. Данная конфигурация задает локальное предпочтение для обновлений, поступающих из AS300 в 200 и для обновлений, поступающих из AS100 в 150:

```
RTC#
router bgp 256
neighbor 1.1.1.1 remote-as 100
neighbor 128.213.11.2 remote-as 256
```

```
bgp default local-preference 150
```

```
RTD#  
router bgp 256  
neighbor 3.3.3.4 remote-as 300  
neighbor 128.213.11.1 remote-as 256  
bgp default local-preference 200
```

В данной конфигурации RTC устанавливает локальное предпочтение всех обновлений в значение 150. Такой же RTD устанавливает локальное предпочтение всех обновлений в значение 200. В AS256 выполняется обмен локальными предпочтениями. Поэтому RTC и RTD понимают, что у сети 170.10.0.0 более высокое значение локального предпочтения в случае, когда обновления поступают из AS300, а не из AS100. Весь трафик в AS256, адреса назначения которого находятся в той сети, передается через RTD в качестве точки выхода.

Использование карт маршрутов обеспечивает больше гибкости. В примере данного раздела все обновления, получаемые RTD, при поступлении обновлений в RTD помечены локальным предпочтением 200. Обновления, поступающие из AS34 также помечены локальным предпочтением 200. Эта метка может быть ненужной. Поэтому можно использовать карты маршрутов для указания определенных обновлений, которые необходимо пометить определенным локальным предпочтением. Например:

```
RTD#  
router bgp 256  
neighbor 3.3.3.4 remote-as 300  
neighbor 3.3.3.4 route-map setlocalin in  
neighbor 128.213.11.1 remote-as 256  
....  
ip as-path access-list 7 permit ^300$  
...  
  
route-map setlocalin permit 10  
match as-path 7  
set local-preference 200  
  
route-map setlocalin permit 20  
set local-preference 150
```

В данной конфигурации у любого обновления, поступающего из AS300, локальное предпочтение равно 200. Все остальные обновления, такие, как обновления, поступающие из AS34, содержат значение 150.

## Атрибут "метрика"

Атрибут "метрика" также называется MULTI\_EXIT\_DISCRIMINATOR, MED (BGP4) или INTER\_AS (BGP3). Данный атрибут подсказывает внешним соседям о предпочтительном маршруте в AS. Атрибут обеспечивает динамический способ влияния на другую AS в отношении достижения определенного маршрута при наличии нескольких точек входа в эту AS. Предпочтительным является меньшее значение метрики.

В отличие от локального предпочтения обмен метриками выполняется между AS. Метрика поступает в AS, но не выходит из нее. Когда обновление поступает в AS с определенной метрикой, эта метрика используется для принятия решения в AS. Когда это обновление поступает в третью AS, метрике снова присваивается значение 0. На схеме в данном разделе показан процесс установки метрик. По умолчанию значение метрики равно 0.



За исключением случаев, когда маршрутизатор получает другие направления, он сравнивает метрики для путей от соседей в той же AS. [Чтобы маршрутизатор сравнивал метрики, поступающие от соседей из других AS, необходимо в маршрутизаторе ввести специальную команду настройки `bgp always-compare-med`.](#)

**Примечание:** Есть две команды настройки BGP, которые могут повлиять на выбор пути на основании дискриминатора для нескольких выходов (MED). [Это команда `bgp deterministic-med` и команда `bgp always-compare-med`.](#) Ввод команды `bgp deterministic-med` обеспечивает сравнение переменной MED при выборе маршрутов в случае, когда выполняют объявление различные равноправные узлы из одной AS. Ввод команды `bgp always-compare-med` обеспечивает сравнение MED для путей от соседей из различных AS. Команду `bgp always-compare-med` целесообразно использовать, когда несколько поставщиков услуг или предприятий договариваются о единой политике настройки MED. [Чтобы получить общее представление о том, как эти команды влияют на выбор пути BGP, см. Чем команда `bgp deterministic-med` отличается от команды `bgp always-compare-med`.](#)

На схеме в данном разделе AS100 получает информацию о сети 180.10.0.0 через три различных маршрута: RTC, RTD и RTB. RTC и RTD находятся в AS300, а RTB находится в AS400.

В данном примере проигнорировано сравнение AS-Path на RTA [командой `bgp bestpath as-path ignore`](#). Это настроено, чтобы вынудить BGP упасть на следующий атрибут для сравнения маршрутов (в этой метрике случая или MED). Если команда будет опущена, то BGP установит маршрут 180.10.0.0 от RTC маршрутизатора, поскольку это имеет самый короткий AS-Path.

Предположим, поступающая из RTC метрика установлена в значение 120, поступающая из RTD метрика равна 200, а поступающая из RTB метрика равна 50. По умолчанию маршрутизатор сравнивает метрики, поступающие от соседей в одной AS. Поэтому RTA может сравнить только метрику, поступающую из RTC, с метрикой, поступающей из RTD. В качестве следующего перехода RTA выбирает RTC, поскольку 120 меньше 200. Когда RTA получает обновление от RTB с метрикой 50, RTA не может сравнить метрику со значением 120, поскольку RTC и RTB находятся в различных AS. RTA должен делать выбор на основании других атрибутов.

**Чтобы явно принудить RTA сравнивать метрики, необходимо в RTA ввести команду `bgp always-compare-med`.** Данный процесс демонстрируют следующие настройки:

RTA#

```
router bgp 100
neighbor 2.2.2.1 remote-as 300
neighbor 3.3.3.3 remote-as 300
neighbor 4.4.4.3 remote-as 400
bgp bestpath as-path ignore
....
```

RTC#

```
router bgp 300
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-map setmetricout out
neighbor 1.1.1.2 remote-as 300
```

```
route-map setmetricout permit 10
set metric 120
```



```
RTD#
router bgp 300
neighbor 3.3.3.2 remote-as 100
neighbor 3.3.3.2 route-map setmetricout out
neighbor 1.1.1.1 remote-as 300
```

```
route-map setmetricout permit 10
set metric 200
```

```
RTB#
router bgp 400
neighbor 4.4.4.4 remote-as 100
neighbor 4.4.4.4 route-map setmetricout out
```

```
route-map setmetricout permit 10
set metric 50
```

С этими настройками RTA использует RTC в качестве следующего перехода с учетом того, что прочие атрибуты одинаковы. Чтобы включить RTB в сравнение меток, необходимо настроить RTA следующим образом:

```
RTA#
router bgp 100
neighbor 2.2.21 remote-as 300
neighbor 3.3.3.3 remote-as 300
neighbor 4.4.4.3 remote-as 400
bgp always-compare-med
```

В данном случае RTA выбирает RTB в качестве лучшего следующего перехода для достижения сети 180.10.0.0.

*Можно также задать метрику при перераспределении маршрутов в BGP при помощи команды `default-metric number`.*

Предположим, что в примере данного раздела RTB передает сеть в AS100 статически. Вот настройка:

```
RTB#
router bgp 400
redistribute static
default-metric 50
```

```
ip route 180.10.0.0 255.255.0.0 null 0
!--- This causes RTB to send out 180.10.0.0 with a metric of 50.
```

## Атрибут "сообщество"

Атрибут "сообщество" является переходным необязательным атрибутом из диапазона от 0 до 4 294 967 200. Атрибут сообщества является способом группирования адресов назначения в определенное сообщества и применения решений о маршрутизации в соответствии с этими сообществами. Помимо всего прочего решения о маршрутах принимаются, получают предпочтения и перераспределяются.

Для задания атрибутов сообщества можно использовать карты маршрутов. У команды `set карты маршрутов` следующий синтаксис:

```
set community community-number [additive] [well-known-community]
```

Вот несколько заранее определенных хорошо известных сообществ, используемых с данной командой:

**no-export** — не дает объявление к узлам eBGP. Поддерживать данный маршрут в AS.

**нет - дают объявление** — не объявляют этот маршрут ни к какому узлу, внутреннему или внешнему.

**Интернет** — Объявляет этот маршрут интернет-сообществу. Данному сообществу принадлежат все маршрутизаторы.

**local-as**—использовать схемы объединения для предотвращения выхода пакетов за пределы локальной AS.

Вот два примера карт маршрутов, определяющих сообщества:

```
• route-map communitymap
  match ip address 1
  set community no-advertise
```

или

```
• route-map setcommunity
  match as-path 1
  set community 200 additive
```

Если не задать ключевое слово **additive**, 200 заменяет любое уже существующее старое сообщество. При использовании ключевого слова **additive**, выполняется добавление 200 к сообществу. Даже при настройке атрибута сообщества данный атрибут по умолчанию не передается соседям. Чтобы атрибут передавался соседям, необходимо использовать следующую команду:

```
neighbor {ip-address | peer-group-name} send-community
```

Например:

```
RTA#
router bgp 100
neighbor 3.3.3.3 remote-as 300
neighbor 3.3.3.3 send-community
neighbor 3.3.3.3 route-map setcommunity out
```

В ПО Cisco IOS начиная с версии 12.0 можно настраивать сообщества в трех различных форматах: десятичном, шестнадцатеричном и в формате AA:NN. По умолчанию ПО Cisco IOS использует старый десятичный формат. **Для настройки и отображения в формате AA:NN необходимо ввести команду ip bgp-community new-format.** Первая часть AA:NN обозначает номер AS, а вторая часть обозначает 2-байтовый номер.

Например:

## Без команды ip bgp-community new-format в глобальной конфигурации ввод команды show ip bgp 6.0.0.0 приведет к отображению значения атрибута сообщества в десятичном формате.

```
, 6553620.
```

```
Router# show ip bgp 6.0.0.0 BGP routing table entry for 6.0.0.0/8, version 7 Paths: (1
available, best #1, table Default-IP-Routing-Table) Not advertised to any peer 1 10.10.10.1 from
10.10.10.1 (200.200.200.1) Origin IGP, metric 0, localpref 100, valid, external, best
Community: 6553620
```

Теперь введите на данном маршрутизаторе команду глобальной настройки ip bgp-community new-format.

```
Router# configure terminal Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# ip bgp-community new-format Router(config)# exit
```

После ввода команды глобальной настройки ip bgp-community new-format значение сообщества отображается в формате AA:NN. `show ip bgp 6.0.0.0` 100:20:

```
Router# show ip bgp 6.0.0.0 BGP routing table entry for 6.0.0.0/8, version 9 Paths: (1
available, best #1, table Default-IP-Routing-Table) Not advertised to any peer 1 10.10.10.1 from
10.10.10.1 (200.200.200.1) Origin IGP, metric 0, localpref 100, valid, external, best
Community: 100:20
```

## Примеры практического применения BGP 3

### Фильтрация BGP

Несколько различных способов фильтрации позволяют управлять отправкой и получением обновлений BGP. Обновления BGP можно фильтровать на основании информации о маршрутах, информации о пути или сообществе. Все способы позволяют добиться одинаковых результатов. Предпочтение одного способа другому зависит от конкретной конфигурации сети.

#### Фильтрация маршрутов

Для ограничения информации о маршрутах, которую изучает или объявляет маршрутизатор, можно фильтровать BGP с использованием обновлений маршрутизации, передаваемых определенному соседу или принимаемых от него. Определяется список доступа и применяется к обновлениям, поступающим к соседу или от него. В режиме настройки маршрутизатора введите следующую команду:

```
neighbor {ip-address | peer-group-name} distribute-list access-list-number {in | out}
```

В данном примере RTB становится источником сети 160.10.0.0 и передает обновление RTC. Если RTC хочет остановить распространение обновлений в AS100, необходимо задать список доступа для фильтрации этих обновлений и применить список доступа при обмене информацией с RTA:

```
RTC#
router bgp 300
network 170.10.0.0
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 distribute-list 1 out

access-list 1 deny 160.10.0.0 0.0.255.255
```

```
access-list 1 permit 0.0.0.0 255.255.255.255
!--- Filter out all routing updates about 160.10.x.x.
```

Использование списков доступа является несколько сложным при работе с суперсетями и может вызывать конфликты.

Предположим, что в примере данного раздела у RTB есть разные подсети 160.10.x.x. Целью является фильтрация обновлений и объявление только 160.0.0.0/8.

**Примечание:** Запись /8 указывает на использование 8 битов маски подсети, которая начинается с левой части IP-адреса. Данный адрес эквивалентен 160.0.0.0 255.0.0.0.

Команда `access-list 1 permit 160.0.0.0 0.255.255.255` разрешает 160.0.0.0/8, 160.0.0.0/9 и так далее. Чтобы ограничить обновление только подсетью 160.0.0.0/8, необходимо использовать расширенный список доступа следующего формата:

```
access-list 101 permit ip 160.0.0.0 0.255.255.255 255.0.0.0 0.0.0.0.
```

Данный список разрешает только 160.0.0.0/8.

[Примеры конфигурации для фильтрации сетей от равноправных узлов BGP см. в статье Блокирование одной или нескольких сетей от равноправного узла BGP.](#) В данном способе используется команда `distribute-list` со стандартным и расширенным списками контроля доступа (ACL), а также фильтрация по списку префиксов.

## Фильтрация путей

Другим типом фильтрации является фильтрация путей.

Можно задавать список доступа для входящих и исходящих обновлений с использованием информации о путях AS BGP. На схеме в данном разделе можно блокировать обновления о 160.10.0.0, чтобы они не поступали в AS100. Для блокирования обновлений определите в RTC список доступа, который не позволяет передавать в AS100 обновления, исходящие из AS200. Введите следующие команды:

```
ip as-path access-list access-list-number {permit | deny} as-regular-expression
neighbor {ip-address | peer-group-name} filter-list access-list-number {in | out}
```

В данном примере прекращается передача обновлений о 160.10.0.0 из RTC в RTA:

```
RTC#
router bgp 300
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 filter-list 1 out
!--- The 1 is the access list number below. ip as-path access-list 1 deny ^200$ ip as-path
access-list 1 permit .*
```

Команда `access-list 1` в данном примере указывает на отклонение любых обновлений с информацией о пути, которая начинается с 200 и заканчивается на 200. `^200$` " ", ^ " ", \$ " ". Поскольку RTB посылает обновления о 160.10.0.0 с информацией о пути, которая начинается с 200 и заканчивается на 200, обновления соответствуют списку доступа. Список доступа отклоняет эти обновления.

```
.* , . " ", * " ". .* , .
```

, ^200\$ ^200? При наличии AS400, как представлено на схеме в данном разделе, в обновлениях, источником которых является AS400, содержится информация о пути в виде (200, 400). В данной информации о пути 200 является первой, а 400 является последней. ^200, 200. Список доступа запрещает передачу этих обновлений в RTA, что не соответствует требованиям.

[Чтобы проверить правильность применения регулярного выражения, введите команду `show ip bgp regexp regular-expression`](#). Данная команда показывает все пути, соответствующие конфигурации регулярного выражения.

## Регулярные выражения AS

В данном разделе описывается создание регулярных выражений.

Регулярное выражение представляет собой шаблон, с которым сравнивается входная строка. При создании регулярного выражения задается строка, которой должна соответствовать входная информация. В случае с BGP задается строка, состоящая из информации о пути, которой должна соответствовать входная строка.

[В примере раздела Фильтрация путей была задана строка `^200\$`](#). Для принятия решения необходимо было, чтобы информация о путях, содержащаяся в поступающих обновлениях, соответствовала строке.

Регулярное выражение состоит из следующих элементов:

### **Диапазон**

Диапазон представляет собой последовательность символов, заключенных между левой и правой квадратными скобками. `[abcd]`.

### **Atom**

Атом представляет собой один символ. Приведем несколько примеров:

.

.

^

^

\$

\$

\

\

-

\_ (,), ({}), ({}), , .

## Часть

Часть является одним из этих символов, который придерживается атома:

\*  
\* 0 .  
+  
+ 1 .  
?  
? .

## Ответвление

Ветвь представляет собой 0 или более объединенных элементов.

Вот несколько примеров регулярных выражений:

a\*

Данное выражение указывает на любое количество букв "a", что включает также их отсутствие.

a+

Данное выражение указывает на необходимость наличия хотя бы одной буквы "a".

ab?a

Данное выражение соответствует "aa" или "aba".

\_100\_

Данное выражение обозначает "через AS100".

\_100\$

Данное выражение указывает на то, что AS100 является источником.

^100 .\*

Данное выражение указывает на передачу из AS100.

^\$

Данное выражение указывает на то, что источником является данная AS.

[Примеры фильтрации с использованием регулярных выражений см. в статье Использование регулярных выражений в BGP.](#)

## Фильтрация по сообществу BGP

В данном документе затронуто фильтрация по маршрутам и фильтрация по путям AS. Другим способом является фильтрация по сообществам. [Сообщества описываются в разделе Атрибут "сообщество", а в данном разделе приведено несколько примеров использования сообществ.](#)

В данном примере требуется, чтобы RTB устанавливал атрибут сообщества в маршрутах BGP, объявляемых RTB, чтобы RTC не распространял эти маршруты внешним равноправным узлам. **Используйте атрибут сообщества no-export.**

```
RTB#
router bgp 200
network 160.10.0.0
neighbor 3.3.3.1 remote-as 300
neighbor 3.3.3.1 send-community
neighbor 3.3.3.1 route-map setcommunity out
```

```
route-map setcommunity
match ip address 1
set community no-export
```

```
access-list 1 permit 0.0.0.0 255.255.255.255
```

**Примечание:** В данном примере для задания сообществу значения no-export используется команда `route-map setcommunity`.

**Примечание:** Команда `neighbor send-community` требуется для передачи данного атрибута в RTC.

Когда RTC получает обновления с атрибутом NO\_EXPORT, RTC не распространяет обновления внешнему равноправному узлу RTA.

**В данном примере RTB установил атрибут сообщества в значение 100 200. Данное действие перед передачей в RTC добавляет значение 100 200 в любое существующее значение сообщества.**

```
RTB#
router bgp 200
network 160.10.0.0
neighbor 3.3.3.1 remote-as 300
neighbor 3.3.3.1 send-community
neighbor 3.3.3.1 route-map setcommunity out
```

```
route-map setcommunity
match ip address 2
set community 100 200 additive
```

```
access-list 2 permit 0.0.0.0 255.255.255.255
```

Список сообществ представляет собой группу сообществ, используемых в условии `match` карты маршрутов. Список сообществ позволяет фильтровать или устанавливать атрибуты, на основании списков сообществ с другими номерами.

```
ip community-list community-list-number {permit | deny} community-number
```

Например, можно определить карту маршрутов `match-on-community`:

```
route-map match-on-community
match community 10
!--- The community list number is 10. set weight 20 ip community-list 10 permit 200 300 !--- The
community number is 200 300.
```

Можно использовать список сообществ для фильтрации или установки определенных параметров, как, например вес и метрика, в определенных обновлениях на основании значения сообщества. Во втором примере данного раздела RTB передает обновления в RTC с сообществом 100 200. Если на основании данных значений в RTC нужно установить вес, можно сделать следующее:

```
RTC#
router bgp 300
neighbor 3.3.3.3 remote-as 200
neighbor 3.3.3.3 route-map check-community in
```

```
route-map check-community permit 10
match community 1
set weight 20
```

```
route-map check-community permit 20
match community 2 exact
set weight 10
```

```
route-map check-community permit 30
match community 3
```

```
ip community-list 1 permit 100
ip community-list 2 permit 200
ip community-list 3 permit internet
```

В данном примере любой маршрут, у которого в атрибуте сообщества есть 100, соответствует списку 1. Данному маршруту присваивается вес 20. Любой маршрут, в котором в качестве сообщества указано только значение 200, соответствует списку 2 и получает вес 20. Ключевое слово `exact` указывает на то, что сообщество содержит только 200 и ничего больше. Последний список сообществ присутствует здесь для того, чтобы не отбрасывались другие обновления. Следует помнить, что все не соответствующие обновления по умолчанию отбрасываются. Ключевое слово `internet` указывает на все маршруты, поскольку все маршрутизаторы являются участниками сообщества Интернет.

[Дополнительные сведения см. в статье Использование значений сообществ BGP для управления политиками маршрутизации в сети вышестоящего поставщика.](#)

## [Соседи BGP и карты маршрутов](#)

Чтобы выполнять фильтрацию входящих или исходящих обновлений, можно использовать команду `neighbor` в сочетании с картами маршрутов.



Карты маршрутов, связанные с оператором `neighbor`, не действуют на входящие обновления при выполнении проверки по IP-адресам:

```
neighbor ip-address route-map route-map-name
```

Предположим, что на схеме в данном разделе необходимо, чтобы RTC изучал информацию из AS200 только о сетях, локальных для AS200 и никакую другую. Кроме того, необходимо устанавливать принимаемым маршрутам вес 20. **Воспользуйтесь комбинацией списков доступа `neighbor` и `as-path`:**

```
RTC#
router bgp 300
network 170.10.0.0
neighbor 3.3.3.3 remote-as 200
neighbor 3.3.3.3 route-map stamp in
```

```
route-map stamp
match as-path 1
set weight 20
```

```
ip as-path access-list 1 permit ^200$
```

Любые обновления, приходящие из AS200, содержат информацию о пути, которая начинается с 200 и заканчивается на 200. Эти обновления разрешены. Любые другие обновления отбрасываются.

Предположим, что требуется следующее:

Принимать обновления, происходящие из AS200 и содержащие значение веса 20

Отбрасывать обновления, происходящие из AS400

Вес, равный 10 для других обновлений

```
RTC#
router bgp 300
network 170.10.0.0
neighbor 3.3.3.3 remote-as 200
neighbor 3.3.3.3 route-map stamp in
```

```
route-map stamp permit 10
match as-path 1
set weight 20
```

```
route-map stamp permit 20
match as-path 2
set weight 10
```

```
ip as-path access-list 1 permit ^200$
```

```
ip as-path access-list 2 permit ^200 600 .*
```

Данный оператор устанавливает вес 20 для обновлений, локальных для AS200.

Оператор также устанавливает вес 10 для обновлений, приходящих из-за AS400, и отбрасывает обновления, приходящие из AS400.

## Использование команды `set as-path prepend`

В некоторых случаях для управления процессом принятия решений в BGP необходимо управлять информацией о пути. Команда, которая используется с картой маршрутов:

```
set as-path prepend as-path# as-path#
```

[Предположим, что на схеме раздела Соседи BGP и карты маршрутов RTC объявляет свою сеть 170.10.0.0 двум различным AS, AS100 и AS200.](#) Когда информация распространяется к AS600, маршрутизаторы в AS600 имеют сведения о возможности доступа к сети приблизительно 170.10.0.0 с помощью двух других маршрутов. Первый маршрут проходит через AS100 с путем (100, 300), а второй маршрут проходит через AS400 с путем (400, 200, 300). Если все прочие атрибуты совпадают, AS600 берет кратчайший путь и выбирает маршрут через AS100.

AS300 получает весь трафик через AS100. Если необходимо повлиять на принятие этого решения со стороны AS300, можно сделать так, чтобы путь через AS100 казался длиннее чем путь, пролегающий через AS400. Это можно сделать добавлением номеров AS в информацию о существующем пути, объявляемую AS100. Обычно используется повторение номера собственной AS следующим способом:

```
RTC#
router bgp 300
network 170.10.0.0
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-map SETPATH out

route-map SETPATH
set as-path prepend 300 300
```

Благодаря этой конфигурации AS600 получает обновления о 170.10.0.0 через AS100 со следующей информацией о пути: (100, 300, 300, 300). Данная информация о пути длиннее, чем (400, 200, 300), что AS600 получает от AS400.

## [Группы сторон BGP](#)

Группа равноправных узлов BGP представляет собой группу соседей BGP с одинаковыми политиками обновления. Обычно политики обновления определяются картами маршрутов, списками распределения и списками фильтров. Вместо того, чтобы определять одинаковые политики для соседей по отдельности, определяется имя группы равноправных узлов и эти политики назначаются группе.

Участники группы равноправных узлов наследуют все параметры конфигурации группы равноправных узлов. В участниках также можно настраивать переопределение этих параметров, если параметры не влияют на исходящие обновления. Можно переопределять только параметры, установленные для входящих.

Чтобы определить группу равноправных узлов, введите следующую команду:

```
neighbor peer-group-name peer-group
```

В данном примере группы равноправных узлов применяются к внутренним и внешним соседям BGP:

```
RTC#
router bgp 300
neighbor internalmap peer-group
neighbor internalmap remote-as 300
neighbor internalmap route-map SETMETRIC out
neighbor internalmap filter-list 1 out
neighbor internalmap filter-list 2 in
neighbor 5.5.5.2 peer-group internalmap
neighbor 5.6.6.2 peer-group internalmap
neighbor 3.3.3.2 peer-group internalmap
neighbor 3.3.3.2 filter-list 3 in
```

**Данная конфигурация определяет группу равноправных узлов с именем internalmap. Конфигурация определяет для группы несколько политик, таких, как карта маршрутов SETMETRIC, которая устанавливает метрику 5 и два разных списка фильтров, 1 и 2. Конфигурация применяет группу равноправных узлов ко всем внутренним соседям, RTE, RTF и RTG. Кроме того, конфигурация определяет отдельный список фильтров 3 для соседнего узла RTE. Данный список фильтров переопределяет в группе равноправных узлов список фильтров 2.**

**Примечание: Можно переопределять только параметры, влияющие на входящие обновления.**

**Вот как можно использовать группы равноправных узлов с внешними соседями. С использованием той же схемы из этого раздела RTC настраивается с группой равноправных узлов externalmap и группа равноправных узлов применяется к внешним соседям.**

```
RTC#
router bgp 300
neighbor externalmap peer-group
neighbor externalmap route-map SETMETRIC
neighbor externalmap filter-list 1 out
neighbor externalmap filter-list 2 in
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 peer-group externalmap
neighbor 4.4.4.2 remote-as 600
neighbor 4.4.4.2 peer-group externalmap
neighbor 1.1.1.2 remote-as 200
neighbor 1.1.1.2 peer-group externalmap
neighbor 1.1.1.2 filter-list 3 in
```

**Примечание: В данных конфигурациях операторы remote-as определяются за рамками группы равноправных узлов, поскольку необходимо определить другие внешние AS. Кроме того, входящие обновления соседа 1.1.1.2 переопределяются при назначении списка фильтров 3.**

[Дополнительные сведения по группам равноправных узлов см. в статье Группы равноправных узлов BGP.](#)

**Примечание: В ПО Cisco IOS 12.0(24)S Cisco была добавлена функция динамического обновления групп равноправных узлов BGP. Данная функция также доступна в более новых версиях ПО Cisco IOS. Функция добавляет новый алгоритм, который динамически вычисляет и оптимизирует группы обновлений соседей, использующих одинаковые**

исходящие политики. Эти соседи могут использовать одинаковые сообщения обновления. В более старых версиях ПО Cisco IOS в основе группы сообщений обновления BGP лежала конфигурация группы равноправных узлов. Данный способ группирования обновлений ограничивал исходящие политики и определенные конфигурации сессий. Функция динамического обновления групп равноправных узлов BGP отделяет репликацию групп обновлений от конфигурации группы равноправных узлов. Это отделение улучшает время схождения и гибкость настройки соседей. [Дополнительные сведения см. в документе Динамическое обновление группы равноправных узлов BGP.](#)

## Примеры практического применения BGP 4

### CIDR и агрегированные адреса

Одним из основных усовершенствований BGP4 по сравнению с BGP3 является бесклассовая междоменная маршрутизация (CIDR). CIDR или организация суперсетей является новым подходом к использованию IP-адресов. В CIDR отсутствует обозначение классов, как, например, класс А, В или С. Например, сеть 192.213.0.0 была недопустимой сетью класса С. Теперь сеть является допустимой суперсетью 192.213.0.0/16. Запись "16" указывает на количество битов маски подсети, которая начинается с левой части IP-адреса. Такое представление похоже на запись 192.213.0.0 255.255.0.0.

Для минимизации размера таблиц маршрутизации используется агрегирование. Агрегирование представляет собой процесс, объединяющий характеристики нескольких различных маршрутов таким образом, что становится возможным объявление одного маршрута. В данном примере RTB создает сеть 160.10.0.0. RTC настраивается на распространение суперсети этого маршрута 160.0.0.0 в RTA:

```
RTB#
router bgp 200
neighbor 3.3.3.1 remote-as 300
network 160.10.0.0

#RTC
router bgp 300
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
network 170.10.0.0
aggregate-address 160.0.0.0 255.0.0.0
```

RTC распространяет в RTA агрегированный адрес 160.0.0.0.

### Команды агрегирования

Если большой набор команд агрегирования. Чтобы получить нужное поведение агрегирования, необходимо разобраться в работе каждой команды.

[Первая команда встречается первой в примере раздела CIDR и агрегированные адреса:](#)

`aggregate-address address-mask`

Данная команда объявляет префиксный маршрут и все более точные маршруты. Команда `aggregate-address 160.0.0.0` распространяет дополнительную сеть 160.0.0.0, но не препятствует распространению 160.10.0.0 в RTA. Результатом является распространение обеих сетей 160.0.0.0 и 160.10.0.0 в RTA, что является объявлением префикса и более

точного маршрута.

**Примечание:** Если в таблице маршрутизации BGP нет более точного маршрута какого-либо адреса, то этот адрес нельзя агрегировать.

Например, RTB не может создать агрегирование для 160.0.0.0, если в RTB нет более точной записи 160.0.0.0 в таблице BGP. Допускается добавление более точного маршрута в таблицу BGP. Добавление маршрута может быть осуществлено при помощи:

Входящих обновлений из других AS

Перераспределения IGP или статических маршрутов в BGP

Команда `network`, например, `network 160.10.0.0`

Если необходимо, чтобы RTC распространял только сеть 160.0.0.0, а не более точный маршрут, введите следующую команду:

```
aggregate-address address mask summary-only
```

Данная команда объявляет только префикс. Команда подавляет все более точные маршруты.

Команда `aggregate 160.0.0.0 255.0.0.0 summary-only` распространяет сеть 160.0.0.0 и подавляет более точный маршрут 160.10.0.0.

**Примечание:** При агрегировании сети, добавленной в BGP при помощи оператора `network`, запись сети всегда добавляется в обновления BGP. Данное добавление выполняется даже при использовании команды `aggregate summary-only`. [Данная ситуация описывается в примере раздела Пример CIDR №1.](#)

```
aggregate-address address-mask as-set
```

Данная команда объявляет префикс и более точные маршруты. Команда включает информацию `as-set` в информацию о путях из обновлений маршрутов.

```
aggregate 129.0.0.0 255.0.0.0 as-set
```

[Данная команда описывается в разделе Пример CIDR №2 \(as-set\).](#)

Если при агрегировании необходимо подавлять уточненные маршруты, определите карту маршрутов и примените ее к агрегированиям. Это действие позволит более выборочно подходить к подавлению уточненных маршрутов.

```
aggregate-address address-mask suppress-map map-name
```

Данная команда объявляет префикс и более точные маршруты. При этом она подавляет объявления опираясь на карты маршрутов. [Предположим, что в случае со схемой из раздела CIDR и агрегированные адреса необходимо агрегировать 160.0.0.0, подавить уточненный маршрут 160.20.0.0 и разрешить распространение 160.10.0.0.](#) Воспользуйтесь следующей картой маршрутов:

```
route-map CHECK permit 10
match ip address 1
```

```
access-list 1 permit 160.20.0.0 0.0.255.255
access-list 1 deny 0.0.0.0 255.255.255.255
```

По определению `suppress-map` здесь присутствует подавление обновлений любых пакетов, определяемых списком доступа.

После примените карту маршрутов к оператору `aggregate`.

```
RTC#
router bgp 300
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 remote-as 100
network 170.10.0.0
aggregate-address 160.0.0.0 255.0.0.0 suppress-map CHECK
```

Ниже представлен другой вариант:

```
aggregate-address address-mask attribute-map map-name
```

Данная команда позволяет устанавливать атрибуты, например, метрику, во время передачи агрегированных маршрутов. Чтобы задать источник агрегированных маршрутов в IGP, примените к команде `aggregate` `attribute-map` следующую карту маршрутов:

```
route-map SETMETRIC
set origin igp
```

```
aggregate-address 160.0.0.0 255.0.0.0 attribute-map SETORIGIN
```

[Дополнительные сведения см. в статье Общие сведения по агрегированию маршрутов в BGP.](#)

## [Пример CIDR 1](#)

Запрос: Разрешить RTB объявлять префикс 160.0.0.0 и подавлять уточненные маршруты. Проблема с данным требованием заключается в том, что сеть 160.10.0.0 является локальной для AS200, и это означает, что AS200 является источником 160.10.0.0. **Невозможно сделать так, чтобы RTB создавал префикс для 160.0.0.0 без распространения записи для 160.10.0.0 даже с использованием команды `aggregate summary-only`.** RTB создает обе сети, поскольку RTB является источником 160.10.0.0. У этой проблемы есть два решения.

Первое решение заключается в использовании статических маршрутов и их перераспределение в BGP. RTB (??).

```
RTB#
router bgp 200
neighbor 3.3.3.1 remote-as 300
redistribute static
!--- This generates an update for 160.0.0.0 !--- with the origin path as "incomplete". ip route
160.0.0.0 255.0.0.0 null0
```

Второе решение в дополнение к статическому маршруту предполагает добавление записи для команды `network`. Данная запись приводит к тому же результату за исключением того,

что запись задает IGP источник обновления.

```
RTB#
router bgp 200
network 160.0.0.0 mask 255.0.0.0
!--- This entry marks the update with origin IGP. neighbor 3.3.3.1 remote-as 300 redistribute
static ip route 160.0.0.0 255.0.0.0 null0
```

### Пример CIDR 2 (актив)

Оператор `as-set` используется в агрегированном маршруте для уменьшения размера информации о пути. При использовании `as-set` номер AS указывается только однажды, вне зависимости от количества раз, которое номер AS содержится в нескольких агрегированных путях. Команда `aggregate as-set` используется в случаях, когда агрегирование информации приводит к потерям информации по отношению к атрибуту пути. В данном примере RTC получает обновления о 160.20.0.0 от RTA и обновления о 160.10.0.0 от RTB. Предположим, что RTC хочет агрегировать сеть 160.0.0.0/8 и переслать сеть в RTD. RTD не знает об источнике данного маршрута. При добавлении оператора `aggregate as-set` RTC получает явное указание генерировать информацию о пути в форме множества `{}`. Это множество содержит всю информацию о пути независимо от первого встретившегося пути.

```
RTB#
router bgp 200
network 160.10.0.0
neighbor 3.3.3.1 remote-as 300
```

```
RTA#
router bgp 100
network 160.20.0.0
neighbor 2.2.2.1 remote-as 300
```

### Пример 1:

У RTC отсутствует оператор `as-set`. RTC передает обновление 160.0.0.0/8 в RTD с информацией о пути (300), как будто бы маршрут происходит из AS300.

```
RTC#
router bgp 300
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
neighbor 4.4.4.4 remote-as 400
aggregate 160.0.0.0 255.0.0.0 summary-only
!--- This command causes RTC to send RTD updates about 160.0.0.0/8 !--- with no indication that
160.0.0.0 actually comes from two different ASs. !--- This may create loops if RTD has an entry
back into AS100 or AS200.
```

### Случай 2:

```
RTC#
router bgp 300
neighbor 3.3.3.3 remote-as 200
neighbor 2.2.2.2 remote-as 100
neighbor 4.4.4.4 remote-as 400
aggregate 160.0.0.0 255.0.0.0 summary-only
aggregate 160.0.0.0 255.0.0.0 as-set
!--- This command causes RTC to send RTD updates about 160.0.0.0/8 !--- with an indication that
```



160.0.0.0 belongs to a set {100 200}.

Следующие два субъекта, Конфедерация BGP и Рефлекторы маршрутов, предназначены для поставщиков услуг Интернета (ISP), которые хотят получить дополнительное управление над использованием равноправных связей iBGP в своих AS.

## Конфедерация BGP

Реализация конфедерации BGP уменьшает сетку элементов iBGP в AS. Хитрость заключается в разделении AS на несколько AS и назначении всей группы одной конфедерации. У каждой AS по отдельности свой полностью связанный iBGP и каждая из них связана с другими AS внутри конфедерации. Даже несмотря на то что у этих AS есть равноправные связи eBGP с AS в пределах конфедерации, AS обмениваются маршрутами как если бы они использовали iBGP. Таким образом, конфедерация сохраняет информацию о следующем переходе, метрике и локальном предпочтении. Внешней среде конфедерация представляется одной AS.

Для настройки конфедерации BGP введите следующую команду:

```
bgp confederation identifier autonomous-system
```

Идентификатор конфедерации представляет собой номер AS группы конфедерации.

При вводе данной команды организовывается равноправная связь между несколькими AS в конфедерации:

```
bgp confederation peers autonomous-system [autonomous-system]
```

Ниже представлен пример конфедерации:

Предположим, что есть AS500, состоящая из девяти адресантов BGP. Есть также и другие адресанты, не работающие с BGP, однако нас интересуют только адресанты BGP, у которых есть соединения eBGP с другими AS. Если необходимо организовать в AS500 полную сеть связи iBGP, то для каждого маршрутизатора потребуется по девять равноправных соединений. Требуется девять равноправных узлов iBGP и один равноправный узел eBGP для внешних AS.

При использовании конфедерации можно разделить AS500 на несколько AS: AS50, AS60 и AS70. AS получает идентификатор конфедерации 500. Внешняя система видит только одну AS, это AS500. Для каждой из AS50, AS60 и AS70 определяется полная сеть равноправных узлов iBGP, а также список равноправных узлов конфедерации при помощи команды `bgp confederation peers`.

Вот пример конфигурации маршрутизаторов RTC, RTD и RTA:

**Примечание:** RTA не знает об AS50, AS60 или AS70. RTA знает только об AS500.

```
RTC#
router bgp 50
bgp confederation identifier 500
bgp confederation peers 60 70
neighbor 128.213.10.1 remote-as 50 (iBGP connection within AS50)
neighbor 128.213.20.1 remote-as 50 (iBGP connection within AS50)
neighbor 129.210.11.1 remote-as 60 (BGP connection with confederation peer 60)
neighbor 135.212.14.1 remote-as 70 (BGP connection with confederation peer 70)
```



```
neighbor 5.5.5.5 remote-as 100 (EBGP connection to external AS100)
```

```
RTD#
```

```
router bgp 60
```

```
bgp confederation identifier 500
```

```
bgp confederation peers 50 70
```

```
neighbor 129.210.30.2 remote-as 60 (IBGP connection within AS60)
```

```
neighbor 128.213.30.1 remote-as 50 (BGP connection with confederation peer 50)
```

```
neighbor 135.212.14.1 remote-as 70 (BGP connection with confederation peer 70)
```

```
neighbor 6.6.6.6 remote-as 600 (EBGP connection to external AS600)
```

```
RTA#
```

```
router bgp 100
```

```
neighbor 5.5.5.4 remote-as 500 (EBGP connection to confederation 500)
```

## [Рефлекторы маршрутов](#)

Другим решением для использования равноправных связей в iBGP в пределах AS являются рефлекторы маршрутов (RR). [Как показано в разделе iBGP, адресант BGP не объявляет третьему адресанту iBGP маршрут, который адресант BGP изучил от другого адресанта iBGP.](#) Можно немного смягчить данное ограничение и обеспечить дополнительный контроль, что позволит маршрутизатору объявлять (или отражать) другим адресантам iBGP маршруты, изученные с использованием iBGP. Данное отражение маршрутов уменьшает количество равноправных узлов iBGP в пределах AS.

В обычных случаях следует поддерживать полную сеть iBGP между RTA, RTB и RTC в AS100. При использовании концепции RR RTC может быть выбран в качестве RR. Таким образом, у RTC будет частичная равноправная связь iBGP с RTA и RTB. Равноправная связь между RTA и RTB не требуется, поскольку RTC является RR для обновлений, поступающих от RTA и RTB.

### [neighbor route-reflector-client](#)

Маршрутизатор с данной командой является RR, а соседи, на которых указывает команда, являются клиентами этого RR. **В примере конфигурации RTC есть команда `neighbor route-reflector-client`, которая указывает на IP-адреса RTA и RTB.** Комбинация из RR и клиентов называется "кластером". В данном примере RTA, RTB и RTC образуют кластер с одним RR в AS100.

Другие равноправные узлы iBGP, которые не являются клиентами RR, называются "неклиенты".

В AS может быть более одного RR. В такой ситуации RR рассматривает другие RR просто как и любого другого адресанта iBGP. Другие RR могут принадлежать этому же кластеру (группе клиентов) или другим кластерам. В ситуации с простой конфигурации можно разделить AS на несколько кластеров. Каждый RR настраивается с другими RR как равноправные узлы-неклиенты, в полностью связанную топологию. Клиенты не должны налаживать равноправные связи с адресантами iBGP, не находящимися в кластере клиентов.

[Рассмотрим следующую диаграмму.](#) RTA, RTB и RTC образуют один кластер. RTC является RR. Для RTC, клиентами являются RTA и RTB, а все остальные являются неклиентами. **Следует помнить, что команда `neighbor route-reflector-client` указывает на клиентов RR.** Тот же RTD является RR для клиентов RTE и RTF. RTG является RR в третьем кластере.

**Примечание:** RTD, RTC и RTG связаны в полную сеть, а маршрутизаторы в кластере – нет. Когда RR получает маршрут, то список маршрутов RR выглядит указанным образом. Однако эти действия зависят от типа равноправного узла:

Маршруты от равноправного узла-неклиента—отражаются всем клиентам в кластере.

Маршруты от равноправного узла-клиента—отражаются всем равноправным узлам-неклиентам и равноправным узлам-клиентам.

Маршруты от равноправного узла по eBGP—обновления рассылаются всем равноправным узлам, и клиентам, и неклиентам.

Вот относительная конфигурация BGP маршрутизаторов RTC, RTD и RTB:

RTC#

```
router bgp 100
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-reflector-client
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 route-reflector-client
neighbor 7.7.7.7 remote-as 100
neighbor 4.4.4.4 remote-as 100
neighbor 8.8.8.8 remote-as 200
```

RTB#

```
router bgp 100
neighbor 3.3.3.3 remote-as 100
neighbor 12.12.12.12 remote-as 300
```

RTD#

```
router bgp 100
neighbor 6.6.6.6 remote-as 100
neighbor 6.6.6.6 route-reflector-client
neighbor 5.5.5.5 remote-as 100
neighbor 5.5.5.5 route-reflector-client
neighbor 7.7.7.7 remote-as 100
neighbor 3.3.3.3 remote-as 100
```

Поскольку присутствует отражение изученных при помощи iBGP маршрутов, может возникнуть петля информации о маршрутах. В схеме RR есть несколько способов избегания таких петель:

**originator-id** — Это - дополнительный, непереходный атрибут BGP, который 4 байта длиной. Этот атрибут создает RR. Данный атрибут содержит идентификатор маршрутизатора (RID), являющегося источником маршрута в локальной AS. Если в результате неправильной настройки информация о маршрутах возвращается к

источнику, информация игнорируется.

**cluster-list**—список кластеров описан в разделе **Несколько RR в кластере**.

## Несколько RR в кластере

Обычно в кластере клиентов находится один RR. В данном случае идентификатор маршрутизатора, который является RR, идентифицирует кластер. Для увеличения резервирования и отказа от единичных источников неисправностей в кластере может быть более одного RR. Необходимо настроить все RR в одном кластере с 4-байтовым идентификатором кластера таким образом, чтобы RR мог распознавать обновления от RR из этого же кластера.

Список кластеров представляет собой последовательность идентификаторов кластеров, через которые прошел маршрут. Когда RR отражает маршрут от клиентов RR неклиентам RR, находящимся за пределами кластера, RR добавляет в список кластеров идентификатор локального кластера. Если список кластеров у данного обновления отсутствует, RR создает список. С помощью данного атрибута RR может узнать, не возвращалась ли информация о маршрутах в тот же кластер в результате неправильной настройки. Если идентификатор локального кластера не обнаружен в списке кластеров, объявление игнорируется.

На схеме данного раздела RTD, RTE, RTF и RTH принадлежат одному кластеру. И RTD, и RTH являются RR для одного кластера.

**Примечание:** Здесь присутствует избыточность, поскольку RTH связан в полную сеть равноправных узлов со всеми RR. При отключении RTD его место занимает RTH.

Вот конфигурации RTH, RTD, RTF и RTC:

RTH#

```
router bgp 100
neighbor 4.4.4.4 remote-as 100
neighbor 5.5.5.5 remote-as 100
neighbor 5.5.5.5 route-reflector-client
neighbor 6.6.6.6 remote-as 100
neighbor 6.6.6.6 route-reflector-client
neighbor 7.7.7.7 remote-as 100
neighbor 3.3.3.3 remote-as 100
neighbor 9.9.9.9 remote-as 300
bgp cluster-id 10
```

RTD#

```
router bgp 100
neighbor 10.10.10.10 remote-as 100
neighbor 5.5.5.5 remote-as 100
neighbor 5.5.5.5 route-reflector-client
neighbor 6.6.6.6 remote-as 100
neighbor 6.6.6.6 route-reflector-client
neighbor 7.7.7.7 remote-as 100
```

```
neighbor 3.3.3.3 remote-as 100
neighbor 11.11.11.11 remote-as 400
bgp cluster-id 10
```

RTF#

```
router bgp 100
neighbor 10.10.10.10 remote-as 100
neighbor 4.4.4.4 remote-as 100
neighbor 13.13.13.13 remote-as 500
```

RTC#

```
router bgp 100
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 route-reflector-client
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 route-reflector-client
neighbor 4.4.4.4 remote-as 100
neighbor 7.7.7.7 remote-as 100
neighbor 10.10.10.10 remote-as 100
neighbor 8.8.8.8 remote-as 200
```

**Примечание:** [Для RTC не требуется команда bgp cluster-id, поскольку в этом кластере существует только один RR.](#)

**ВАЖНОЕ ПРИМЕЧАНИЕ:** В данной конфигурации не используются группы равноправных узлов. Не используйте группы равноправных узлов, если у клиентов кластера нет прямой связи с равноправными узлами iBGP и клиенты обмениваются обновлениями с использованием RR. При настройке групп равноправных узлов потенциальное удаление источника маршрута в RR передается всем клиентам в кластере. Данная передача может вызвать неполадки.

[По умолчанию в RR включена подкоманда bgp client-to-client reflection.](#) При отключении отражения BGP между клиентами в RR и организации резервных равноправных связей BGP между клиентами можно без опасений использовать группы равноправных узлов. См. [Ограничения Групп одноранговых узлов](#) для получения дополнительной информации.

## [RR и обычные адресанты BGP](#)

В AS могут быть адресанты BGP, которые не понимают концепцию RR. В данном документе такие адресанты называются обычными адресантами BGP. Схема RR предусматривает сосуществование с такими обычными адресантами BGP. Эти маршрутизаторы могут быть участниками группы клиентов или группы неклиентов. Существование таких маршрутизаторов позволяет легко и постепенно переходить от текущей модели iBGP к модели RR. Можно начать создавать кластеры, настроив один маршрутизатор в качестве RR, а другие RR и клиентов RR сделать обычными равноправными узлами iBGP. Постепенно можно увеличивать количество кластеров.

На данной схеме RTD, RTE и RTF обладают концепцией отражения маршрутов. RTC, RTA и RTB являются "обычными" маршрутизаторами. Эти маршрутизаторы нельзя настроить в качестве RR. Между этими маршрутизаторами и RTD можно организовать обычную полную

сеть iBGP. Позже, при готовности к обновлению, можно сделать из RTC RR, клиентами которого будут RTA и RTB. Клиенты не обязаны понимать схему отражения маршрутов, обновление требуется только для RR.

Вот конфигурации RTD и RTC:

RTD#

```
router bgp 100
neighbor 6.6.6.6 remote-as 100
neighbor 6.6.6.6 route-reflector-client
neighbor 5.5.5.5 remote-as 100
neighbor 5.5.5.5 route-reflector-client
neighbor 3.3.3.3 remote-as 100
neighbor 2.2.2.2 remote-as 100
neighbor 1.1.1.1 remote-as 100
neighbor 13.13.13.13 remote-as 300
```

RTC#

```
router bgp 100
neighbor 4.4.4.4 remote-as 100
neighbor 2.2.2.2 remote-as 100
neighbor 1.1.1.1 remote-as 100
neighbor 14.14.14.14 remote-as 400
```

При готовности к обновлению RTC и назначению RTC в качестве RR, удалите полную сеть iBGP и сделайте RTA и RTB клиентами RTC.

### [Избегание образования петель информации о маршрутах](#)

До сих пор в документе упоминалось два атрибута, которые можно использовать с целью избегания потенциального образования петель информации: **originator-id** и **cluster-list**.

Другим способом контроля за петлями является использование дополнительных ограничений с оператором **set** исходящих карт маршрутов. Оператор **set** для исходящих карт маршрутов не влияет на маршруты, отражаемые равноправным узлам iBGP.

Можно также наложить дополнительные ограничения на **nexthop-self**, что является параметром настройки отдельных равноправных узлов. При использовании **nexthop-self** в RR, оператор влияет только на следующий переход изученных маршрутов eBGP, поскольку следующий переход отражаемых маршрутов не должен изменяться.

### [Отключение переключающихся маршрутов](#)

В ПО Cisco IOS версии 11.0 было добавлено отключение переключающихся маршрутов. Отключение маршрутов представляет собой механизм, сводящий к минимуму нестабильность, вызываемую переключениями. Отключение маршрутов также уменьшает колебания в сети. Необходимо обозначить критерии определения плохо работающих маршрутов. Переключающийся маршрут получает за каждое переключение штраф, равный 1000. Когда общий штраф достигает заданного "предела подавления", выполняется подавление объявления маршрута. Штраф уменьшается экспоненциально на основании заданного "времени уменьшения наполовину". Когда общий штраф становится меньше

заданного "предела повторного использования", выполняется отмена подавления объявления маршрута.

Отключение не применяется к маршрутам, которые являются для AS внешними и изучаются с использованием iBGP. При этом отключение маршрутов позволяет избежать большого штрафа для равноправных узлов iBGP для маршрутов, которые являются для AS внешними.

Штраф уменьшается каждые 5 секунд. Отмена подавления маршрутов выполняется каждые 10 секунд. Маршрутизатор хранит информацию об отключении до тех пор, пока штраф не станет меньше половины "предела повторного использования". В этот момент маршрутизатор удаляет информацию.

По умолчанию функция отключения маршрутов отключена. При необходимости в будущем можно сделать так, чтобы по умолчанию данная функция была включена. Отключение маршрутов контролируется следующими командами:

**bgp dampening**—включает отключение.

**no bgp dampening**—выключает отключение.

*bgp dampening half-life-time*изменяет время уменьшения наполовину.

Следующая команда устанавливает все параметры одновременно:

*bgp dampening half-life-time reuse suppress maximum-suppress-time*

Описание синтаксиса:

*half-life-time*—диапазон составляет 145 минут. По умолчанию установлено значение 15 минут.

*reuse-value*диапазон составляет 1–20 000. По умолчанию установлено значение 750.

*suppress-value*диапазон составляет 1–20 000. По умолчанию установлено значение 2000.

*max-suppress-time* — Это - максимальный срок действия для подавления маршрута. Диапазон составляет 1–255 минут. По умолчанию в 4 раза больше времени уменьшения наполовину.

RTB#

hostname RTB

interface Serial0

ip address 203.250.15.2 255.255.255.252

```
interface Serial1
 ip address 192.208.10.6 255.255.255.252
```

```
router bgp 100
 bgp dampening
 network 203.250.15.0
 neighbor 192.208.10.5 remote-as 300
```

```
RTD#
 hostname RTD
```

```
interface Loopback0
 ip address 192.208.10.174 255.255.255.192
```

```
interface Serial0/0
 ip address 192.208.10.5 255.255.255.252
```

```
router bgp 300
 network 192.208.10.0
 neighbor 192.208.10.6 remote-as 100
```

Конфигурация RTB для отключения маршрутов с параметрами, установленными в значения по умолчанию. Если предположить, что соединение eBGP с RTD является стабильным, таблица BGP RTB выглядит следующим образом:

```
RTB# show ip bgp BGP table version is 24, local router ID is 203.250.15.2 Status codes: s
suppressed, d damped, h history, * valid, > best, i - internal Origin codes: i - IGP, e - EGP, ?
- incomplete Network Next Hop Metric LocPrf Weight Path *> 192.208.10.0 192.208.10.5 0 0 300 i
*> 203.250.15.0 0.0.0.0 0 32768 i
```

**Для имитации переключения маршрута в RTD вводится команда `clear ip bgp 192.208.10.6`. Таблица BGP RTB выглядит следующим образом:**

```
RTB# show ip bgp BGP table version is 24, local router ID is 203.250.15.2 Status codes: s
suppressed, d damped, h history, * valid, > best, i - internal Origin codes: i - IGP, e - EGP, ?
- incomplete Network Next Hop Metric LocPrf Weight Path h 192.208.10.0 192.208.10.5 0 0 300 i *>
203.250.15.0 0.0.0.0 0 32768 i
```

BGP 192.208.10.0 history. Данное состояние означает, что лучший путь до маршрута отсутствует, но при этом есть информация о переключении маршрута.

```
RTB# show ip bgp 192.208.10.0 BGP routing table entry for 192.208.10.0 255.255.255.0, version 25
Paths: (1 available, no best path) 300 (history entry) 192.208.10.5 from 192.208.10.5
(192.208.10.174) Origin IGP, metric 0, external Dampinfo: penalty 910, flapped 1 times in
0:02:03
```

Маршрут получил штраф за переключение, но штраф меньше "предела подавления". Значение по умолчанию – 2000. Подавление маршрута еще не выполнялось. Если маршрут переключается еще несколько раз, увидим следующее:

```
RTB# show ip bgp BGP table version is 32, local router ID is 203.250.15.2 Status codes: s
```

```
suppressed, d damped, h history, * valid, > best, i - internal Origin codes: i - IGP, e - EGP, ?  
- incomplete Network Next Hop Metric LocPrf Weight Path *d 192.208.10.0 192.208.10.5 0 0 300 i  
*> 203.250.15.0 0.0.0.0 0 32768 i RTB# show ip bgp 192.208.10.0 BGP routing table entry for  
192.208.10.0 255.255.255.0, version 32 Paths: (1 available, no best path) 300, (suppressed due  
to dampening) 192.208.10.5 from 192.208.10.5 (192.208.10.174) Origin IGP, metric 0, valid,  
external Dampinfo: penalty 2615, flapped 3 times in 0:05:18 , reuse in 0:27:00
```

Маршрут был отключен или подавлен. Маршрут начнет использоваться вновь после того, как значение штрафа достигнет "значения повторного использования". В данном случае значение повторного использования равно 750, значению по умолчанию. Информация об отключении удаляется, когда штраф становится меньше половины предела повторного использования. В данном случае удаление происходит когда штраф достигает значения 375 ( $750/2=375$ ). Следующие команды отображают и удаляют статистическую информацию по переключениям:

**show ip bgp flap-statistics**—отображает статистику переключений для всех путей.

*show ip bgp flap-statistics regex regular-expression*—отображает статистику переключений для всех маршрутов, соответствующих регулярному выражению.

*show ip bgp flap-statistics filter-list list*—отображает статистику переключений для всех путей, проходящих через фильтр.

*show ip bgp flap-statistics A.B.C.D m.m.m.m*—отображает статистику переключений для одной записи.

*show ip bgp flap-statistics A.B.C.D m.m.m.m longer-prefix*—отображает статистику переключений для некоторого количества определенных записей.

**show ip bgp neighbor [dampened-routes] | [flap-statistics]** —отображает статистику переключений для всех путей от соседа.

**clear ip bgp flap-statistics** —удаляет статистику переключений для всех маршрутов.

*clear ip bgp flap-statistics regex regular-expression*—удаляет статистику переключений для всех маршрутов, соответствующих регулярному выражению.

*clear ip bgp flap-statistics filter-list list*—удаляет статистику переключений для всех путей, проходящих через фильтр.

*clear ip bgp flap-statistics A.B.C.D m.m.m.m*—удаляет статистику переключений для одной записи.

*clear ip bgp A.B.C.D flap-statistics*—удаляет статистику переключений для всех путей от соседа.



## Выбор пути BGP

[После ознакомления с атрибутами и терминологией BGP см. статью Алгоритм выбора наилучшего пути BGP.](#)

## Примеры практического применения BGP 5

### Практический пример построения

В данном разделе описан пример построения, который отображает конфигурацию и таблицы маршрутизации так, как они отображаются в маршрутизаторах Cisco.

В данном разделе описывается способ пошагового создания такой конфигурации и проблемы, которые могут возникать в процессе создания. При наличии AS, которая подключается к двум ISP с использованием eBGP, всегда следует запускать iBGP в своей AS, что позволит лучше контролировать свои маршруты. В данном примере iBGP работает в AS100 между RTA и RTB, а в качестве IGP используется OSPF. Предположим, что есть подключение к двум ISP, AS200 и AS300. Вот первый шаг настройки для всех маршрутизаторов:

**Примечание:** Эти настройки не являются окончательными.

```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 203.250.13.41 255.255.255.0

interface Ethernet0
 ip address 203.250.14.1 255.255.255.0

interface Serial0
 ip address 128.213.63.1 255.255.255.252

router ospf 10
 network 203.250.0.0 0.0.255.255 area 0

router bgp 100
 network 203.250.13.0
 network 203.250.14.0
 neighbor 128.213.63.2 remote-as 200
 neighbor 203.250.15.2 remote-as 100
 neighbor 203.250.15.2 update-source Loopback0

RTF#
hostname RTF

ip subnet-zero

interface Ethernet0
 ip address 203.250.14.2 255.255.255.0
```

```
interface Serial1
 ip address 203.250.15.1 255.255.255.252
```

```
router ospf 10
 network 203.250.0.0 0.0.255.255 area 0
```

```
RTB#
hostname RTB
```

```
ip subnet-zero
```

```
interface Serial0
 ip address 203.250.15.2 255.255.255.252
```

```
interface Serial1
 ip address 192.208.10.6 255.255.255.252
```

```
router ospf 10
 network 203.250.0.0 0.0.255.255 area 0
```

```
router bgp 100
 network 203.250.15.0
 neighbor 192.208.10.5 remote-as 300
 neighbor 203.250.13.41 remote-as 100
```

```
RTC#
hostname RTC
```

```
ip subnet-zero
```

```
interface Loopback0
 ip address 128.213.63.130 255.255.255.192
```

```
interface Serial2/0
 ip address 128.213.63.5 255.255.255.252
```

```
!
```

```
interface Serial2/1
 ip address 128.213.63.2 255.255.255.252
```

```
router bgp 200
 network 128.213.0.0
 neighbor 128.213.63.1 remote-as 100
 neighbor 128.213.63.6 remote-as 400
```

```
RTD#
hostname RTD
```

```
ip subnet-zero
```

```
interface Loopback0
 ip address 192.208.10.174 255.255.255.192
```

```
interface Serial0/0
 ip address 192.208.10.5 255.255.255.252
!
interface Serial0/1
 ip address 192.208.10.2 255.255.255.252
```

```
router bgp 300
 network 192.208.10.0
 neighbor 192.208.10.1 remote-as 500
 neighbor 192.208.10.6 remote-as 100
```

```
RTE#
hostname RTE
```

```
ip subnet-zero
```

```
interface Loopback0
 ip address 200.200.10.1 255.255.255.0
```

```
interface Serial0
 ip address 195.211.10.2 255.255.255.252
```

```
interface Serial1
 ip address 128.213.63.6 255.255.255.252
 clockrate 1000000
```

```
router bgp 400
 network 200.200.10.0
 neighbor 128.213.63.5 remote-as 200
 neighbor 195.211.10.1 remote-as 500
```

```
RTG#
hostname RTG
```

```
ip subnet-zero
```

```
interface Loopback0
 ip address 195.211.10.174 255.255.255.192
```

```
interface Serial0
 ip address 192.208.10.1 255.255.255.252
```

```
interface Serial1
 ip address 195.211.10.1 255.255.255.252
```

```
router bgp 500
 network 195.211.10.0
 neighbor 192.208.10.2 remote-as 300
 neighbor 195.211.10.2 remote-as 400
```

**Для объявления сетей всегда следует пользоваться командой network или перераспределять статические записи в BGP. Этот способ является лучшим по сравнению с перераспределением IGP в BGP. В данном примере для добавления сетей в BGP используется команда network.**

Теперь переходим к отключению интерфейса s1 в RTB, как если бы соединения между RTB и RTD не существовало. Ниже приведена таблица BGP RTB:

```
RTB# show ip bgp BGP table version is 4, local router ID is 203.250.15.2 Status codes: s
suppressed, d damped, h history, * valid, > best, i - internal Origin codes: i - IGP, e - EGP, ?
- incomplete Network Next Hop Metric LocPrf Weight Path *i128.213.0.0 128.213.63.2 0 100 0 200 i
*i192.208.10.0 128.213.63.2 100 0 200 400 500 300 i *i195.211.10.0 128.213.63.2 100 0 200 400
500 i *i200.200.10.0 128.213.63.2 100 0 200 400 i *>i203.250.13.0 203.250.13.41 0 100 0 i
*>i203.250.14.0 203.250.13.41 0 100 0 i *>203.250.15.0 0.0.0.0 0 32768 i
```

В данной таблице используются следующие обозначения:

I - , iBGP.

I - , IGP.

Path- . Например, сеть 128.213.0.0 изучается по пути 200 со следующим переходом 128.213.63.2.

**Примечание:** У любой локально созданной записи как, например, 203.250.15.0, следующим переходом будет 0.0.0.0.

Символ > — Указывает, что BGP выбрал лучший маршрут. [При принятии решения BGP действует по алгоритму, описанному в статье Алгоритм выбора наилучшего пути BGP.](#) BGP берет один наилучший путь до сети назначения, устанавливает путь в таблицу маршрутизации IP и объявляет путь другим равноправным узлам BGP.

**Примечание:** Next Hop. RTB знает о 128.213.0.0 через следующий переход 128.213.63.2, который является следующим переходом eBGP, сохраняемым в iBGP.

Таблица маршрутизации IP:

```
RTB# show ip route Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP D -
EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area E1 - OSPF external type 1, E2 - OSPF
external type 2, E - EGP i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default Gateway of last resort is not set 203.250.13.0 255.255.255.255 is subnetted, 1 subnets O
203.250.13.41 [110/75] via 203.250.15.1, 02:50:45, Serial0 203.250.15.0 255.255.255.252 is
subnetted, 1 subnets C 203.250.15.0 is directly connected, Serial0 O 203.250.14.0 [110/74] via
203.250.15.1, 02:50:46, Serial0
```

Очевидно, что ни одна из записей BGP не попала в таблицу маршрутов. Здесь есть две проблемы.

Первая проблема заключается в том, что следующий переход для этих записей, 128.213.63.2, является недоступным. Отсутствует способ доступа по следующему переходу с использованием IGP, функции которого выполняет OSPF. RTB не изучил от OSPF информацию о 128.213.63.0. Можно запустить OSPF в интерфейсе s0 RTA и сделать его пассивным. При этом RTB будет знать, как получить доступ к следующему переходу 128.213.63.2. Ниже приведена данная конфигурация RTA:

RTA#

```
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 203.250.13.41 255.255.255.0

interface Ethernet0
 ip address 203.250.14.1 255.255.255.0

interface Serial0
 ip address 128.213.63.1 255.255.255.252

router ospf 10
 passive-interface Serial0
 network 203.250.0.0 0.0.255.255 area 0
 network 128.213.0.0 0.0.255.255 area 0
```

```
router bgp 100
 network 203.250.0.0 mask 255.255.0.0
 neighbor 128.213.63.2 remote-as 200
 neighbor 203.250.15.2 remote-as 100
 neighbor 203.250.15.2 update-source Loopback0
```

**Примечание: Чтобы изменить следующий переход, можно ввести команду `bgp nexthopself` между RTA и RTB.**

Новая таблица BGP в RTB выглядит следующим образом:

```
RTB# show ip bgp BGP table version is 10, local router ID is 203.250.15.2 Status codes: s
suppressed, d damped, h history, * valid, > best, i - internal Origin codes: i - IGP, e - EGP, ?
- incomplete Network Next Hop Metric LocPrf Weight Path *>i128.213.0.0 128.213.63.2 0 100 0 200
i *>i192.208.10.0 128.213.63.2 100 0 200 400 500 300 i *>i195.211.10.0 128.213.63.2 100 0 200
400 500 i *>i200.200.10.0 128.213.63.2 100 0 200 400 i *>i203.250.13.0 203.250.13.41 0 100 0 i
*>i203.250.14.0 203.250.13.41 0 100 0 i *> 203.250.15.0 0.0.0.0 0 32768 i
```

**Примечание: Все записи имеют >, что означает, что BGP может достигнуть следующего перехода.**

Таблица маршрутизации:

```
RTB# show ip route Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP D -
EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area E1 - OSPF external type 1, E2 - OSPF
external type 2, E - EGP i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default Gateway of last resort is not set 203.250.13.0 255.255.255.255 is subnetted, 1 subnets O
203.250.13.41 [110/75] via 203.250.15.1, 00:04:46, Serial0 203.250.15.0 255.255.255.252 is
subnetted, 1 subnets C 203.250.15.0 is directly connected, Serial0 O 203.250.14.0 [110/74] via
203.250.15.1, 00:04:46, Serial0 128.213.0.0 255.255.255.252 is subnetted, 1 subnets O
128.213.63.0 [110/138] via 203.250.15.1, 00:04:47, Serial0
```

Вторая проблема заключается в том, что записи BGP до сих пор отсутствуют в таблице маршрутов. Единственное отличие лишь в том, что 128.213.63.0 теперь доступен через OSPF. Проблема в данном случае кроется в синхронизации. BGP не помещает эти записи в таблицу маршрутизации и не передает записи в обновлениях BGP из-за отсутствия синхронизации с IGP.

**Примечание:** У RTF не указаны сети 192.208.10.0 и 195.211.10.0, поскольку BGP еще не был перераспределен в OSPF.

В данном случае при отключении синхронизации записи появляются в таблице маршрутов. Однако связь все еще отсутствует.

При отключении синхронизации в RTB произойдет следующее:

```
RTB# show ip route Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP D -
EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area E1 - OSPF external type 1, E2 - OSPF
external type 2, E - EGP i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default Gateway of last resort is not set B 200.200.10.0 [200/0] via 128.213.63.2, 00:01:07 B
195.211.10.0 [200/0] via 128.213.63.2, 00:01:07 B 192.208.10.0 [200/0] via 128.213.63.2,
00:01:07 203.250.13.0 is variably subnetted, 2 subnets, 2 masks O 203.250.13.41 255.255.255.255
[110/75] via 203.250.15.1, 00:12:37, Serial0 B 203.250.13.0 255.255.255.0 [200/0] via
203.250.13.41, 00:01:08 203.250.15.0 255.255.255.252 is subnetted, 1 subnets C 203.250.15.0 is
directly connected, Serial0 O 203.250.14.0 [110/74] via 203.250.15.1, 00:12:37, Serial0
128.213.0.0 is variably subnetted, 2 subnets, 2 masks B 128.213.0.0 255.255.0.0 [200/0] via
128.213.63.2, 00:01:08 O 128.213.63.0 255.255.255.252 [110/138] via 203.250.15.1, 00:12:37,
Serial0
```

Таблица маршрутов выглядит прекрасно, однако эти сети являются недостижимыми. RTF в центре не знает, как достигнуть сетей:

```
RTF# show ip route Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP D -
EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area E1 - OSPF external type 1, E2 - OSPF
external type 2, E - EGP i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default Gateway of last resort is not set 203.250.13.0 255.255.255.255 is subnetted, 1 subnets O
203.250.13.41 [110/11] via 203.250.14.1, 00:14:15, Ethernet0 203.250.15.0 255.255.255.252 is
subnetted, 1 subnets C 203.250.15.0 is directly connected, Serial1 C 203.250.14.0 is directly
connected, Ethernet0 128.213.0.0 255.255.255.252 is subnetted, 1 subnets O 128.213.63.0 [110/74]
via 203.250.14.1, 00:14:15, Ethernet0
```

В данном случае при отключении синхронизации проблема сохраняется. Однако синхронизация понадобится в последующем для решения других вопросов. Необходимо выполнить перераспределение BGP в OSPF на RTA с метрикой 2000:

```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 203.250.13.41 255.255.255.0

interface Ethernet0
 ip address 203.250.14.1 255.255.255.0

interface Serial0
 ip address 128.213.63.1 255.255.255.252

router ospf 10
 redistribute bgp 100 metric 2000 subnets
 passive-interface Serial0
```

```
network 203.250.0.0 0.0.255.255 area 0
```

```
network 128.213.0.0 0.0.255.255 area 0
```

```
router bgp 100
```

```
network 203.250.0.0 mask 255.255.0.0
```

```
neighbor 128.213.63.2 remote-as 200
```

```
neighbor 203.250.15.2 remote-as 100
```

```
neighbor 203.250.15.2 update-source Loopback0
```

Таблица маршрутизации выглядит следующим образом:

```
RTB# show ip route Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP D -
EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area E1 - OSPF external type 1, E2 - OSPF
external type 2, E - EGP i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default Gateway of last resort is not set O E2 200.200.10.0 [110/2000] via 203.250.15.1,
00:00:14, Serial0 O E2 195.211.10.0 [110/2000] via 203.250.15.1, 00:00:14, Serial0 O E2
192.208.10.0 [110/2000] via 203.250.15.1, 00:00:14, Serial0 203.250.13.0 is variably subnetted,
2 subnets, 2 masks O 203.250.13.41 255.255.255.255 [110/75] via 203.250.15.1, 00:00:15, Serial0
O E2 203.250.13.0 255.255.255.0 [110/2000] via 203.250.15.1, 00:00:15, Serial0 203.250.15.0
255.255.255.252 is subnetted, 2 subnets C 203.250.15.8 is directly connected, Loopback1 C
203.250.15.0 is directly connected, Serial0 O 203.250.14.0 [110/74] via 203.250.15.1, 00:00:15,
Serial0 128.213.0.0 is variably subnetted, 2 subnets, 2 masks O E2 128.213.0.0 255.255.0.0
[110/2000] via 203.250.15.1, 00:00:15,Serial0 O 128.213.63.0 255.255.255.252 [110/138] via
203.250.15.1, 00:00:16, Serial0
```

Записи BGP исчезли, поскольку у OSPF расстояние лучше, чем у iBGP. Расстояние OSPF равно 110, а расстояние iBGP равно 200.

Необходимо отключить синхронизацию в RTA для того, чтобы RTA мог объявить 203.250.15.0. Данное действие является обязательным, поскольку RTA не выполняет синхронизацию с OSPF из-за различия в масках. Синхронизация в RTB должна быть отключена, чтобы RTB мог объявить 203.250.13.0. Это действие является обязательным в RTB по той же причине.

Теперь необходимо включить интерфейс s1 RTB и посмотреть, как будут выглядеть маршруты. Кроме того, необходимо включить OSPF на последовательном соединении 1 RTB, чтобы оно было пассивным. Данный шаг позволяет RTA узнать о следующем переходе 192.208.10.5 с помощью IGP. Если этот шаг пропустить, возникает петля маршрутизации, поскольку для достижения следующего перехода 192.208.10.5 необходимо пройти другим путем с помощью eBGP. Ниже приведено несколько настроек RTA и RTB:

```
RTA#
```

```
hostname RTA
```

```
ip subnet-zero
```

```
interface Loopback0
```

```
ip address 203.250.13.41 255.255.255.0
```

```
interface Ethernet0
```

```
ip address 203.250.14.1 255.255.255.0
```

```
interface Serial0
```

```
ip address 128.213.63.1 255.255.255.252
```

```

router ospf 10
 redistribute bgp 100 metric 2000 subnets
 passive-interface Serial0
 network 203.250.0.0 0.0.255.255 area 0
 network 128.213.0.0 0.0.255.255 area 0

router bgp 100
 no synchronization
 network 203.250.13.0
 network 203.250.14.0
 neighbor 128.213.63.2 remote-as 200
 neighbor 203.250.15.2 remote-as 100
 neighbor 203.250.15.2 update-source Loopback0

```

```

RTB#
hostname RTB

```

```

ip subnet-zero

interface Serial0
 ip address 203.250.15.2 255.255.255.252

interface Serial1
 ip address 192.208.10.6 255.255.255.252

```

```

router ospf 10
 redistribute bgp 100 metric 1000 subnets
 passive-interface Serial1
 network 203.250.0.0 0.0.255.255 area 0
 network 192.208.0.0 0.0.255.255 area 0

```

```

router bgp 100
 no synchronization
 network 203.250.15.0
 neighbor 192.208.10.5 remote-as 300
 neighbor 203.250.13.41 remote-as 100

```

**Таблицы BGP выглядят следующим образом:**

```

RTA# show ip bgp BGP table version is 117, local router ID is 203.250.13.41 Status codes: s
suppressed, d damped, h history, * valid, > best, i -internal Origin codes: i - IGP, e - EGP, ?
- incomplete Network Next Hop Metric LocPrf Weight Path *> 128.213.0.0 128.213.63.2 0 0 200 i
*>i192.208.10.0 192.208.10.5 0 100 0 300 i *>i195.211.10.0 192.208.10.5 100 0 300 500 i *
128.213.63.2 0 200 400 500 i *> 200.200.10.0 128.213.63.2 0 200 400 i *> 203.250.13.0 0.0.0.0 0
32768 i *> 203.250.14.0 0.0.0.0 0 32768 i *>i203.250.15.0 203.250.15.2 0 100 0 i RTB# show ip
bgp BGP table version is 12, local router ID is 203.250.15.10 Status codes: s suppressed, d
damped, h history, * valid, > best, i -internal Origin codes: i - IGP, e - EGP, ? - incomplete
Network Next Hop Metric LocPrf Weight Path *>i128.213.0.0 128.213.63.2 0 100 0 200 i *
192.208.10.5 0 300 500 400 200 i *> 192.208.10.0 192.208.10.5 0 0 300 i *> 195.211.10.0
192.208.10.5 0 300 500 i *>i200.200.10.0 128.213.63.2 100 0 200 400 i * 192.208.10.5 0 300 500
400 i *>i203.250.13.0 203.250.13.41 0 100 0 i *>i203.250.14.0 203.250.13.41 0 100 0 i *>
203.250.15.0 0.0.0.0 0 32768 i

```

**Есть несколько способов построения сети для обмена информацией с двумя различными**



ISP, AS200 и AS300. Одним способом является организация основного ISP и резервного ISP. Можно изучить частичные маршруты одного из ISP и маршруты по умолчанию к обоим ISP. В данном примере происходит получение частичных маршрутов от AS200 и только локальных маршрутов от AS300. И RTA, и RTB создают маршруты по умолчанию в OSPF, при этом RTB является предпочтительным, поскольку обладает метрикой с меньшим значением. Таким образом можно распределять исходящий трафик между двумя ISP.

Возможно асимметричное распределение, если трафик, выходящий из RTA, возвращается через RTB. Это может возникнуть при использовании одного пула IP-адресов, одной главной сети при обмене информацией с двумя ISP. В результате агрегирования вся AS может выглядеть для внешнего мира одним большим объектом. Точками доступа к сети могут выступать RTA или RTB. Можно обнаружить, что весь входящий в AS трафик приходит через одну точку даже при наличии нескольких точек доступа в Интернет. В примере при обмене информацией с двумя ISP присутствуют две разных больших сети.

Другой потенциальной причиной асимметрии может быть различная длина объявленных путей доступа к AS. Возможно, один поставщик услуг находится ближе к определенному месту назначения, чем другой. В примере трафик из AS400, для которой ваша сеть является сетью назначения, всегда поступает через RTA, по причине более короткого пути. Можно попытаться повлиять на это решение. **Чтобы добавить в обновления номера путей и длина пути стала казаться больше, можно воспользоваться командой set as-path prepend.** А с помощью атрибутов, таких, как локальное предпочтение, метрика или вес, AS400 может установить в качестве точки выхода AS200. В этом случае ничего сделать нельзя.

Данная настройка является последней для всех маршрутизаторов:

```
RTA#
hostname RTA

ip subnet-zero

interface Loopback0
 ip address 203.250.13.41 255.255.255.0

interface Ethernet0
 ip address 203.250.14.1 255.255.255.0

interface Serial0
 ip address 128.213.63.1 255.255.255.252

router ospf 10
 redistribute bgp 100 metric 2000 subnets
 passive-interface Serial0
 network 203.250.0.0 0.0.255.255 area 0
 network 128.213.0.0 0.0.255.255 area 0
 default-information originate metric 2000

router bgp 100
 no synchronization
 network 203.250.13.0
 network 203.250.14.0
 neighbor 128.213.63.2 remote-as 200
 neighbor 128.213.63.2 route-map setlocalpref in
 neighbor 203.250.15.2 remote-as 100
```

```
neighbor 203.250.15.2 update-source Loopback0
```

```
ip classless
```

```
ip default-network 200.200.0.0
```

```
route-map setlocalpref permit 10
```

```
set local-preference 200
```

В RTA локальное предпочтение маршрутов, поступающих из AS200, установлено в значение 200. Кроме того, сеть 200.200.0.0 выбирается кандидатом в сети по умолчанию. **Команда ip default-network позволяет выбрать сеть по умолчанию.**

Кроме того, в данном примере использование команды default-information originate с OSPF добавляет маршрут по умолчанию в домен OSPF. В данном примере эта команда также

используется с протоколом связи промежуточных систем (протокол IS-IS) и BGP. Для RIP выполняется автоматическое перераспределение 0.0.0.0 в RIP без какой-либо дополнительной настройки. Для IGRP и EIGRP добавление информации по умолчанию в домен IGP выполняется после перераспределения BGP в IGRP и в EIGRP. Кроме того, с использованием IGRP и EIGRP можно перераспределить статический маршрут до 0.0.0.0 в домен IGP.

```
RTF#
```

```
hostname RTF
```

```
ip subnet-zero
```

```
interface Ethernet0
```

```
ip address 203.250.14.2 255.255.255.0
```

```
interface Serial1
```

```
ip address 203.250.15.1 255.255.255.252
```

```
router ospf 10
```

```
network 203.250.0.0 0.0.255.255 area 0
```

```
ip classless
```

```
RTB#
```

```
hostname RTB
```

```
ip subnet-zero
```

```
interface Loopback1
```

```
ip address 203.250.15.10 255.255.255.252
```

```
interface Serial0
```

```
ip address 203.250.15.2 255.255.255.252
```

```
!
```

```
interface Serial1
```

```
ip address 192.208.10.6 255.255.255.252
```

```
router ospf 10
```

```
redistribute bgp 100 metric 1000 subnets
```

```
passive-interface Serial1
```

```

network 203.250.0.0 0.0.255.255 area 0
network 192.208.10.6 0.0.0.0 area 0
default-information originate metric 1000
!
router bgp 100
no synchronization
network 203.250.15.0
neighbor 192.208.10.5 remote-as 300
neighbor 192.208.10.5 route-map localonly in
neighbor 203.250.13.41 remote-as 100
!
ip classless
ip default-network 192.208.10.0
ip as-path access-list 1 permit ^300$

```

```

route-map localonly permit 10
match as-path 1
set local-preference 300

```

Для RTB локальное предпочтение обновлений, поступающих из AS300, установлено в значение 300. Данное значение больше, чем локальное предпочтение обновлений iBGP, поступающих из RTA. Таким образом, AS100 выбирает RTB для локальных маршрутов AS300. Любые другие маршруты в RTB при их наличии, передаются внутри с локальным предпочтением 100. Данное значение меньше, чем локальное предпочтение 200 обновлений, поступающих из RTA. Поэтому предпочтительным является RTA.

**Примечание:** Были объявлены только локальные маршруты AS300. , ^300\$, . , ISP, ^300\_[0-9]\*.

Ниже приведены выходные данные регулярного выражения, которое указывает на локальные маршруты AS300:

```

RTB# show ip bgp regexp ^300$ BGP table version is 14, local router ID is 203.250.15.10 Status
codes: s suppressed, d damped, h history, * valid, > best, i - internal Origin codes: i - IGP, e
- EGP, ? - incomplete Network Next Hop Metric LocPrf Weight Path *> 192.208.10.0 192.208.10.5 0
300 0 300 RTC# hostname RTC ip subnet-zero interface Loopback0 ip address 128.213.63.130
255.255.255.192 interface Serial2/0 ip address 128.213.63.5 255.255.255.252 ! interface
Serial2/1 ip address 128.213.63.2 255.255.255.252 router bgp 200 network 128.213.0.0 neighbor
128.213.63.1 remote-as 100 neighbor 128.213.63.1 distribute-list 1 out neighbor 128.213.63.6
remote-as 400 ip classless access-list 1 deny 195.211.0.0 0.0.255.255 access-list 1 permit any

```

В RTC выполняется агрегирование 128.213.0.0/16 и указываются определенные маршруты для добавления в AS100. Если ISP откажется это сделать, необходимо будет выполнять фильтрацию на входящем конце AS100.

```

RTD#
hostname RTD

ip subnet-zero

interface Loopback0
ip address 192.208.10.174 255.255.255.192
!
interface Serial0/0
ip address 192.208.10.5 255.255.255.252

```

```

!
interface Serial0/1
 ip address 192.208.10.2 255.255.255.252

router bgp 300
 network 192.208.10.0
 neighbor 192.208.10.1 remote-as 500
 neighbor 192.208.10.6 remote-as 100

RTG#
hostname RTG

ip subnet-zero

interface Loopback0
 ip address 195.211.10.174 255.255.255.192

interface Serial0
 ip address 192.208.10.1 255.255.255.252

interface Serial1
 ip address 195.211.10.1 255.255.255.252

router bgp 500
 network 195.211.10.0
 aggregate-address 195.211.0.0 255.255.0.0 summary-only
 neighbor 192.208.10.2 remote-as 300
 neighbor 192.208.10.2 send-community
 neighbor 192.208.10.2 route-map setcommunity out
 neighbor 195.211.10.2 remote-as 400
!
ip classless
access-list 1 permit 195.211.0.0 0.0.255.255
access-list 2 permit any
route-map setcommunity permit 20
 match ip address 2
!
route-map setcommunity permit 10
 match ip address 1
 set community no-export

```

**Демонстрация использования фильтрации по сообществам в RTG. Добавляется сообщество no-export для обновлений 195.211.0.0 в направлении RTD.** Таким образом, RTD не экспортирует этот маршрут RTB. Однако в этом случае RTB все равно не принимает эти маршруты.

```

RTE#
hostname RTE

ip subnet-zero

interface Loopback0
 ip address 200.200.10.1 255.255.255.0

```

```

interface Serial0
 ip address 195.211.10.2 255.255.255.252

interface Serial1
 ip address 128.213.63.6 255.255.255.252

router bgp 400
 network 200.200.10.0
 aggregate-address 200.200.0.0 255.255.0.0 summary-only
 neighbor 128.213.63.5 remote-as 200
 neighbor 195.211.10.1 remote-as 500

```

```
ip classless
```

**RTE выполняет агрегирование 200.200.0.0/16. Ниже приведены итоговые таблицы BGP и таблицы маршрутизации для RTA, RTF и RTB:**

```

RTA# show ip bgp BGP table version is 21, local router ID is 203.250.13.41 Status codes: s
suppressed, d damped, h history, * valid, > best, i - internal Origin codes: i - IGP, e - EGP, ?
- incomplete Network Next Hop Metric LocPrf Weight Path *> 128.213.0.0 128.213.63.2 0 200 0 200
i *>i192.208.10.0 192.208.10.5 0 300 0 300 i *> 200.200.0.0/16 128.213.63.2 200 0 200 400 i *>
203.250.13.0 0.0.0.0 0 32768 i *> 203.250.14.0 0.0.0.0 0 32768 i *>i203.250.15.0 203.250.15.2 0
100 0 i RTA# show ip route Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -
BGP D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area E1 - OSPF external type 1, E2
- OSPF external type 2, E - EGP i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default Gateway of last resort is 128.213.63.2 to network 200.200.0.0 192.208.10.0 is variably
subnetted, 2 subnets, 2 masks O E2 192.208.10.0 255.255.255.0 [110/1000] via 203.250.14.2,
00:41:25, Ethernet0 O 192.208.10.4 255.255.255.252 [110/138] via 203.250.14.2, 00:41:25,
Ethernet0 C 203.250.13.0 is directly connected, Loopback0 203.250.15.0 is variably subnetted, 3
subnets, 3 masks O 203.250.15.10 255.255.255.255 [110/75] via 203.250.14.2, 00:41:25, Ethernet0
O 203.250.15.0 255.255.255.252 [110/74] via 203.250.14.2, 00:41:25, Ethernet0 B 203.250.15.0
255.255.255.0 [200/0] via 203.250.15.2, 00:41:25 C 203.250.14.0 is directly connected, Ethernet0
128.213.0.0 is variably subnetted, 2 subnets, 2 masks B 128.213.0.0 255.255.0.0 [20/0] via
128.213.63.2, 00:41:26 C 128.213.63.0 255.255.255.252 is directly connected, Serial0 O*E2
0.0.0.0/0 [110/1000] via 203.250.14.2, Ethernet0/O B* 200.200.0.0 255.255.0.0 [20/0] via
128.213.63.2, 00:02:38 RTF# show ip route Codes: C - connected, S - static, I - IGRP, R - RIP, M
- mobile, B - BGP D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area E1 - OSPF
external type 1, E2 - OSPF external type 2, E - EGP i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS
level-2, * - candidate default Gateway of last resort is 203.250.15.2 to network 0.0.0.0
192.208.10.0 is variably subnetted, 2 subnets, 2 masks O E2 192.208.10.0 255.255.255.0
[110/1000] via 203.250.15.2, 00:48:50, Serial1 O 192.208.10.4 255.255.255.252 [110/128] via
203.250.15.2, 01:12:09, Serial1 203.250.13.0 is variably subnetted, 2 subnets, 2 masks O
203.250.13.41 255.255.255.255 [110/11] via 203.250.14.1, 01:12:09, Ethernet0 O E2 203.250.13.0
255.255.255.0 [110/2000] via 203.250.14.1, 01:12:09, Ethernet0 203.250.15.0 is variably
subnetted, 2 subnets, 2 masks O 203.250.15.10 255.255.255.255 [110/65] via 203.250.15.2,
01:12:09, Serial1 C 203.250.15.0 255.255.255.252 is directly connected, Serial1 C 203.250.14.0
is directly connected, Ethernet0 128.213.0.0 is variably subnetted, 2 subnets, 2 masks O E2
128.213.0.0 255.255.0.0 [110/2000] via 203.250.14.1, 00:45:01, Ethernet0 O 128.213.63.0
255.255.255.252 [110/74] via 203.250.14.1, 01:12:11, Ethernet0 O E2 200.200.0.0 255.255.0.0
[110/2000] via 203.250.14.1, 00:03:47, Ethernet0 O*E2 0.0.0.0 0.0.0.0 [110/1000] via
203.250.15.2, 00:03:33, Serial1

```

**Примечание:** Таблица RTF указывает на то, что маршрут до сетей, локальных для AS300, например 192.208.10.0, проходит через RTB. Доступ к другим известным сетям, например,

200.200.0.0, осуществляется через RTA. Шлюз последнего маршрута назначен RTB. Если что-либо случится с соединением между RTB и RTD, то, что RTA объявит по умолчанию, добавляется с метрикой 2000.

```
RTB# show ip bgp BGP table version is 14, local router ID is 203.250.15.10 Status codes: s
suppressed, d damped, h history, * valid, > best, i - internal Origin codes: i - IGP, e - EGP, ?
- incomplete Network Next Hop Metric LocPrf Weight Path *>i128.213.0.0 128.213.63.2 0 200 0 200
i *> 192.208.10.0 192.208.10.5 0 300 0 300 i *>i200.200.0.0/16 128.213.63.2 200 0 200 400 i
*>i203.250.13.0 203.250.13.41 0 100 0 i *>i203.250.14.0 203.250.13.41 0 100 0 i *> 203.250.15.0
0.0.0.0 0 32768 i RTB# show ip route Codes: C - connected, S - static, I - IGRP, R - RIP, M -
mobile, B - BGP D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area E1 - OSPF
external type 1, E2 - OSPF external type 2, E - EGP i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS
level-2, * - candidate default Gateway of last resort is 192.208.10.5 to network 192.208.10.0 *
192.208.10.0 is variably subnetted, 2 subnets, 2 masks B* 192.208.10.0 255.255.255.0 [20/0] via
192.208.10.5, 00:50:46 C 192.208.10.4 255.255.255.252 is directly connected, Serial1
203.250.13.0 is variably subnetted, 2 subnets, 2 masks O 203.250.13.41 255.255.255.255 [110/75]
via 203.250.15.1, 01:20:33, Serial0 O E2 203.250.13.0 255.255.255.0 [110/2000] via 203.250.15.1,
01:15:40, Serial0 203.250.15.0 255.255.255.252 is subnetted, 2 subnets C 203.250.15.8 is
directly connected, Loopback1 C 203.250.15.0 is directly connected, Serial0 O 203.250.14.0
[110/74] via 203.250.15.1, 01:20:33, Serial0 128.213.0.0 is variably subnetted, 2 subnets, 2
masks O E2 128.213.0.0 255.255.0.0 [110/2000] via 203.250.15.1, 00:46:55, Serial0 O 128.213.63.0
255.255.255.252 [110/138] via 203.250.15.1, 01:20:34, Serial0 O*E2 0.0.0.0/0 [110/2000] via
203.250.15.1, 00:08:33, Serial0 O E2 200.200.0.0 255.255.0.0 [110/2000] via 203.250.15.1,
00:05:42, Serial0
```

## Дополнительные сведения

- [BGP: часто задаваемые вопросы](#)
- [Примеры настроек BGP в брандмауэрах PIX](#)
- [Использование HSRP для обеспечения избыточности в сети с многоканальным протоколом BGP](#)
- [Настройка резервирования и BGP в режиме одиночного маршрутизатора на MSFC Cat6000](#)
- [Обеспечение оптимальной маршрутизации и сокращение объема памяти, используемой протоколом BGP](#)
- [Устранение неисправностей BGP](#)
- [Устранение проблем с перегрузкой CPU, вызываемой процессом сканера или маршрутизатора BGP](#)
- [Распределение нагрузки в одно- и многоканальной среде BGP: Примеры конфигураций](#)
- [Страница поддержки BGP](#)
- [Cisco Systems – техническая поддержка и документация](#)