

# BGP протокол (перевод на русский)

## Table of Contents

### [Использование BGP \(Border Gateway Protocol - протокол пограничного роутинга\) для междоменного роутинга](#)

#### **Основы BGP**

- [Внутридоменный \(Internal\) BGP](#)
  - [Интерфейсы Loopback](#)
- [Междоменный \(External\) BGP](#)
  - [Мультихоповый \(Multihop\) EBGP](#)
  - [EBGP и Load Balancing](#)
  - [Синхронизация](#)

#### **BGP и Route Map (роуто мепы)**

- [Анонсирование сетей](#)
  - [Ре-анонсирование \(Redistributing\) статических роутов](#)
  - [Ре-анонсирование \(Redistributing\) динамических роутов](#)
  - [Использование команды network](#)

#### **Алгоритм принятия решений BGP**

- [Атрибут AS\\_path](#)
- [Атрибут Origin](#)
- [Атрибут Next Hop](#)
  - [Атрибут Next Hop Attribute а мультидоступовая среда\(Multiaccess Media\)](#)
  - [Атрибут Next Hop и среда без бродкастов \(Nonbroadcast Media Access\)](#)
- [Атрибут вес \(weight\)](#)
  - [Использование листов доступа для установки веса](#)
  - [Использование роуте-мепов для установки веса](#)
  - [Использование команды neighbor weight для установки веса](#)
- [Атрибут Local Preference](#)
  - [Использование команды bgp default local-preference](#)
  - [Использование роуте-мепов для установки Local Preference](#)
- [Атрибут Multi-Exit Discriminator](#)
- [Атрибут Community](#)
- [Итоговая схема выбора пути в BGP](#)

#### **Контроль потоков BGP апдейтов**

- [Administrative Distance](#)
- [Фильтрация BGP](#)
  - [Фильтрация префиксов](#)
  - [Фильтрация AS\\_path](#)
  - [Фильтрация в Route Map](#)
  - [Фильтрация по Community](#)
- [Группы BGP Peer Groups](#)
- [CIDR и агрегатированные адреса](#)
  - [Агрегатирование и статический роутинг](#)
  - [Агрегатирование и AS-SET](#)
- [Конфидерации](#)
- [Рефлекторы](#)
  - [Использование Originator ID](#)
  - [Использование Cluster List](#)
  - [Роуте рефлекторы и традиционные BGP устройства](#)
- [Мигание BGP роутов - Route Flap Dampening](#)

#### **Практический пример дизайна**

- [Определение статуса BGP](#)
- [Исправление проблемы Next Hop](#)
- [Выключение синхронизации](#)
- [Ре-анонсирование OSPF](#)
- [Управление несимметрией](#)
- [Окончательная конфигурация](#)

#### **Подведение итогов**

---

# Using the Border Gateway Protocol for Interdomain Routing

## Использование BGP для настройки маршрутизации между доменами

*Это первая редакция, пока не будут исправлены ошибки, ее не рекомендуется использовать в учебных целях*

Border Gateway Protocol (BGP), описанный в RFC 1771, предназначен для маршрутизации между автономными системами. Этот протокол включает в себя защиту от "зацикливания". Автономная система [AS] - это набор роутеров, которые работают под управлением одного администратора, или одной группы администраторов, и используют общую стратегию маршрутизации. BGP часто используется в сетях между ISP (Internet Service Providers). В данном руководстве мы рассмотрим, как работает BGP и как Вы можете его использовать для маршрутизации между другими сетями, роутеры которых тоже используют BGP. Ниже рассматриваются следующие главы:

- [BGP Fundamentals](#)
- [BGP Decision Algorithm](#)
- [Controlling the Flow of BGP Updates](#)
- [Practical Design Example](#)

---

**Note** Мы будем рассматривать BGP Version 4.

---

## I. BGP Fundamentals

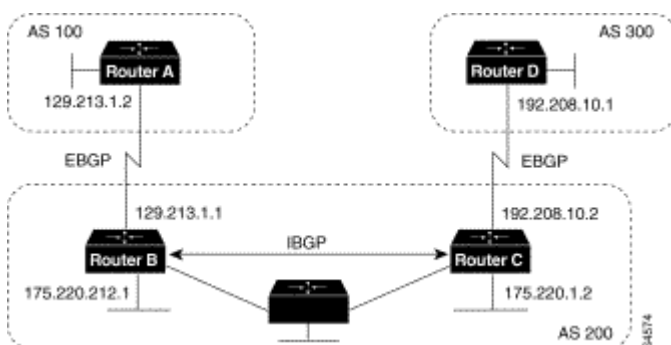
Эта глава представляет информацию о BGP, и включает в себя следующие разделы:

- [I.1 Internal BGP](#)
- [I.2 External BGP](#)
- [I.3 BGP and Route Maps](#)
- [I.4 Advertising Networks](#)

Роутеры, которые принадлежат одной и той же AS и обмениваются BGP update'ами, работают по Internal BGP (IBGP); роутеры, которые принадлежат разным AS и тоже обмениваются BGP update'ами, работают по External BGP (EBGP). Команды для конфигурации IBGP и EBGP одинаковы, исключая команду конфигурации роутера "neighbor ebgp-multihop" (которая описана в разделе "2" [External BGP](#)" этой главы). Поэтому везде, где мы будем упоминать о IBGP и EBGP, это будет делаться в контексте, что routing updates происходят либо в пределах одной AS, либо между AS.

[На рисунке 12-1](#) демонстрируется различие между EBGP и IBGP.

**Figure 12-1: EBGP, IBGP, and Multiple ASs**



Перед тем, как обмениваться информацией о маршрутах с внешними AS, BGP должен гарантировать, что сети \_внутри его\_ AS достижимы. Это обеспечивается комбинацией обмена информацией о маршрутах по IBGP между роутерами внутри AS и передачей информации о маршрутах BGP в один из Interior Gateway Protocols (IGPs), которые работают внутри AS (например, Interior Gateway Routing Protocol IGRP, Intermediate System-to-Intermediate System IS-IS, Routing Information Protocol RIP и Open Shortest Path First OSPF).

BGP использует TCP в качестве транспортного протокола (port 179). Любые два роутера, между которыми открыто tcp соединение для обмена информацией о маршрутизации, называются "peers" или "neighbors". На приведенном выше рисунке роутеры A и B являются BGP peers, равно как и роутеры B и C, C и D. Routing Information состоит из незакольцованной (loop-free) карты автономных систем. Заметим, что внутри AS BGP peers не являются подключенными непосредственно друг к другу (грубо говоря, не соединены между собой напрямую проводом).

BGP peers иницируют обмен полными таблицами BGP маршрутизации между собой. Позже они посылают лишь incremental updates. Кроме этого, BGP peers обмениваются keepalive messages (чтобы удостовериться, что связь между ними не потеряна), и notification messages (сообщениями об ошибках и другой служебной информацией).

Для простой конфигурации роутеров на приведенном выше рисунке используются следующие команды:

```
! Router A router bgp 100 neighbor 129.213.1.1 remote-as 200! Router B router bgp 200 neighbor 219.213.1.2 remote-as 100 neighbor 175.220.1.2 remote-as 200! Router C router bgp 200 neighbor 175.220.212.1 remote-as 200 neighbor 192.208.10.1 remote-as 300! Router D router bgp 300 neighbor 192.208.10.2 remote-as 200
```

Как видно из приведенной выше конфигурации, router bgp global configuration команды присваивают роутеру номер его AS и включают обмен BGP маршрутами между автономными системами.

Команда конфигурации **neighbor [ip address] remote-as [AS]** добавляет запись в таблицу маршрутизации BGP, и в этой записи говорится о том, что peer идентифицируется его IP адресом внутри его автономной системы. Для роутеров, в которых используется EBGP, neighbors обычно соединены напрямую, и IP адрес neighbor'a - это IP адрес сетевого интерфейса граничной с вашим роутером стороны. (Исключением из этого правила является "[EBGP Multihop](#)," который описан позднее в этом же параграфе).

Для роутеров, на которых используется IBGP, IP адресом может быть адрес любого из интерфейсов роутера в сети, которая входит в вашу AS.

**Некоторые замечания к автономным системам, указанным на [Figure 12-1](#):**

\* Роутеры A и B обмениваются информацией о маршрутизации по EBGP, а роутеры B и C - по IBGP. Заметим, что EBGP peers соединены непосредственно, а IBGP - нет, но поскольку между ними работает IGP, он позволяет им достигать друг друга и обмениваться информацией.

\* Все роутеры, обменивающиеся BGP таблицами внутри AS, должны установить "peers relations" между собой. Это означает, что BGP speakers внутри AS должны быть логически связаны. BGP4 использует два механизма, которые облегчают выполнение требования о полной логической взаимосвязи роутеров внутри AS. Это "Confederations" и "Route Reflectors", которые описываются ниже.

\* AS 200 является "транзитом" для AS 100 и AS 300; То есть AS 200 используется для транзитной передачи пакетов между AS 100 и AS 300.

Для проверки того, работают (доступны ли) BGP peers, вы можете использовать команду:

```
show ip bgp neighbors
```

для роутера "A" с рисунка, приведенного выше, вы получите следующую информацию:

```
RouterA# show ip bgp neighbors BGP neighbor is 129.213.1.1, remote AS 200, external link BGP version 4, remote router ID 175.220.212.1 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 state - указывает, работает и доступен ли peer с вашего роутера; Любое значение state, не совпадающее с established, означает что peer не доступен. Remote router ID - значение главного IP адреса peer роутера (или его главного loopback iface'a). table version number - каждый раз, когда в таблицу маршрутов добавляется новая информация, этот номер увеличивается на 1. Если этот номер постоянно увеличивается, это означает, что роутер "flapping", то есть его таблица маршрутизации все время получает новые значения.
```

Заметим, что после того, как вы изменили конфигурацию своего роутера, надо выполнить команду:

```
clear ip bgp [ip address]
```

для того, чтобы сделать reset BGP сессии с вашим peer'ом.

## **I.1 Internal BGP**

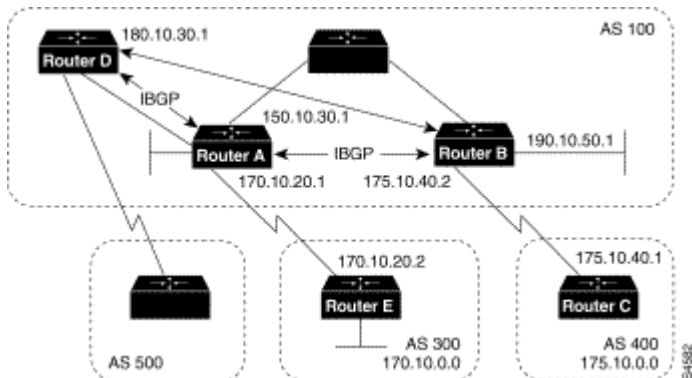
Internal BGP (IBGP) является одной из форм протокола BGP для обмена информацией о маршрутах внутри AS. В принципе, вместо использования IBGP, информация о путях к роутерам, пришедшая по EBGP, может быть redistributed (преобразована) в IGP в пределах данной AS, а затем быть redistributed обратно в другую AS. Однако, использование IBGP является гораздо более гибким, и обеспечивает более эффективные пути для управления обменом информацией как внутри AS, так

и вовне, предоставляя возможность согласовывать обмен информацией о путях к данной AS с внешними neighbors.

Например, с помощью IBGP можно управлять точкой выхода из AS.

**Figure 12-2 shows a topology that demonstrates IBGP.**

**Figure 12-2: Internal BGP Example**



**Приведенные ниже команды конфигурируют Routers A and B в AS 100, и Router C в AS 400:**

```
!Router A
router bgp 100
neighbor 180.10.30.1 remote-as 100
neighbor 190.10.50.1 remote-as 100
neighbor 170.10.20.2 remote-as 300
network 150.10.0.0
!Router B
router bgp 100
neighbor 150.10.30.1 remote-as 100
neighbor 175.10.40.1 remote-as 400
neighbor 180.10.30.1 remote-as 100
network 190.10.50.0
!Router C
router bgp 400
neighbor 175.10.40.2 remote-as 100
network 175.10.0.0
!Router D
router bgp 100
neighbor 150.10.30.1 remote-as 100
neighbor 190.10.50.1 remote-as 100
network 190.10.0.0
```

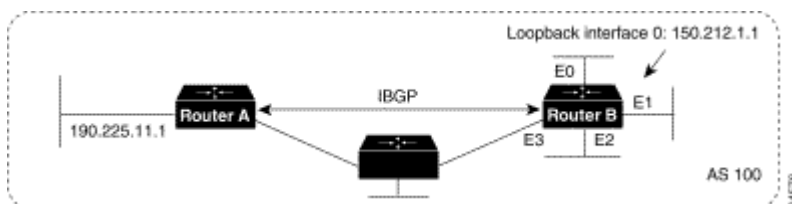
**Замечание:** Когда BGP speaker принимает IBGP update'ы от других BGP speaker'ов в пределах одной AS, он передает эти update'ы только своим neighbors по EBGP.

Например, на приведенном выше рисунке, если бы не было IBGP соединения между роутерами B и D, то роутер A пересылал бы update'ы, принятые от B по IBGP, на роутер E (по EBGP). Если же вы хотите, чтобы роутер "D" принимал update'ы от роутера "B", то роутер "B" надо сконфигурировать так, чтобы роутер "D" стал его BGP peer'ом (что и сделано выше).

### 1.1.1 Loopback Interfaces

Loopback interfaces часто используются IBGP peer'ами. Преимущество использования loopback interfaces заключается в том, что настройка маршрутизации перестает быть связана с IP адресом физического интерфейса. При этом, даже если физический интерфейс будет down, это никоим образом не повлияет на маршрутизацию между автономными системами, т.к. он будет использовать некий псевдо-ip адрес интерфейса loopback, как идентифицирующий данный роутер.

**Figure 12-3: Use of Loopback Interfaces**



На приведенном выше рисунке роутеры A и B обмениваются таблицами маршрутизации по IBGP в пределах AS 100. Если при конфигурировании роутера A в качестве IP адреса neighbor'a мы укажем IP адрес одного из его eth iface'ов, то все, конечно же, будет работать. До тех пор, пока этот eth iface по каким-то причинам не упадет. Если это случится, то обмен таблицами маршрутизации, равно как и трафиком, прекратится в результате того, что станет невозможным установить TCP connection между роутерами A и B.

Поэтому, вместо того чтобы указывать IP адрес eth iface'a роутера B, мы сделаем следующее:

а) сконфигурируем роутер B таким образом, чтобы он получил некоторый дополнительный IP адрес -- адрес его loopback iface'a.

б) настроим обмен таблицами маршрутизации между роутерами A и B таким образом, что на роутере A в качестве IP адреса команде neighbor будет указан IP адрес loopback iface'a роутера B.

в) Роутер В сконфигурируем таким образом, чтобы IP адрес выходящих с него пакетов - был IP адрес интерфейса loopback.

Конфигурация роутера "А":

```
!Router A router bgp 100 neighbor 150.212.1.1 remote-as 100
```

Конфигурация роутера "В":

```
!Router B loopback interface 0 ip address 150.212.1.1 255.255.0.0 !router bgp 100 neighbor 190.225.11.1 remote-as 100 neighbor 190.225.11.1 update-source loopback 0
```

Таким образом, в конфигурации роутера А мы можем указать IP адрес loopback iface'a роутера В (150.212.1.1) в команде neighbor.

Заметим, что loopback ifaces редко используются на роутерах, которые работают по EBGP, поскольку эти роутеры и так зависят от непосредственного физического соединения друг с другом - то есть при падении одного из iface'ов на одном из роутеров связь так или иначе будет прервана, будь то eth или serial iface'ы.

## I.2 External BGP

EBGP используется тогда, когда два bgp speaker'a обмениваются таблицами маршрутизации, находясь в разных AS.

В данном параграфе рассматриваются команды, с помощью которых решаются вопросы некоторой нестандартной конфигурации при обмене таблицами маршрутизации peer'ов из разных AS.

- [I.2.1 EBGP Multihop](#)
- [I.2.2 EBGP Load Balancing](#)
- [I.2.3 Synchronization](#)

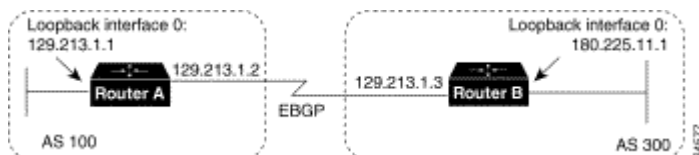
### I.2.1 EBGP Multihop

Multihop используется только с EBGP.

Обычно оба EBGP speaker'a связаны друг с другом через wide-area network (WAN). Однако, иногда требуется настроить обмен трафиком между двумя AS, которые не являются соединенными непосредственно.

**На рисунке 12-4 приведена топология сети, в которой полезно использовать команду neighbor ebgp-multihop.**

**Figure 12-4: EBGP Multihop**



Заметим, что соединение А -- В может быть не непосредственным в данном случае.

Рассмотрим конфигурацию EBGP на роутере "А":

```
!Router A loopback interface 0 ip address 129.213.1.1 !router bgp 100 neighbor 180.225.11.1 remote-as 300 neighbor 180.225.11.1 ebgp-multihop neighbor 180.225.11.1 update-source loopback 0
```

Роутер "В"

```
!Router B loopback interface 0 ip address 180.225.11.1 router bgp 300 neighbor 129.213.1.1 remote-as 100 neighbor 129.213.1.1 ebgp-multihop neighbor 129.213.1.1 update-source loopback 0
```

В обоих случаях (и в конфигурации роутера А, и в В) команда

`neighbor [ip address] ebgp-multihop`

означает тот факт, что для достижения соседней AS \_следующим шагом\_ будет [ip address], указанный в данной команде neighbor, а не ip адрес сетевого интерфейса AS.

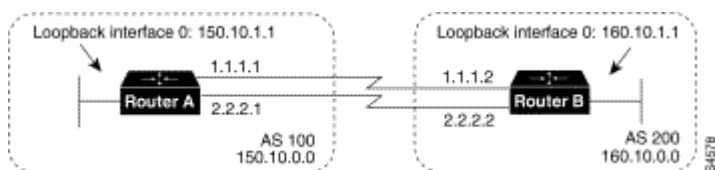
**Замечание:** В случае, когда взаимодействуют два роутера, принадлежащие разным автономным системам, и не являющиеся соединенными непосредственно, требуется прописывать маршрут к "следующему hop'у". Как это делать - будет рассказано немного позже.

### I.2.2 EBGP Load Balancing

Команда neighbor ebgp-multihop, и возможность использования такого понятия, как loopback interface, позволяют конфигурировать load balancing между разными автономными системами,

которые соединены между собой по serial lines.

**Figure 12-5: Load Balancing over Parallel Serial Lines**



```
!Router A interface 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
```

Следующие команды конфигурируют Load Balancing для роутера "B":

```
!Router B interface 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 ebgp-multihop neighbor 150.10.1.1 update-source loopback 0 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
```

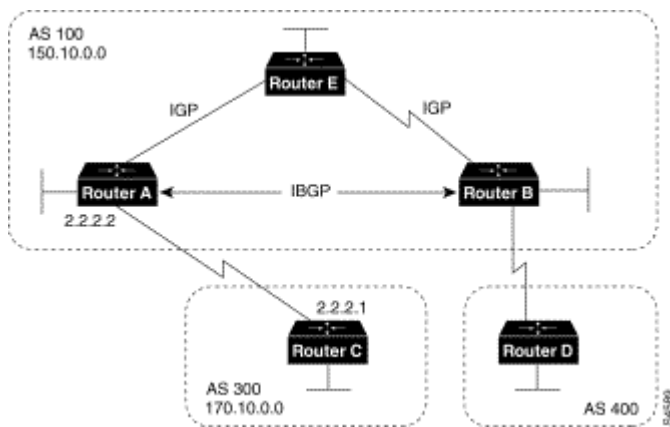
Команды neighbor ebgp-multihop и neighbor update-source позволяют установить next-hop'ом loopback iface для EBGP, это позволяет настроить и использовать load balancing. Статические роуты используются для того, чтобы сети соседней AS были доступны по каждому из них. Итак, роутер "A" может достичь next hop 160.10.1.1, используя один из путей: через реальный serial interface 1.1.1.2, или 2.2.2.2; Аналогично, роутер "B" может достичь next hop 150.10.1.1 одним из двух путей, через любой из реальных serial interface'ов "A": а именно через 1.1.1.1 или через 2.2.2.1

Тот же самый результат можно получить, настроив IGP.

### 1.2.3 Synchronization

Когда наша AS обеспечивает transit service информации в другую автономную систему, и если в нашей AS присутствуют роутеры, на которых не поддерживается BGP, transit traffic может быть утерян в случае, если на одном из промежуточных не-BGP роутеров неизвестен IGP маршрут к удаленной автономной системе.

**Figure 12-6: Synchronization**



Ключевой момент: роутеры "A" и "B" проводами не связаны, они связаны только логически по IBGP; Трафик же идет через роутер "E", который не поддерживает IBGP, но зато работает по IGP. Этот роутер "E" внутри AS100 и надо научить маршрутам, проходящим через AS100 - иначе весь трафик по этим маршрутам, идущий через "E", будет утерян.

Подробнее:

Допустим, роутер C отправляет update'ы, содержащие маршрут к сети 170.10.0.0, роутеру A. Поскольку A и B взаимодействуют по IBGP, то роутер B получает информацию о пути к сети 170.10.0.0 через IBGP.

Если роутер B захочет обратиться к сети 170.10.0.0, он отправит трафик роутеру E. Но, если роутер A не делает redistribute маршрутов в IGP (в-частности, о сети 170.10.0.0), роутер E не будет знать путей для достижения сети 170.10.0.0 и пакеты будут утеряны.

Если роутер "B" передаст маршруты в AS 400, и hosts этой автономной системы попробуют обратиться к сети 170.10.0.0 до того, как роутер "E" узнает об этой сети по IGP, трафик пришедший от "D" к "B" с destination 170.10.0.0 и направленный на "E" будет утерян (dropped).

Данная ситуация разрешается с помощью правила синхронизации обмена путями по BGP, которое (разберем на нашем примере) заключается в том, что если AS100 передает трафик от одной системы (300) к другой (400), роутер промежуточной (100) автономной системы не будет принимать маршруты по BGP от граничных AS (например, информацию 170.10.0.0) до тех пор, пока все роутеры той AS, к которой он принадлежит (и через которые в результате пойдет трафик на 170.10.0.0) не узнают о сети 170.10.0.0 по \_IGP\_ (в конечном счете, в результате трансляции BGP table --> IGP table). То есть, роутер "B" передаст информацию о маршрутах к сети 170.10.0.0 на роутер "D" AS400 только после того, как сеть 170.10.0.0 станет доступна внутри AS100 через протокол IGP.

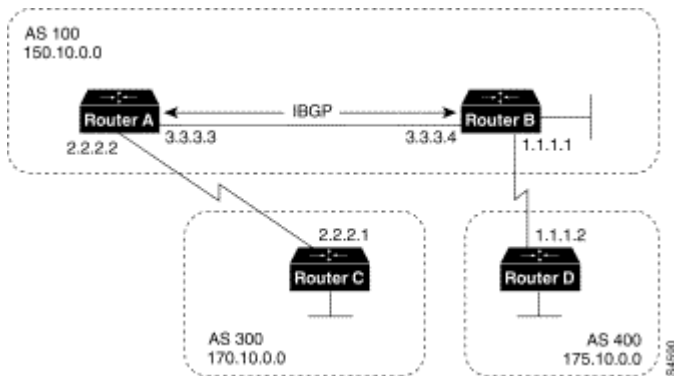
Заметим, что при выполнении некоторых условий (они описаны ниже) можно отключить синхронизацию, которая на самом деле замедляет скорость передачи данных:

\*) Если ваша AS не передает трафик от одной чужой AS к другой AS.

\*) Все transit routers в вашей AS поддерживают BGP.

**Figure 12-7** На приведенном ниже рисунке рассматривается ситуация, в которой можно отключить *synchronization*.

**Figure 12-7: Disabled Synchronization**



```
!Router A network 150.10.0.0 neighbor 3.3.3.4 remote-as 100 neighbor 2.2.2.1 remote-as 300 no synchronization
!Router B 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
!Router D router bgp 400 neighbor 1.1.1.1 remote-as 100 network 175.10.0.0
```

Команда "no synchronisation" означает, что роутер "B" добавит в свою таблицу маршрутов информацию с роутера "C" о сети 170.10.0.0 и передаст эту информацию на роутер "D" другой AS 400 без того, чтобы сначала убедиться в том, что маршрут к сети 170.10.0.0 в пределах AS 100 стал известен по IGP.

### I.3 BGP and Route Maps

Route Maps используются в BGP для управления таблицами маршрутов и указания условий, при выполнении которых маршруты передаются между доменами. Формат route map следующий:

**route-map map-tag [[permit | deny] | [sequence-number]] map tag** - это имя,

которое идентифицирует route map; **sequence number** - идентифицирует положение данного правила в route map относительно других правил в том же самом route map.

Например, вы можете использовать следующие команды конфигурации роутера для определения route map с именем MYMAP:

**route-map MYMAP permit 10!** здесь будет набор правил, которые выполняются первыми **route-map MYMAP permit 20!** а здесь - набор правил, которые выполняются в том случае, если не выполняются правила, указанные в разделе с **sequence number = 10**

Когда BGP применяет MYMAP к routing updates, первым выполняется правило с наименьшим номером. Команда "match" указывает критерий, равенство которому должно проверяться, а команда "set" указывает действие, которое должно быть выполнено над маршрутом, если в таблице встречается правило, подходящее под условие match.

Простой пример route map:

```
route-map MYMAP permit 10 match ip address 1.1.1.1 set metric 5
```

в нашем простом примере, если в update'e таблицы маршрутизации встречается IP адрес 1.1.1.1, BGP устанавливает этому update'у (вернее, той записи, которая встретилась в таблице маршрутизации и удовлетворила нашему критерию "match") значение metric = 5 (его смысл будет

рассмотрен ниже). После чего update пересылается neighbor'у, который указан в строке, например, следующего вида:

```
neighbor 1.2.3.4 route-map MYMAP out,
```

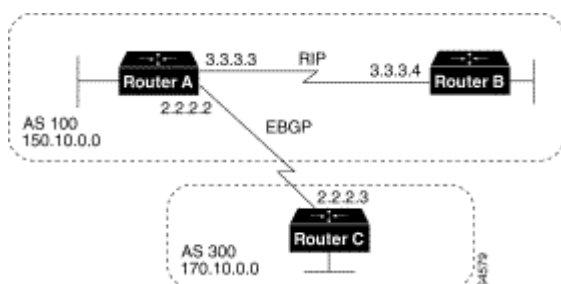
поскольку в описании самого набора правил route-map стоит слово permit.

После этого набор route-map'ов с большими sequence numbers не рассматривается, естественно. В том случае, когда update не встречает подходящий критерий "match", BGP использует правило из route-map с большим номером, и так далее. Если ни один из критериев не подходит, то update не передается дальше.

В том случае, когда update удовлетворяет критерию, указанному в match, а route-map содержит ключевое слово "deny", BGP также заканчивает обработку списка критериев, и update не передается дальше.

**Замечание:** Route map не может быть использован для фильтрации incoming BGP update'ов, основываясь на IP адресе. Однако, вы можете управлять outgoing update'ами, используя в качестве критерия сравнение IP адреса в правиле match.

### Figure 12-8: Route Map Example



На этом рисунке роутеры "А" и "В" в качестве протокола маршрутизации между собой используют RIP. Роутеры "А" и "С" - BGP. Например, если вы захотите, чтобы роутер "А" передавал на "В" информацию о маршрутах к сети 170.10.0.0 с метрикой 2, а все остальные пути - с метрикой 5, роутер "А" должен быть сконфигурирован следующим образом:

```
!Router A
router rip
network 3.0.0.0
network 2.0.0.0
network 150.10.0.0
passive-interface serial 0
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
```

Итак, если передаваемый через роутер "А" путь встречает IP адрес 170.10.0.0, то дальше он передается с metric 2;

Если ip address передаваемого через роутер "А" пути не равен 170.10.0.0, то далее он передается с метрикой 5.

Предположим, что на роутере "С" (рис. выше) вы хотите установить community attribute=300 для outgoing updates о сети 170.10.0.0

```
!Router C
router bgp 300
network 170.10.0.0
neighbor 2.2.2.2
remote-as 100
neighbor 2.2.2.2
route-map SETCOMMUNITY out
!route-map SETCOMMUNITY
permit 10
match ip address 1
set community 300
!access-list 1
permit 0.0.0.0 255.255.255.255
```

Access list 1 запрещает какие бы то ни было update'ы для сети 170.10.0.0 и разрешает update'ы для всех других сетей.

**Q/ERROR:** Наверное, надо добавить строки

```
access-list 1 permit 170.10.0.0 0.0.255.255 access-list 2 deny 0.0.0.0
255.255.255.255
```

вместо той, которая используется выше, а так же расписать

```
route-map SETCOMMUNITY permit 10/20 соответственно.
```

Задача-то заключалась в том, чтобы присвоить community для outgoing updates о сети 170.10.0.0.

## Advertising Networks

Сети, которые расположены внутри данной AS, будем называть "**originate**", то есть включенными в данную AS. Для информирования других автономных систем о сетях, которые включает данная AS, данная AS должна анонсировать эти сети во внешний мир. То есть передавать маршруты об этих сетях тем AS, которые являются neighbors.

Протокол BGP поддерживает три пути, с помощью которых AS может анонсировать данные сети во внешний мир. А именно:

- [Redistributing Static Routes](#)
- [Redistributing Dynamic Routes](#)



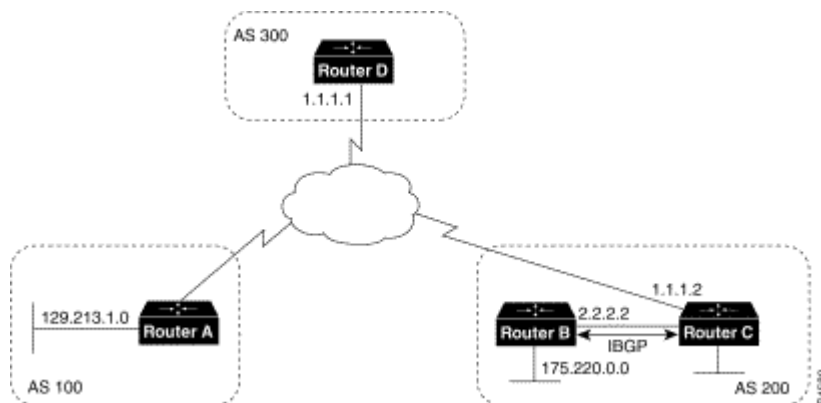
- **Using the network Command**

**Важно помнить**, что пути, которые мы анонсируем таким образом, являются дополнением к другим путям в таблице маршрутизации BGP, которые роутер узнал по протоколу bgp от своих внутренних и внешних соседей (neighbors).

BGP всегда передает имеющуюся у него информацию от одного peer'a к другим peer'ам. Разница заключается в том, что пути, полученные с помощью команд "network" и "redistribute", указывают, что AS данного роутера является *originating* для данной сети

В данном параграфе мы рассмотрим сеть, которая показана на рис. ниже:

**Figure 12-9: Network Advertisement Example 1**



#### I.4.1 Redistributing Static Routes

Итак, один из путей анонсирования сети (или подсети), которая включена в AS - это redistribute static routes into BGP. Разница между redistribute static и redistribute dynamic routes заключается лишь в том, что в случае redistribute static route, BGP устанавливает флаг origin attribute (в таблице BGP маршрутов) для данного пути равным "?", то есть "Incomplete".

**Замечание:** другие значения флага Origin Attribute рассмотрены ниже, в главе "[Origin Attribute](#)".  
Итак, для конфигурирования роутера "C" на приведенном выше рисунке, как роутера, который включает путь к сети 175.220.0.0 в BGP, используются следующие команды:

```
!Router Crouter bgp 200neighbor 1.1.1.1 remote-as 300redistribute static!ip route 175.220.0.0 0.0.255.255 null 0
```

Команда роутера **redistribute static** означает, что все статические маршруты будут redistribute в BGP. А статические маршруты, как видим, прописываются командой "**ip route**", которая также присутствует в конфигурации Router'a C.

Команда роутера ip route устанавливает статический маршрут для сети 175.220.0.0; Теоретически, указание "null 0" интерфейса означает, что пакеты, предназначенные для сети 175.220.0.0, будут потеряны. На практике же, условие "ip route 175.220.0.0 0.0.255.255 null 0" представляет собой довольно специфическое правило для пакетов, предназначенных любой другой сети, кроме как 175.220.0.0, и роутер отправит эти пакеты мимо соответствующего интерфейса.

**Note:** Рекламирование сетей с помощью статической маршрутизации - это лучший путь для анонсирования сетей, поскольку он предотвращает пути от роутеров, которые могут быть подвержены могут быть подвержены "flapping".

Заметим в любом случае, что команда конфигурации роутера "redistribute" (будь то static или dynamic) является лишь способом, которым BGP роуты добавляются в маршруты IGP route table.

**Q/ERROR:** А не наоборот - IGP в BGP? А:

#### I.4.2 Redistributing Dynamic Routes

Другой путь анонсирования своих сетей - это redistribute dynamic routes. В этом случае вы делаете redistribute IGP routes (полученных по таким протоколам, как Enhanced IGRP, IGRP, IS-IS, OSPF и RIP) в BGP. Некоторые из этих IGP роутов могли быть получены и из BGP, поэтому при использовании динамического redistributing'a of маршрутов вам требуется настраивать access-lists для предотвращения пере-редистрибуции маршрутов, пришедших откуда-то по bgp, оттранслированных в igp - обратно в BGP.

Предположим, что на рисунке выше, Routers B и C работают по IBGP, что роутер "C" имеет информацию о сети 129.213.1.0 (полученную по bgp), и что роутер "B" делает redistributing 129.213.1.0 обратно в Enhanced IGRP. Итак, конфигурируем роутер "C" следующим образом:

```
!Router C!router eigrp 10network 175.220.0.0redistribute bgp 200
-> в eigrpredistribute connecteddefault-metric 1000 100 250 100 1500!router bgp
200neighbor 1.1.1.1 remote-as 300neighbor 2.2.2.2 remote-as 200neighbor 1.1.1.1
distribute-list 1 outredistribute eigrp 10!access-list 1 permit 175.220.0.0 0.0.255.255
```

Итак, что мы имеем выше: команда "redistribute eigrp 10" делает redistribute (попросту говоря, преобразование) таблиц маршрутизации протокола Enhanced IGRP for your process ID 10 в BGP. **Примечание:** Обычно преобразование BGP --> IGP следует опасаться, по причине включения очень большого количества путей в таблицу IP маршрутизации.

Далее. Команда конфигурации роутера "neighbor 1.1.1.1 distribute-list 1 out" применяет access-list 1 для выходящих анонсируемых этим роутером путей одному из своих neighbor'ов: 1.1.1.1 (Router D). Access list 1 указывает, что сеть 175.220.0.0 будет анонсирована. Все другие сети, такие как 129.213.1.0, подразумеваются защищенными от анонсирования. Access list запрещает обратное ре-транслирование сети 129.213.1.0 в BGP, чтобы она не выглядела originated AS 200, и позволяет BGP анонсировать сеть 175.220.0.0, как сеть AS 200.

**Примечание:** Redistribution of dynamic routes требует осторожности в использовании access lists для предотвращения попадания некоторых путей, пришедших по BGP, обратно в BGP. Когда это возможно, вам следует использовать команду "network" (она описана позднее в этом параграфе) или команду "redistribute static" вместо redistribute dynamic routes.

### 1.4.3 Using the network Command

Другой путь для анонсирования сетей - это использование команды "**network**". Когда команда network используется с BGP, в качестве ее аргумента указывается сеть, которая будет анонсироваться данной AS, как входящая в нее. (Для сравнения: когда эта команда используется с одним из IGP протоколов, например с RIP'ом, команда network идентифицирует интерфейс, с которым работает IGP). Команда network работает и для сетей, о которых роутер узнал динамически, и для сетей, пути к которым сконфигурены с помощью команд статической маршрутизации. Origin attribute маршрутов, который присутствует в таблице маршрутизации в результате выполнения команды конфигурации **network**, равен "i", то есть IGP.

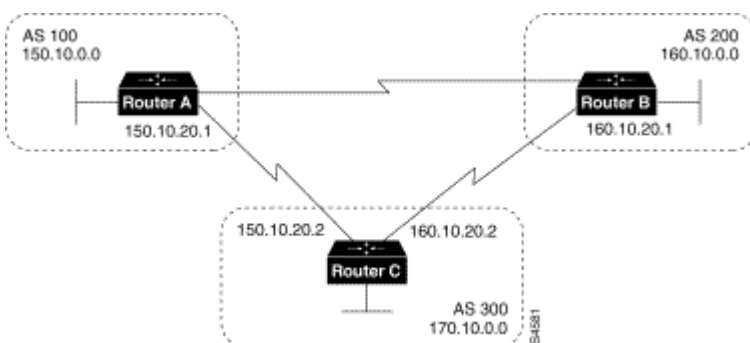
Команды, приводимые ниже, конфигурируют роутер "C" для того, чтобы он анонсировал сеть 175.220.0.0:

```
!Router Crouter bgp 200neighbor 1.1.1.1 remote-as 300network 175.220.0.0
```

Команда network в данном случае означает, что Router "C" генерирует запись в таблице BGP маршрутизации о сети 175.220.0.0.

**На рисунке 12-10 показана другая топология сети, в которой также используется команда network:**

**Figure 12-10: Network Advertisement Example 2**



```
!Router Arouter bgp 100neighbor 150.10.20.2 remote-as 300network 150.10.0.0 !Router
Brouter bgp 200neighbor 160.10.20.2 remote-as 300network 160.10.0.0 !Router C
router bgp 300neighbor 150.10.20.1 remote-as 100neighbor 160.10.20.1 remote-as 200
network 170.10.0.0
```

Для того, чтобы быть уверенным в отсутствии зацикливаний (loop-free interdomain topology), BGP не принимает updates, которые originates from its own AS. Поясним: на приведенном выше рисунке, если Router A генерирует updates о сети 150.10.0.0 с установленным origin в значение AS100, и отправит эти updates роутеру C, то роутер C передаст updates роутеру B с тем же самым origin'ом AS100. Когда роутер B отправит свои updates роутеру A, тот признает свои собственные updates и игнорирует их.

## II. BGP Decision Algorithm

**BGP speaker** - это роутер, который обмениваются маршрутами с другим роутером (тоже с BGP speaker'ом) по протоколу BGP.

Итак, когда BGP speaker принимает updates от нескольких AS, и в каждом из этих updates содержатся различные пути к одной и той же точке в Inet, встает вопрос о выборе оптимального маршрута к этой точке, будь то автономная система, или сеть, или конкретный хост с IP адресом. Выбрав оптимальный путь в некоторую точку, роутер рассказывает своим соседям об этом пути. Алгоритм выбора оптимального пути базируется на нескольких параметрах (next hop, administrative weights, local preference, origin of the route, path length и др.), которые содержатся в update'е, а так же на некоторых других факторах, устанавливаемых при конфигурации BGP peer'ов.

Эта глава описывает следующие атрибуты и факторы, которые использует BGP в процессе принятия решения о наилучшем пути к некоторой точке в сети.

- [II.1 AS\\_path Attribute](#)
- [II.2 Origin Attribute](#)
- [II.3 Next Hop Attribute](#)
- [II.4 Weight Attribute](#)
- [II.5 Local Preference Attribute](#)
- [II.6 Multi-Exit Discriminator Attribute](#)
- [II.7 Community Attribute](#)

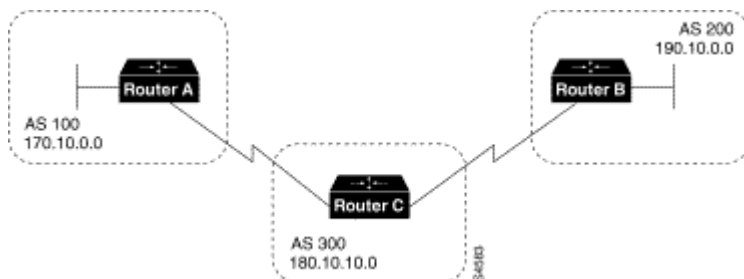
Итак, рассмотрим все перечисленные выше факторы по очереди.

### II.1 AS\_path Attribute

Всякий раз, когда updates передаются через AS, BGP добавляет AS number к update'у. AS path attribute - это список номеров автономных систем, через которые update прошел (а, следовательно, пройдет и дейтаграмма) в процессе достижения своей цели. AS-SET - набор чисел, а именно номеров автономных систем, которые были пройдены update'ом.

**Рассмотрим сеть, которая показана на [Figure 12-11](#).**

**Figure 12-11: AS\_path Attribute**



На этом рисунке роутер B анонсирует сеть 190.10.0.0 как входящую в AS 200 и в этом случае AS\_path равен 200. Когда update о 190.10.0.0 передается в соседнюю AS 300, роутер C добавляет номер своей AS (а именно, 300) к AS\_path, и таким образом, когда update достигнет роутера A, два номера AS будут прицеплены к нему: 200 и 300. Таким образом, AS\_path attribute для достижения сети 190.10.0.0 из роутера A - это 300, 200. Аналогично, AS\_path для достижения сети 170.10.0.0 с роутера B - это 300, 100.

### II.2 Origin Attribute

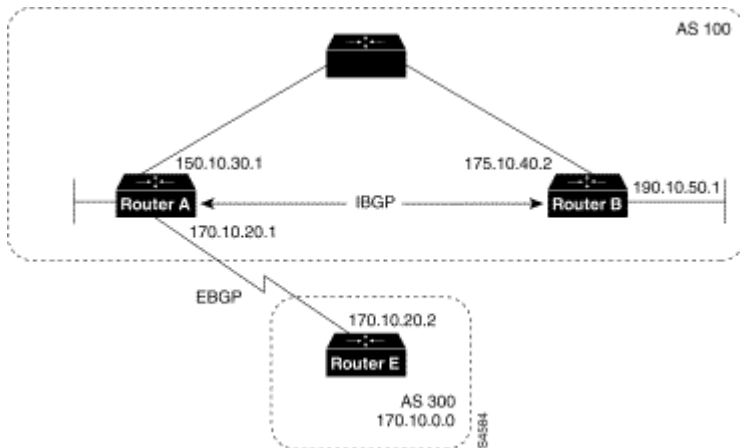
**Origin attribute** содержит информацию о том, кто является "хозяином" данного пути. То есть кто инициировал запись в таблице BGP маршрутизации о данном конкретном пути.

Origin attribute может принимать одно из трех значений:

**IGP** - route является внутренним по отношению к данной AS. Это значение **i** устанавливается как атрибут записи в таблице маршрутизации в том случае, когда роутер использует команду "network" для добавления данного пути в таблицу BGP маршрутизации. IGP origin type представлен буквой "i" в результатах show ip bgp команды. **EGP** - информация о данном пути получена через Exterior Gateway Protocol **e** (EGP). EGP origin type представлен буквой "e" в выводе show ip bgp. **Incomplete** - origin of route неизвестен или получен каким-то третьим? (не

одним из описанных выше) способом. А так же origin incomplete имеет место в тех случаях, когда route получен путем преобразования маршрута (например, полученного из записи о статической маршрутизации как redistribute static) в BGP. В этом случае origin attribute выглядит как "?".

**Figure 12-12: Origin Attribute**



Роутеры конфигурируются следующими командами:

```
!Router A
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 null 0
!Router B
router bgp 100
neighbor 150.10.30.1 remote-as 100
network 190.10.50.0
!Router E
router bgp 300
neighbor 170.10.20.1 remote-as 100
network 170.10.0.0
```

Исходя из данной конфигурации, получаем следующие значения AS\_path и Origin attribute в таблице маршрутизации:

1. в BGP route table на роутере A путь достижения сети 170.10.0.0 имеет AS\_path 300 и Origin attribute "IGP" (поскольку для анонсирования своей сети роутер "E" использует команду network 170.10.0.0).
2. в BGP route table на роутере A путь к сети 190.10.50.0 имеет AS\_path=пусто, и Origin attribute = "IGP" (по той же причине).
3. в BGP route table на роутере E путь к сети 150.10.0.0: AS\_path=100, Orig. attr = "IGP" (по той же причине).
4. в BGP route table на роутере E путь к 190.10.0.0: AS\_path=100, Orig. attr. = Incomplete (тк на "A" данная сеть анонсируется через "redistribute route").

### II.3 Next Hop Attribute

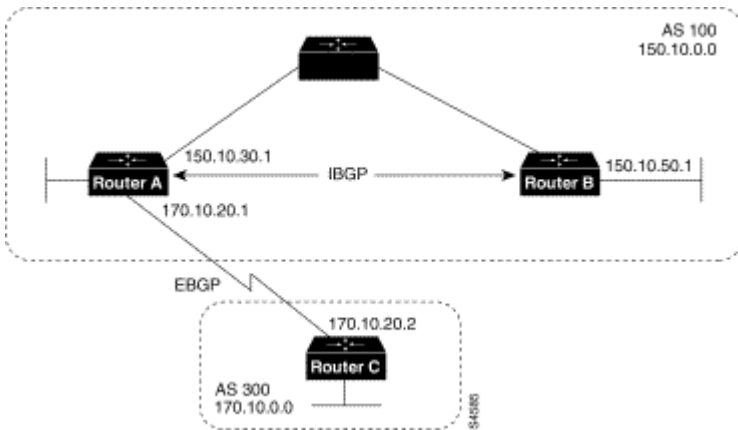
В BGP понятие **"Next Hop Attribute"** означает IP адрес роутера, который будет следующим в цепочке роутеров на пути достижения дейтаграммой конечной точки.

Естественно, что для EBGP понятие "next hop" - это IP адрес neighbor'a, который указан в конфигурации нашего роутера командой:

```
neighbor [ip_addr] remote-as [as]
```

Исключением является тот случай, когда имеем дело с Multiaccess Media, (тогда next hop должен быть IP address роутера в данной подсети).

**Figure 12-13: Next Hop Attribute**



На приведенном выше рисунке роутер "С" анонсирует сеть 170.10.0.0 роутеру "А" с next hop attribute 170.10.20.2. А роутер "А" анонсирует 150.10.0.0 роутеру "С" с next hop = 170.10.20.1. Небольшое пояснение: на таблице маршрутизации надо смотреть со стороны того роутера, на котором она имеет место быть; Для того, чтобы с роутера "С" увидеть сеть 150.10.0.0, надо обратиться к роутеру "А" (его ip address 170.10.20.1). Для этого роутер "А" и анонсирует сеть своей AS, указывая next hop для таблицы роутера "С" как 170.10.20.1;

**Заметим, что есть такое правило: next hop маршрута, полученного по EBGP, передается без изменения в IBGP.**

А именно: роутер "А" анонсирует сеть 170.10.0.0 своему IBGP peer'у - роутеру "В" с next hop attribute 170.10.20.2, поэтому в таблице маршрутизации на роутере "В" путь к данной сети будет иметь next hop'ом исходный 170.10.20.2, вместо 150.10.30.1 (который должен был бы быть по логике вещей). Из-за этого надо быть совершенно уверенными, что роутер "В" может достичь 170.10.20.2 через IGP (например, через RIP). Иначе роутер "В" потеряет пакеты, предназначенные для 170.10.0.0.

Например, если роутер "В" работает по IGRP, то роутер "А" должен поддерживать IGRP для сети 170.10.0.0; Конечно, вы можете захотеть отключить IGRP на линке к "С" и тогда "А" и "С" будут обмениваться только BGP update'ами.

Следующие команды используются для конфигурации роутеров с приведенного выше рисунка:

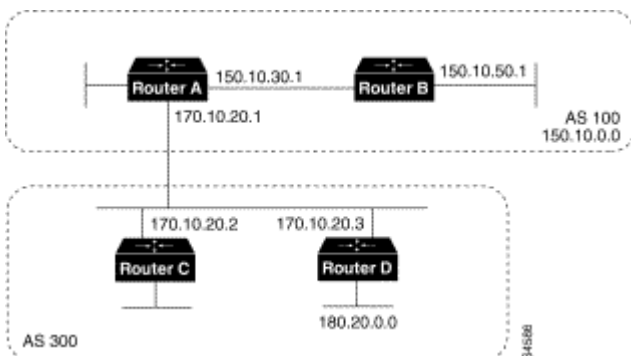
```
!Router A
router bgp 100
neighbor 170.10.20.2 remote-as 300
neighbor 150.10.50.1 remote-as 100
network 150.10.0.0
!Router B
router bgp 100
neighbor 150.10.30.1 remote-as 100
!Router C
router bgp 300
neighbor 170.10.20.1 remote-as 100
network 170.10.0.0
```

Заметим, что Router "С" анонсирует сеть 170.10.0.0 роутеру "А" с next hop attribute 170.10.20.2, а роутер "А" в свою очередь анонсирует (как бы передает) эту сеть 170.10.0.0 роутеру "В" с исходным next hop attr. 170.10.20.2 (собственно, эти слова - всего лишь повторение того, что уже сказано выше). Итак, next hop путей, полученных по EBGP, передается без изменения в IBGP neighbor'ам.

### II.3.1 Next Hop Attribute and Multiaccess Media

При работе в Multiaccess Media, например в Ethernet сети, BGP может устанавливать значение next hop attribute по-другому.

**Figure 12-14: Next Hop Attribute and Multiaccess Media**



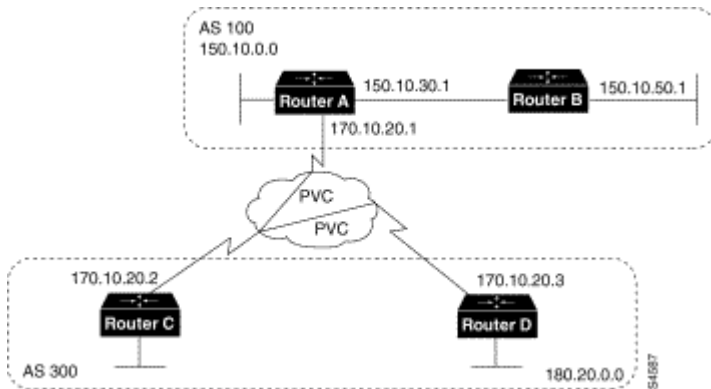
Сразу заметим, что роутеры А,С,Д находятся в одной \_подсети\_. Это важно. Итак, пусть С и D обмениваются таблицами маршрутизации по протоколу OSPF, и кроме этого роутер С обменивается таблицами маршрутизации с роутером А по протоколу BGP. Роутер С может достигнуть сеть 180.20.0.0 через 170.10.20.3; Когда С отправляет BGP update'ы роутеру А о пути, по

которому доступна сеть 180.20.0.0, он указывает в качестве next hop адрес 170.10.20.3, вместо своего собственного адреса 170.10.20.2 - это делается потому, что все три роутера находятся в одной и той же подсети, и роутеру "A" будет более очевидно использовать для достижения сети 180.20.0.0 адрес роутера D: 170.10.20.3.

### Next Hop Attribute and Nonbroadcast Media Access

На рисунке 12-15 три сети соединены в Nonbroadcast Media Access (NBMA), такой как Frame Relay.

Figure 12-15: Next Hop Attribute and Nonbroadcast Media Access



Если роутеры A,C,D используют общую схему обмена данными, такую как Frame Relay (или любой из NBMA), то роутер "C" анонсирует сеть 180.20.0.0 роутеру "A" со значением next hop 170.10.20.3, точно так же как если бы эти роутеры были включены в сеть Ethernet. Но, в данном случае есть одна проблема: она заключается в том, что роутер "A" не имеет постоянного прямого соединения с роутером "D" (это называется "direct Permanent Virtual Connection, или PVC) и не может достичь этого самого next hop'a, который ему указан в таблице маршрутизации. Таким образом, маршрутизация не работает. Для решения данной проблемы, используется команда роутера:

*next-hop-self*

Итак, конфигурация роутера "C":

```
!Router Crouter bgp 300neighbor 170.10.20.1 remote-as 100neighbor 170.10.20.1 next-hop-self
```

Последняя из приведенных выше команд указывает роутеру "C", что надо анонсировать сеть 180.20.0.0 со значением next-hop-attribute равным 170.10.20.2.

## II.4 Weight Attribute

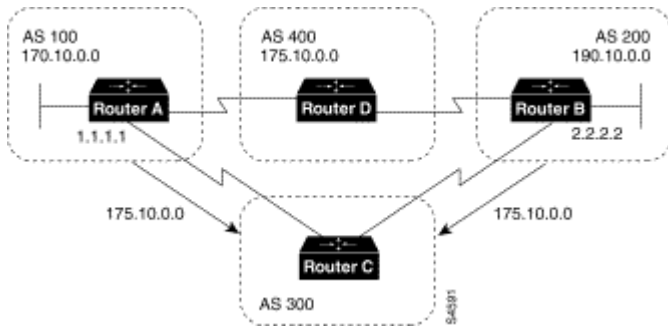
**Weight attribute** - это специальный атрибут, который используется роутерами Cisco в процессе выбора наилучшего пути к какому-то роутеру, или сети - если к нему ведет больше одного пути. Значение "weight attribute" является локальным для роутера, на котором оно устанавливается. Это значение не передается по таблицам маршрутизации на другие роутеры.

По умолчанию, значение **weight attribute = 32768** для тех путей, которые берут начало на данном роутере, и **0** - для всех остальных путей.

Пути (маршруты) с большим "weight" являются предпочтительными, когда есть несколько путей до какой-то точки в inet.

Рассмотрим сеть, показанную на **рис. 12-16**.

Figure 12-16: Weight Example



На данном рисунке роутеры "А" и "В" узнают о сети 175.10.0.0 от AS 400, и каждый из роутеров передает updates об этой сети на роутер "С". Таким образом, в таблице маршрутов на роутере "С" появляются два пути для достижения сети 175.10.0.0; и этот роутер должен выбрать, какой из маршрутов ему использовать. Поэтому на роутере "С" устанавливаем весовые коэффициенты для обоих маршрутов к 175.10.0.0 (пришедших соответственно от "А" и "В"). Причем, если установить weight больше для updates, пришедших с "А", то "С" будет использовать в качестве next hop'a для достижения 175.10.0.0 роутер "А". Если же на "С" установить weight для маршрутов, пришедших с "В" больше, то маршрутизация к данной сети пойдет через "В".

Существуют три способа, с помощью которых на "С" можно установить **"weight"** для update'ов, пришедших с роутера "А":

- [II.4.1 Использование Access List для установки Weight Attribute](#)
- [II.4.2 Использование a Route Map для установки Weight Attribute](#)
- [II.4.3 Использование the neighbor weight Command для установки Weight Attribute](#)

#### II.4.1 Использование Access List для установки Weight Attribute

Приведенные ниже команды конфигурации роутера "С" используют access-lists и значение AS\_path для присваивания router updates коэффициента "weight".

```
!Router Crouter bgp 300neighbor 1.1.1.1 remote-as 100neighbor 1.1.1.1 filter-list 5
weight 2000neighbor 2.2.2.2 remote-as 200neighbor 2.2.2.2 filter-list 6 weight 1000!ip
as-path access-list 5 permit ^100$!ip as-path access-list 6 permit ^200$
```

В данном примере весовой коэффициент 2000 присваивается всем update'ам, пришедшим от neighbor'a с IP адресом 1.1.1.1 - это устанавливается в access-list'e 5. Этот access-list 5 разрешает (тк стоит permit) update'ы, у который AS\_path начинается с 100 (см "^") и заканчивается 100 (см "\$").

Заметим, что символы ^ и \$ используются в regular expressions, и если вы хотите познакомиться с ними поближе, обратитесь к приложению о Regular Expressions в Cisco Internetwork Operating System (Cisco IOS) software configuration guide and command references.

В приведенном выше примере весовой коэффициент 1000 присваивается всем update'ам от neighbor'a 2.2.2.2 - это описано в access-list'e 6. Access list 6 разрешает все update'ы, чьи AS\_path начинаются с 200 и заканчиваются 200 (т.е. содержат только одно значение AS 200 в AS\_path).

В результате данной конфигурации весовой коэффициент 2000 присваивается всем route update'ам, пришедшим от AS 100, и весовой коэффициент 1000 присваивается всем update'ам, пришедшим от AS 200.

#### II.4.2 Using a Route Map to Set the Weight Attribute

Приведенные ниже команды конфигурации роутера "С" используют route map для присваивания weight attribute приходящим на данный router update'ам.

```
!Router Crouter bgp 300neighbor 1.1.1.1 remote-as 100neighbor 1.1.1.1 route-map
SETWEIGHTIN inneighbor 2.2.2.2 remote-as 200neighbor 2.2.2.2 route-map SETWEIGHTIN
in!ip as-path access-list 5 permit ^100$!route-map SETWEIGHTIN permit 10match as-path
5set weight 2000route-map SETWEIGHTIN permit 20set weight 1000
```

В данном случае первый набор правил (permit 10) устанавливает weight attr. равным 2000 для всех update'ов, пришедших с AS 100 (и имеющих AS\_path=100), а второй набор правил (permit 20) устанавливает weight = 1000 всем остальным update'ам (в том числе и с AS200, что и требовалось выше).

### II.4.3 Using the neighbor weight Command to Set the Weight Attribute

```
!Router Crouter bgp 300neighbor 1.1.1.1 remote-as 100neighbor 1.1.1.1 weight
2000neighbor 2.2.2.2 remote-as 200neighbor 2.2.2.2 weight 1000
```

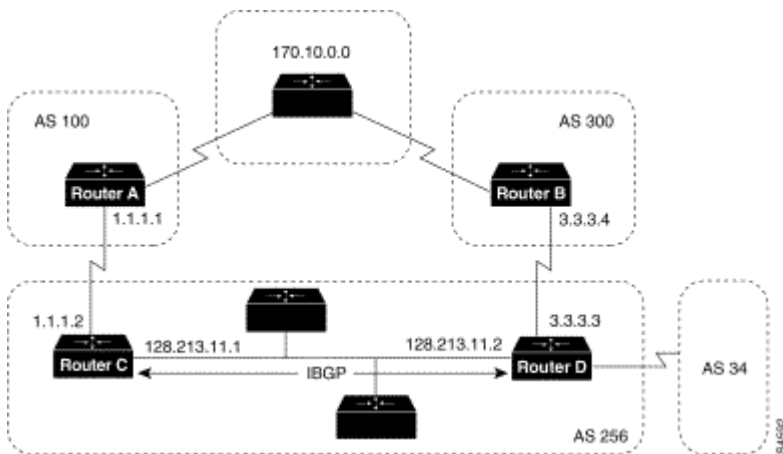
Данная конфигурация устанавливает weight 2000 для всех маршрутов, пришедших \_от\_ AS100, и weight 1000 для всех маршрутов, пришедших \_от\_ AS200.

Разумеется, поскольку weight для маршрутов, пришедших от AS100, выше, то роутер "C" будет обмениваться трафиком с роутером 1.1.1.1 AS100 с большим приоритетом.

### II.5 Local Preference Attribute

Когда для доступа к некоторой точке сети существует несколько путей, local preference attribute указывает наилучший путь. Чем больше local preference для данного конкретного пути, тем выше его приоритет. Значение local preference по умолчанию - 100. В отличие от weight attribute, который относится только к данному локальному роутеру, local preference является частью routing update и имеет силу при обмене маршрутами среди роутеров в пределах одной AS (на той AS, на одном из роутеров которой lpref установлен).

Figure 12-17: Local Preference



Итак, AS256 получает route updates о пути к сети 170.10.0.0 от AS100 и AS300. Есть два способа установить local preference:

- [II.5.1 Using the bgp default local-preference Command](#)
- [II.5.2 Using a Route Map to Set Local Preference](#)

Рассмотрим сначала первый способ, как самый простой (но не самый лучший).

#### II.5.1 Using the bgp default local-preference Command

Итак, в приведенной ниже конфигурации роутеров "C" и "D" используется команда **bgp default local-preference [n]**

```
!Router Crouter bgp 256neighbor 1.1.1.1 remote-as 100neighbor 128.213.11.2 remote-as
256bgp default local-preference 150 !Router Drouter bgp 256neighbor 3.3.3.4 remote-as
300neighbor 218.213.11.1 remote-as 256bgp default local-preference 200
```

Конфигурация роутера "C" при помощи bgp default local-preference означает, что все update'ы от AS300 получают lpref=150. Конфигурация роутера "D" означает, что всем update'ам с AS100 устанавливается lpref =200.

Поскольку маршруты внутри AS обмениваются по IBGP (между роутерами "C" и "D"), то оба этих роутера определяют (исходя из значения lpref передаваемых между ними маршрутов), что наилучший маршрут к 170.10.0.0 - через роутер "B" в AS300.

#### II.5.2 Using a Route Map to Set Local Preference

Использование route-мар предоставляет больше гибкости, чем использование bgp default local preference , поскольку если при использовании bgp default local preference на "D" все пути, прошедшие через роутер "D" (например) приобретают значение lpref = 200, в том числе и update'ы с AS 34. А при использовании route-мар можно указать, каким именно автономным системам добавлять lpref, а каким - нет.



Рассмотрим конфигурацию роутера "D", при которой Ipref устанавливается только для update'ов, полученных от AS 300.

```
!Router Drouter bgp 256neighbor 3.3.3.4 remote-as 300neighbor 3.3.3.4 route-map SETLOCALIN in - Q/ERROR: в оригинале не было "neighbor"neighbor 128.213.11.1 remote-as 256!ip as-path access-list 7 permit ^300$ - Q/ERROR: в оригинале не было "access-list"!route-map SETLOCALIN permit 10match as-path 7set local-preference 200!route-map SETLOCALIN permit 20! пусто
```

В соответствии с данной конфигурацией, local preference attribute всех пришедших от AS 300 update'ов будет установлен равным 200. Правило "20" в SETLOCALIN предназначено для того, чтобы принимать updates от всех остальных роутеров.

## II.6 Multi-Exit Discriminator Attribute

Multi-Exit Discriminator Attribute - это способ указания наилучшего пути своим внешним neighbor'ам.

Меньшее значение MED является более предпочтительным. Значение по умолчанию = 0.

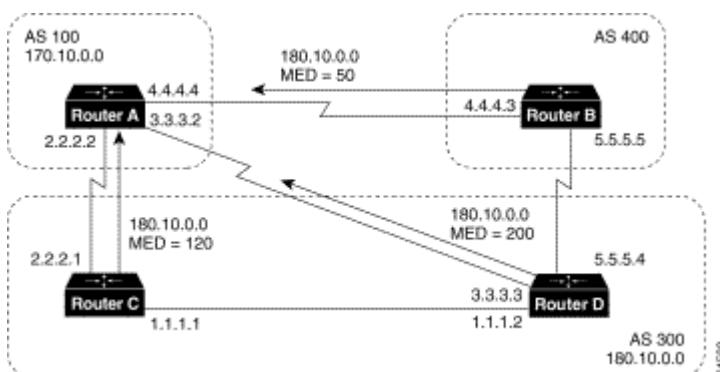
**Примечание:** в BGP Version 3 MED называется Inter-AS\_Metric.

В отличие от Local Preference (когда обмен Ipref'ами происходит в пределах одной AS) использование MED ведет к тому, что эта характеристика передается между автономными системами. Но - с одним ограничением, а именно - MED'ы передаются на \_соседнюю\_ AS, но не передаются за ее пределы (вернее, при передаче за пределы системы, получившей MED, это значение обнуляется).

Если отдельно не обговорено, роутер сравнивает MED attributes с путями от внешних neighbor'ов, которые принадлежат одной и той же AS (от C и D, например). Если же вы хотите сравнивать MED attributes между автономными системами, нужно в конфигурации роутера (на котором хотите включить такое сравнение) использовать следующую команду:

```
bgp always-compare-med.
```

**Figure 12-18: MED Example**



**Суть:** с помощью коэффициента MED, заданного на роутерах, которые анонсируют сеть 180.10.0.0 или просто передают update'ы о этой сети, пришедшие от своих соседей, можно указать наилучший путь к этой сети через один из роутеров.

Рассмотрим конфигурацию всех четырех роутеров.

```
!Router Arouter bgp 100neighbor 2.2.2.1 remote-as 300neighbor 3.3.3.3 remote-as 300neighbor 4.4.4.3 remote-as 400!Router Brouter bgp 400neighbor 4.4.4.4 remote-as 100neighbor 4.4.4.4 route-map SETMEDOUT outneighbor 5.5.5.4 remote-as 300!route-map SETMEDOUT permit 10set metric 50!Router Crouter bgp 300neighbor 2.2.2.2 remote-as 100neighbor 2.2.2.2 route-map SETMEDOUT outneighbor 5.5.5.5 remote-as 400neighbor 1.1.1.2 remote-as 300!route-map SETMEDOUT permit 10set metric 120!Router Drouter bgp 300neighbor 3.3.3.2 remote-as 100neighbor 3.3.3.2 route-map SETMEDOUT outneighbor 1.1.1.1 remote-as 300!route-map SETMEDOUT permit 10set metric 200
```

По умолчанию, BGP сравнивает параметр MED в таблице маршрутизации для маршрутов, пришедших от neighbor'ов, которые принадлежат одной и той же AS (не той, в которой находится роутер, а см. рис.). В нашем примере это AS 300. В этом случае роутер "A" может сравнивать MED attributes в роутах, пришедших от "C" (MED = 120) и "D" (MED = 200). И не обращает внимания на то, что update'ы, пришедшие от "B", имеют более низкий MED.

Итак, роутер "A" выберет роутер "C" как best path для достижения сети 180.10.0.0; Для того, чтобы заставить роутер "A" сравнивать MED'ы для update'ов с роутера "B" (который не принадлежит той же AS, что и C и D), используется команда **bgp always-compare-med:**

```
!Router Arouter bgp 100neighbor 2.2.2.1 remote-as 300neighbor 3.3.3.3 remote-as
300neighbor 4.4.4.3 remote-as 400bgp always-compare-med
```

Итак, после этого роутер "А" выберет путь к сети 180.10.0.0 через роутер "В", тк MED для путей, анонсируемых роутером "В" роутеру "А", наименьший. Конечно, это имеет место в том случае, когда все остальные атрибуты для сравнения путей (Ipref, weight, etc) являются одинаковыми. Вы можете также установить MED attribute и в случае redistributing routes into BGP. Например, на роутере "В" вы можете добавить static route в BGP со значением MED=50:

```
!Router Brouter bgp 400redistribute staticdefault metric 50!ip route 160.10.0.0
255.255.0.0 null 0
```

Данная конфигурация означает, что роутер "В" будет отправлять updates для 160.10.0.0 с MED attribute = 50.

## II.7 Community Attribute

**Community attribute** - это способ группирования маршрутов (communities), к которым может быть установлено одно из правил, таких как "acceptance", "preference", "redistribution". Как и MED Attribute, Community передается соседним neighbor'ам и служит для управления маршрутами к вашей AS.

Для установки community attribute используется route-map.

Несколько заранее определенных Community attributes содержатся в табл. 12-1.

**Table 12-1: Predefined Communities Community: Meaning: no-**

Community	Meaning
<b>export</b>	Не передавать эти маршруты EBGP peer'ам
<b>advertise</b>	Не передавать эти маршруты вообще
<b>internet</b>	Передавать эти пути в internet community; Все маршруты внутри сети относятся к этому community.

На примере, приведенном ниже, community attribute устанавливаются с помощью route-map.

```
route-map COMMUNITYMAPmatch ip address 1set community no-advertise!route-map
SETCOMMUNITYmatch as-path 1set community 200 additive
```

Если вы используете ключевое слово **"additive"**, то указанное значение 200 добавляется к уже существующему community attribute ("no-advertise", как было указано выше). В противном случае, новое значение community перезаписывало бы значение, которое было задано раньше. Для того, чтобы передать community attribute своему neighbor'у, вы должны использовать команду:

```
neighbor [ip address] send-community
```

А именно:

```
router bgp 100neighbor 3.3.3.3 remote-as 300neighbor 3.3.3.3 send-communityneighbor
3.3.3.3 route-map SETCOMMUNITY out
```

Подробнее об использовании community для фильтрации update'ов см. в главе "Community Filtering" ниже.

**Для просмотра примеров, как Community Attribute используется для фильтрации update'ов, смотрите section "[Community Filtering](#)," later in this chapter.**

## Summary of the BGP Path Selection Process

BGP всегда выбирает один из маршрутов - как наилучший. Когда маршрут выбран, BGP помещает его в свою таблицу маршрутизации, и передает его своим neighbor'ам. Для выбора наилучшего маршрута BGP использует следующие критерии:

1. Если путь, указанный в поле next-hop, недоступен, этот update убивается.
2. Лучшим считается путь с наибольшим значением weight;
3. Если weight путей одинаковы, то выбирается путь с наибольшим Ipref;
4. Если weight и Ipref одинаковы, то выбирается путь, который появился в таблице маршрутов BGP на данном роутере. (подразумевается, что это будет самый наикратчайший путь);
5. Если нет путей, которые "родились" на данном роутере, то выбирается путь с AS\_path минимальной длины;
6. Если все пути имеют одинаковую длину (одинаковое количество hop'ов, следовательно), выбирается путь с "lowest origin type". Считается, что IGP < EGP < Incomplete.;
7. Если Origin attr. code одинаковы для всех путей, выбирается путь с наименьшим MED attr.;
8. Если и MED одинаковы для всех возможных путей, предпочтительным считается external path перед internal path;
9. Если опять все пути одинаковы, то prefer the path through the closest IGP neighbor;
10. Выбирается путь с наименьшим IP адресом из тех, которые указываются by the BGP router ID.

### III. Controlling the Flow of BGP Updates

Сначала - лирическое отступление: в предыдущих главах рассматривались вопросы:

а) Основные понятия. Протокол BGP, таблица маршрутизации BGP, neighbor'ы. Взаимодействия роутеров, принадлежащих одной и той же, или разным автономным системам, для обмена маршрутами.

б) Выбор оптимального маршрута из имеющихся в таблице BGP маршрутизации. Факторы, влияющие на выбор оптимального маршрута.

В этой главе мы будем рассматривать способы, с помощью которых можно управлять BGP update'ами, которые пересылаются между автономными системами.

Итак,

В этой главе рассматривается механизм управления передачей BGP update'ов. Он включает в себя следующие части:

- [III.1 Administrative Distance](#)
- [III.2 BGP Filtering](#)
- [III.3 BGP Peer Groups](#)
- [III.4 CIDR and Aggregate Addresses](#)
- [III.5 Confederations](#)
- [III.6 Route Reflectors](#)
- [III.7 Route Flap Dampening](#)

#### III.1 Administrative Distance

Обычно, о некоем маршруте роутер может узнать из более чем одного протокола. Механизм "Administrative Distance" используется для распознавания путей, о которых роутер узнал, используя несколько протоколов.

Путь с меньшим значением administrative distance прописываются в IP routing table.

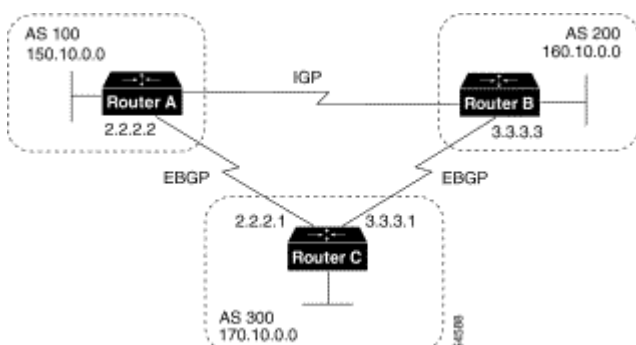
По умолчанию, BGP использует Administrative Distances из табл. 12-2.

Distance	Default value	Function	External	20
Присваивается путям, полученным из	EBGP	Internal		200
----- // -----	IGP	Local		200
				Пути, которые рождены на

**ЗАМЕЧАНИЕ:** Distance не влияет на алгоритм выбора BGP пути, но влияет на процесс записи путей, полученных по BGP, в IP routing table.

Обычно, если маршрут приходит через BGP, он прописывается в IP routing table, поскольку его "distance" = 20. Однако иногда две AS имеют как бы "back door" между собой, полученный по IGP, кроме пути, полученного по EBGP. При этом, с помощью определенных команд на одном из этих роутеров, можно управлять маршрутизацией таким образом, что наилучшим путем будет путь, полученный через IGP, а запасным путем (который будет использоваться в IP route table в случае падения основного пути) будет путь, полученный через BGP update.

Figure 12-19: Back Door Example



На данном рисунке роутеры "А" и "С", равно как и роутеры "С" и "В", взаимодействуют по EBGP. Роутеры "А" и "В" взаимодействуют по IGP (RIP, IGRP, Enhanced IGRP, or OSPF). Значения default distance для RIP, IGRP, Enhanced IGRP и OSPF равны 120, 100, 90 и 110 соответственно. Поскольку все эти default distance больше, чем def distance для EBGP (20), то по правилу, описанному выше, для в таблицу IP маршрутов на роутере "А" должен быть прописан маршрут, полученный по EBGP. Более детально: роутер "А" получает updates о сети 160.10.0.0 по двум протоколам: EBGP и IGP. Поскольку default distance для EBGP меньше, чем для IGP, то роутер "А" выберет путь к данной сети, полученный им от роутера "С". Но мы можем изменить эту картину (как раз в этом и заключается использование back door): например, если мы хотим, чтобы роутер "А" использовал

путь к 160.10.0.0, полученный по IGP от роутера "B", можно воспользоваться одним из трех приведенных ниже способов:

- \*) Изменить external distance у EBGP. Но это не рекомендуется, поскольку distance влияет на все updates между роутерами, и может случиться путаница при взаимодействии нескольких протоколов между собой;
- \*) Изменить distance у IGP. Не рекомендуется по той же причине;
- \*) Использовать конфигурацию BGP back door. Это правильно.

Итак, для использования IGP connection as back door, существует команда конфигурации роутера:

```
network [net] backdoor
```

С ее помощью сконфигурируем роутер "A" таким образом, что он будет в свою IP table прописывать update'ы, полученные от роутера "B".

```
!Router Arouter eigrp 10network 150.10.0.0router bgp 100neighbor 2.2.2.1 remote-as 300network 160.10.0.0 backdoor
```

Команда network 160.10.0.0 backdoor на роутере "A" выдает маршрут к сети 160.10.0.0, как локальный, и прописывает его в IP routing table со значением distance = 200. Поскольку сеть 160.10.0.0 также анонсируется и по Enhanced IGRP (с роутера "B", причем с distance=90) то именно полученный по Enhanced IGRP маршрут будет прописан в IP route table и будет использоваться как основной. Если же IGRP маршрут упадет, то EBGP маршрут будет прописан в IP routing table и будет использован для прохождения трафика.

**Примечание:** Несмотря на то, что BGP трактует сеть 160.10.0.0 как local, это не значит, что данная сеть анонсируется как локальная для данного роутера.

## III.2 BGP Filtering

Существует возможность управления отправкой/получением update'ов с использованием нескольких методов фильтрации BGP update'ов:

- [III.2.1 Prefix Filtering](#)
- [III.2.2 AS\\_path Filtering](#)
- [III.2.3 Route Map Filtering](#)
- [III.2.4 Community Filtering](#)

Любой из перечисленных методов может быть использован для достижения одного и того же результата. Выбор того, или иного метода зависит от вида сети, в котором его предполагается использовать.

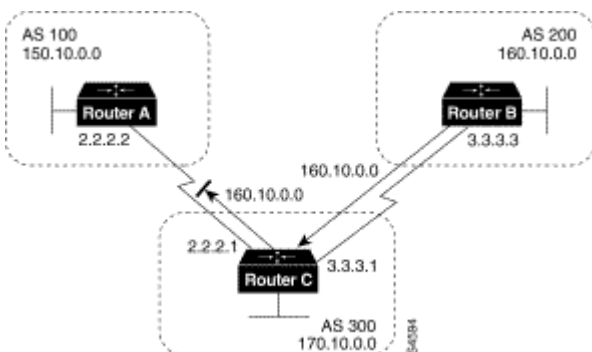
Рассмотрим все 4 метода filtering'a BGP update'ов по отдельности.

### III.2.1 Prefix Filtering

Для фильтрации информации о маршрутах, которые данный роутер получает, или информации, которую данный роутер, наоборот, анонсирует далее в сеть, можно использовать фильтр, основанный на routing updates, идущих к/от neighbor'ов.

На рисунке ниже мы видим сеть, которая демонстрирует использование prefix filtering.

**Figure 12-20: Route Filtering**



На данном рисунке роутер "B" является originated для сети 160.10.0.0 и он отправляет update'ы об этой сети на роутер "C". Если Вы хотите запретить передачу update'ов об этой сети с роутера "C" на роутер "A" AS100, вы можете воспользоваться access-list'ом:

```
!Router Crouter bgp 300network 170.10.0.0neighbor 3.3.3.3 remote-as 200neighbor 2.2.2.2 remote-as 100neighbor 2.2.2.2 distribute-list 1 out!access-list 1 deny 160.10.0.0.0.0.255.255access-list 1 permit 0.0.0.0 255.255.255.255
```

В этом примере конфигурации, как вы видите, комбинация команд `neighbor [ip] distribute-list 1 out` и `access-list 1` запрещает роутеру "C" передавать update'ы о сети 160.10.0.0 neighbor'у 2.2.2.2 AS100, но разрешает все остальные update'ы в его направлении.

Использование `access-list`'а для фильтрации сетей целиком делается просто. Но, предположим, что роутер "B" на приведенном выше рисунке имеет другую подсеть от 160.10.x.x, а вы хотите анонсировать только 160.0.0.0/8.

Access list:

```
access-list 1 permit 160.0.0.0 0.255.255.255
```

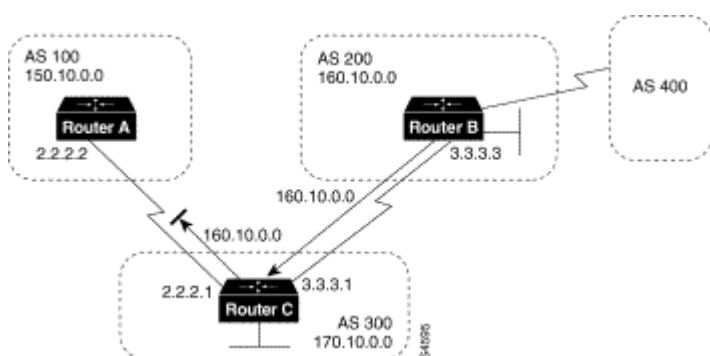
разрешает сети 160.0.0.0/8, 160.0.0.0/9 и так далее. Для того, чтобы роутер пропускал update'ы только о 160.0.0.0/8, следует использовать `extended access list`, например:

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

### III.2.2 AS\_path Filtering

**AS\_path filtering** - это возможность фильтровать приходящие, и уходящие из данного роутера update'ы, основываясь на значении AS\_path attribute.

Figure 12-21: AS\_path Filtering



```
!Router Cneighbor 3.3.3.3 remote-as 200neighbor 2.2.2.2 remote-as 100neighbor 2.2.2.2 filter-list 1 out!ip as-path access-list 1 deny ^200$ip as-path access-list 1 permit .*
```

В данном примере `access-list 1` на роутере "C" запрещает все update'ы, в которых AS\_path начинается и заканчивается на 200.

Итак, роутер "B" посылает updates о сети 160.10.0.0, у которой AS\_path attribute начинается и заканчивается на 200, и как только этот update получит Router "C", и сравнит со своим `access-list`'ом, он игнорирует этот update, то есть не передаст его на "A".

**ЗАМЕТИМ**, что update, который заканчивается на 200, будет разрешен данным `access-list`'ом. Таким образом, updates от AS 400, которые проходят через AS 200 (у них AS\_path = 200,400) не игнорируются. А вот если в правиле `access-list` написать ^200 вместо ^200\$, то эти updates не будут переданы на "A".

В правиле, в котором встречаются метасимволы . и \*, "." означает любой символ, а "\*" означает многократное его повторение. Так что это любой AS\_path.

Если Вы хотите проверить, работает ли введенное вами regular expression так, как предполагалось, используйте следующую EXEC команду:

```
show ip bgp regexp regular-expression
```

По выполнении этой команды роутер отобразит все пути, которые удовлетворяют данному регулярному выражению.

### III.2.3 Route Map Filtering

Команда конфигурации роутера **neighbor [ip] route-map** может быть использована для приложения route map к incoming и outgoing routes.

**Замечание:** команда `neighbor [ip] route-map` ничего не дает для incoming updates, если проверяется равенство IP адресов.

Предположим, что вы хотите, чтобы роутер "C" узнавал только о сетях, которые являются локальными для AS200, и больше не получал никаких маршрутов (т.е. чтобы "C" не знал о AS100, AS400, AS600 от AS200). Также, допустим вы хотите, чтобы все маршруты, которые роутер "C" получает от AS200, имели weight attribute равный 20. Это достигается следующей конфигурацией роутера "C":

```
!Router Crouter bgp 300network 170.10.0.0neighbor 3.3.3.3 remote-as 200neighbor 3.3.3.3
route-map STAMP in!route-map STAMP permit 10match as-path 1set weight 20!ip as-path
access-list 1 permit ^200$
```

В данной конфигурации access list 1 разрешает любые update'ы, чьи AS\_path начинаются и заканчиваются на 200 (то есть этот access list разрешает только update'ы, которые оригинальны для AS200). Все другие update'ы игнорируются.

Предположим, что для той же схемы, которая нарисована выше, мы хотим следующее:

!) Все так же принимать update'ы, оригинальные для AS200, и присваивать им weight attribute 20;

!) Не принимать update'ы, которые в AS\_path содержат AS400;

!) Принимать все остальные update'ы, и устанавливать им weight attribute = 10.

Это достигается такой конфигурацией:

```
!Router Crouter bgp 300network 170.10.0.0neighbor 3.3.3.3 remote-as 200neighbor 3.3.3.3
route-map STAMP in!route-map STAMP permit 10match as-path 1set weight 20!route-map
STAMP permit 20match as-path 2!route-map STAMP permit 30set weight 10!ip as-path
access-list 1 permit ^200$ip as-path access-list 2 deny _400_
```

В данной конфигурации роутера "С" указано, что access-list 1 разрешает любой update с AS\_path attribute, начинающимся и заканчивающимся на 200, что означает разрешение всех update'ов, для которых AS200 является originated. Weight attribute для разрешенных update'ов устанавливается равным 20.

Access-list 2 запрещает все update'ы, содержащие 400 в AS\_path. Все остальные update'ы будут иметь weight attribute = 10, как указано в правиле "30" route-map'a STAMP.

Далее, предположим, что на рис. выше Router "С" анонсирует свою собственную сеть 170.10.0.0 автономным системам AS100 и AS200. Когда update'ы об этой сети достигнут AS600, роутеры этой автономной системы получают два пути к нашей сети: через AS100 с AS\_path=100,300 и через AS400 с AS\_path=400,200,300. Допустим, что все остальные значения attributes, которые могут повлиять на выбор пути к нашей сети, одинаковы. В этом случае роутер в AS600 выберет тот путь, который короче, т.е. через AS100.

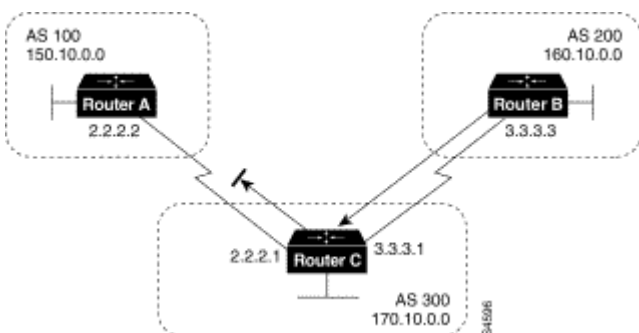
Если вы захотите сделать так, чтобы конфигурация роутера "С" влияла на выбор пути к сети 170.10.0.0, который будет производиться на роутере "D", вы можете это сделать... прибавив к AS\_path дополнительные номера AS, как правило просто повторив несколько раз номер AS300, в которую и входит роутер "С":

```
!Router Crouter bgp 300network 170.10.0.0neighbor 2.2.2.2 remote-as 100neighbor 2.2.2.2
route-map SETPATH out!route-map SETPATH permit 10set as-path prepend 300 300
```

Команда конфигурации роутера "С" set as-path prepend 300 300 указывает роутеру "С", что, прежде чем отправить update на 2.2.2.2, к пути AS\_path надо прибавить 300 дважды (искусственно удлинив путь). В результате, AS\_path для сети 170.10.0.0, которая достигнет AS600 через AS100, будет иметь вид: 100,300,300,300; а AS\_path, полученная AS600 через AS400, будет короче: 400,200,300 - она и будет выбрана роутером AS600 для доступа к сети 170.10.0.0.

### III.2.4 Community Filtering

Figure 12-23: Community Filtering



Предположим, что мы не хотим, чтобы роутер "С" передавал полученные от роутера "В" маршруты на роутер "А".

Это делается установкой Community Attribute update'ам, отправляемым роутером "В" на роутер "С" (делается на роутере "В"!!!!):

```
!Router Brouter bgp 200network 160.10.0.0neighbor 3.3.3.1 remote-as 300neighbor 3.3.3.1
send-communityneighbor 3.3.3.1 route-map SETCOMMUNITY out!route-map SETCOMMUNITY permit
```

```
10match ip address 1set community no-export!route-map SETCOMMUNITY permit 20!access-list 1 permit 0.0.0.0 255.255.255.255
```

Для маршрутов, которые посылаются роутеру "C" (neighbor с IP адресом 3.3.3.1), роутер "B" использует route-map с названием SETCOMMUNITY. Данный route-map для любого update'a, достигшего роутера "\_C\_", и удовлетворяющего правилу access-list 1, устанавливает community равным "no-export". Команда конфигурации роутера "B" send-community используется для передачи данного community attribute роутеру "C" (3.3.3.1).

Когда роутер "C" получает update'ы от роутера "B", он не передает их на роутер "A", поскольку community attribute = no-export.

Существует еще один способ использования фильтров, в которых используется community attribute: это команда конфигурации

### ip community-list

Итак,

```
!Router Brouter bgp 200network 160.10.0.0neighbor 3.3.3.1 remote-as 300neighbor 3.3.3.1 send-communityneighbor 3.3.3.1 route-map SETCOMMUNITY out!route-map SETCOMMUNITY permit 10match ip address 2set community 100 200 additive!route-map SETCOMMUNITY permit 20!access list 2 permit 0.0.0.0 255.255.255.255
```

При данной конфигурации, роутер "B" добавляет значения 100 и 200 к значению community любого update'a, который достигнет neighbor'a с IP адресом 3.3.3.1; Далее: для конфигурирования роутера "\_C\_" используется "ip community-list" команда, с помощью которой устанавливается "weight" в том случае, когда у пришедшего на роутер "C" update'a community attribute содержит в себе 100 и/или 200:

```
!Router Crouter bgp 300neighbor 3.3.3.3 remote-as 200neighbor 3.3.3.3 route-map check-community in!route-map check-community permit 10match community 1set weight 20!route-map check-community permit 20match community 2 exactset weight 10!route-map check-community permit 30match community 3!ip community-list 1 permit 100ip community-list 2 permit 200ip community-list 3 permit internet
```

В данной конфигурации, любой update, приходящий на "C" и содержащий 100 в community attribute, удовлетворяет правилу ip community-list 1 и в результате этому маршруту присваивается весовой коэффициент weight = 20. Аналогично, если в comm. attr. есть \_только\_ 200, то маршруту присвоится weight = 10. Последний же в списке community list (list 3) использует ключевое слово "internet", и разрешает все остальные update'ы без изменения из атрибутов.

**Note:** "internet" означает \_все\_ update'ы, поскольку все маршруты всегда являются членами community = internet.

## III.3 BGP Peer Groups

"BGP Peer Group" - это группа BGP neighbor'ов, которые используют одни и те же update policies.

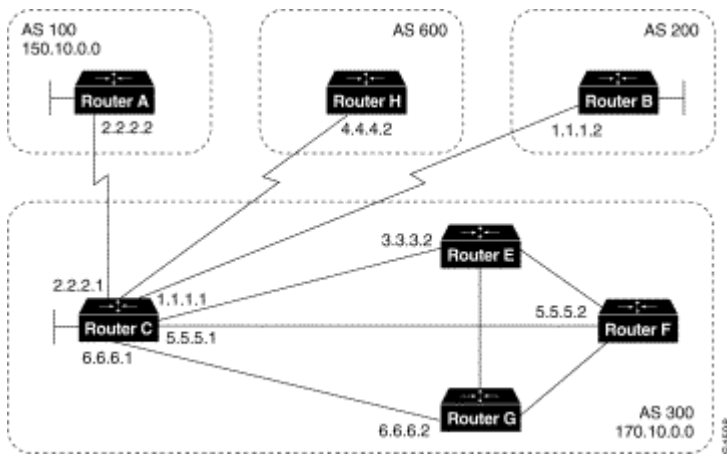
Update policies обычно устанавливаются через route maps, distribution lists, и filter lists.

Вместо того, чтобы определять одни и те же policies для каждого neighbor'a, вы можете создать peer group name и присвоить policies этой peer group.

Роутеры, входящие в peer group, наследуют все опции конфигурации этой самой peer group.

Роутеры в peer group могут также изменять эти опции, с условием что измененные значения не будут передаваться далее в outgoing updates. Это означает, что менять ту или иную опцию конфигурации можно только для incoming updates.

### Figure 12-24: BGP Peer Groups



Итак, следующая конфигурация роутера "С" включает в себя BGP peer group "INTERNALMAP" на данном роутере и используется во всех роутерах AS300:

```
!Router Crouter bgp 300!neighbor INTERNALMAP peer-groupneighbor INTERNALMAP remote-as 300neighbor INTERNALMAP route-map INTERNAL outneighbor INTERNALMAP filter-list 1 outneighbor INTERNALMAP filter-list 2 in!neighbor 5.5.5.2 peer-group INTERNALMAPneighbor 6.6.6.2 peer-group INTERNALMAPneighbor 3.3.3.2 peer-group INTERNALMAPneighbor 3.3.3.2 filter-list 3 in
```

Данная конфигурация определяет следующие policies для INTERNALMAP peer group:

- \*) Route map с названием INTERNAL
- \*) Filter list для outgoing updates (filter list 1)
- \*) Filter list для incoming updates (filter list 2)

В данной конфигурации peer group применяется ко всем \_внутренним\_ neighbor'ам - роутерам E, F и G. В ней также отдельно определено использование filter list'a для incoming updates от neighbor'a 3.3.3.2 (Router E). Этот filter list используется только для incoming updates.

Теперь рассмотрим другую конфигурацию роутера "С". В этой конфигурации создается BGP peer group name "EXTERNALMAP" на роутере "С" и она применяется к роутерам из AS100, 200, 600 :

```
!Router Crouter bgp 300!neighbor EXTERNALMAP peer-groupneighbor EXTERNALMAP route-map SETMEDneighbor EXTERNALMAP filter-list 1 outneighbor EXTERNALMAP filter-list 2 in!neighbor 2.2.2.2 remote-as 100neighbor 2.2.2.2 peer-group EXTERNALMAPneighbor 4.4.4.2 remote-as 600neighbor 4.4.4.2 peer-group EXTERNALMAPneighbor 1.1.1.2 remote-as 200neighbor 1.1.1.2 peer-group EXTERNALMAPneighbor 1.1.1.2 filter-list 3 in
```

Как видим, в этой конфигурации команда neighbor remote-as помещается вовне того блока, где прописывается neighbor ..... постольку поскольку используются несколько разных AS, а не одна, как в первом примере с peer-group.

Так же заметим, что в данной конфигурации определяется filter-list 3, который может быть использован для изменения (перезаписи) configuration options для incoming updates от neighbor'a 1.1.1.2 (Router B).

### III.4 CIDR and Aggregate Addresses

Протокол BGP4 поддерживает безклассовую внутримоментную маршрутизацию (Classless InterDomain Routing - CIDR), которая и является его главным преимуществом перед BGP3. CIDR также известен под названием supernetting.

**CIDR** - это новый способ определения IP адресов, который исключает (не использует) концепцию классов (Class A, B, C и тд). Например, сеть 192.213.0.0 с "неправильным" классом сети "С" (см адрес - в третьем октете не может стоять 0 для сети класса C), является совершенно легальной supernet, когда она представлена в CIDR notation как 192.213.0.0/16;

Запись "/16" показывает, что маска подсети состоит из 16 бит (считая слева).

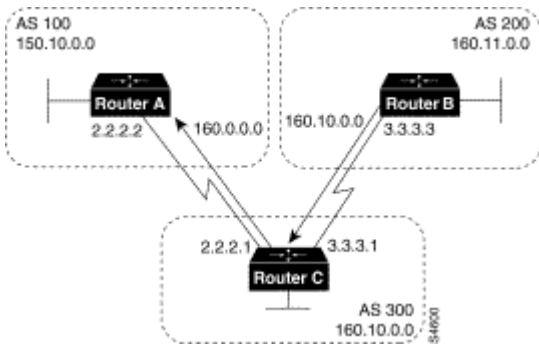
Заметим, что 192.213.0.0/16 и 192.213.0.0 255.255.0.0 - одно и то же.

**more specific route** - это маршрут на подблок.

CIDR делает простым aggregate маршрутов. Aggregation - это процесс преобразования нескольких различных маршрутов в один маршрут (prefix aggregate route), который и будет анонсироваться. Такой подход уменьшает размеры таблицы маршрутизации.

Figure 12-25: Aggregation





На этом рисунке роутер "B" из AS200 является originated для сети 160.11.0.0 и он анонсирует эту сеть роутеру "C" из AS300. Ниже приведена конфигурация роутера "C" для того, чтобы передавать aggregate address 160.0.0.0 роутеру "A":

```
!Router C router bgp 300 neighbor 3.3.3.3 remote-as 200 neighbor 2.2.2.2 remote-as 100 network 160.10.0.0 aggregate-address 160.0.0.0 255.0.0.0
```

Команда конфигурации роутера aggregate-address анонсирует prefix route (в данном случае, 160.0.0.0/8) и все more specific routes.

**Замечание:** роутер не может aggregate адреса, если он не имеет more specific route к этим адресам в своей таблице маршрутизации. More specific route могут появиться в таблице BGP маршрутизации из incoming updates от других AS, или могут быть получены по IGP (redistributed into BGP), или могут быть установлены с помощью команды конфигурации роутера "network".

Если вы хотите, чтобы роутер "C" передавал только prefix route, и не хотите, чтобы он передавал more specific route, используйте ключевое слово "summary only" в команде aggregate-address:

```
aggregate-address 160.0.0.0 255.0.0.0 summary-only
```

Эта команда передает prefix (160.0.0.0/8) и не передает more specific routes, которые данный роутер может иметь в BGP routing table.

**Замечание:** Если вы используете команду конфигурации "network" для анонсирования сети, запись о данной сети всегда будет появляться в BGP update'ax, даже если вы используете ключевое слово summary-only в команде конфигурации aggregate-address.

Если вы хотите запретить некоторые specific routes, когда принимаете маршруты, вы можете использовать route map. Например, если вы хотите, чтобы Router C принимал маршруты о 160.0.0.0 и запретить прием route 160.20.0.0, но все-таки разрешив маршруты к 160.10.0.0, используйте следующие команды:

```
!Router C router bgp 300 neighbor 3.3.3.3 remote-as 200 neighbor 2.2.2.2 remote-as 100 network 160.10.0.0 aggregate-address 160.0.0.0 255.0.0.0 suppress-map CHECK ! route-map CHECK permit 10 match ip address 1 ! access-list 1 deny 160.20.0.0 0.0.255.255 access-list 1 permit 0.0.0.0 255.255.255.255
```

Если вы хотите, чтобы на роутере устанавливалось значение некоторых attributes, когда он передает aggregate route, используйте attribute map:

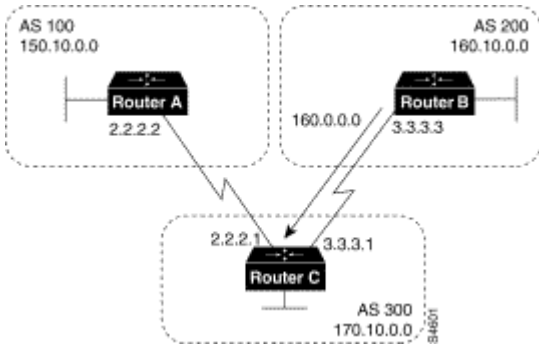
```
aggregate-address 160.0.0.0 255.0.0.0 attribute-map SETORIGIN ! route-map SETORIGIN permit 10 set origin igp
```

### III.4.1 Aggregation and Static Routes

Во-первых, заметим: Aggregation при Static routes конфигурируется на том роутере, на котором эти самые static routes прописаны.

На рисунке 12-26, приведенном ниже, показано как static routes могут быть использованы для создания aggregate'ов.

**Figure 12-26: CIDR Aggregation Example**



Допустим, вы хотите, чтобы роутер "B" анонсировал prefix 160.0.0.0, но не передавал все остальные more specific routes.

Конфигурация роутера "B", приведенная ниже, выполняет redistribute static aggregate into BGP:  
**!Router B router bgp 200 neighbor 3.3.3.1 remote-as 300 redistribute static ! ip route 160.0.0.0 255.0.0.0 null 0**

В результате этой конфигурации получаем, что роутер "B" анонсирует сеть 160.0.0.0 с origin attribute = Incomplete.

**Замечание:** вместо redistribute static можно использовать команду network дает тот же самый эффект, только origin attr для маршрута к сети 160.0.0.0 будет IGP, а не Incomplete:

**!Router B router bgp 200 neighbor 3.3.3.1 remote-as 300 network 160.0.0.0 mask 255.0.0.0 !**

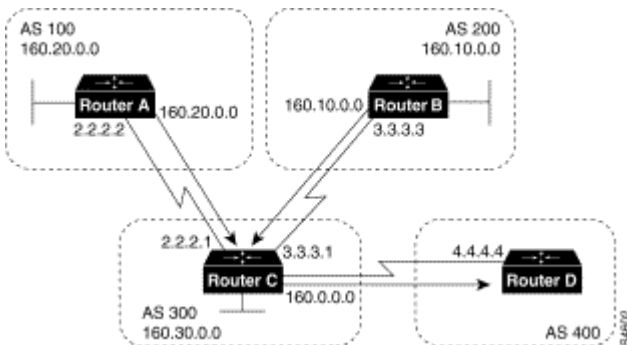
**ip route 160.0.0.0 255.0.0.0 null 0**  
**Замечание:** Использование static routes, как показано в двух последних примерах конфигурации (с помощью команд redistribute static и network) - наиболее предпочтительный способ добавления aggregate route, поскольку использование static routes предотвращает нежелательные route flaps.

### III.4.2 Aggregation and AS-SET

Когда aggregates (обобщенные маршруты, так сказать) генерируются из more specific routes, атрибуты AS\_path этих routes объединяются в так называемый AS\_SET. Это используется для предотвращения заикливания маршрутов.

На рисунке ниже демонстрируется использование AS-SET при агрегировании адресов.

**Figure 12-27: CIDR Aggregation Example with AS-SET**



На этом рисунке роутер "C" принимает updates о сети 160.20.0.0 от роутера "A" и updates о сети 160.10.0.0 от роутера "B". Если роутер "C" агрегирует сеть 160.0.0.0/8 и отправляет update'ы об этой сети роутеру "D", то AS\_path для этой сети будет указывать, что AS300 является \_origin\_ для нее.

Но в том случае, если роутер "D" имеет другой путь к AS100, update'ы (о сети 160.0.0.0/8) с "C" на "D" могут вести к заикливанию маршрутов.

Для того, чтобы этого не случилось, используется команда конфигурации роутера aggregate-address с ключевым словом "as-set":

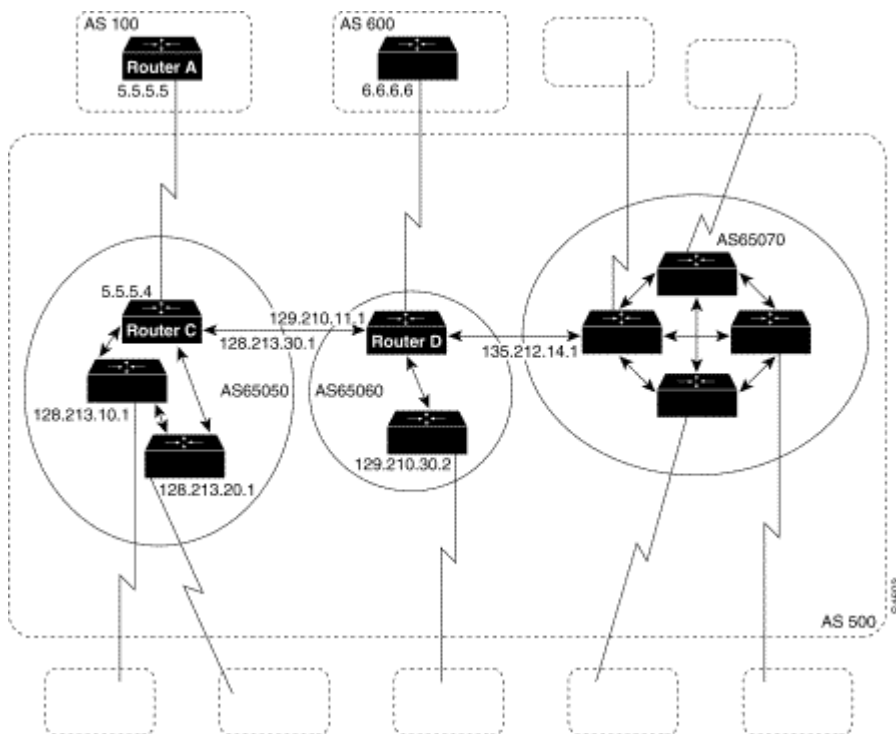
**!Router C neighbor 3.3.3.3 remote-as 200 neighbor 2.2.2.2 remote-as 100 neighbor 4.4.4.4 remote-as 400 aggregate-address 160.0.0.0 255.0.0.0 as-set**

Ключевое слово "as-set" указывает роутеру "C", что в update'ы о сети 160.0.0.0/8 должна быть включена информация о том, что сеть 160.0.0.0 принадлежит "to a set" (в данном случае, 100 и 200).

### III.5 Confederations

**Confederations** - это механизм уменьшения IBGP петель (связей) внутри автономной системы. На рисунке ниже роутеры без названий я обозначил просто как "Router".

**Figure 12-28: Confederations**



На данном рисунке AS500 содержит 9 роутеров (не считая тех, которые не работают по BGP). Без confederations, BGP будет требовать, чтобы роутеры, входящие в AS500, были полностью взаимосвязаны между собой. Это означает, что каждый роутер нуждается в том, чтобы у него было IBGP соединение с со всеми остальными 8 роутерами; Кроме того, пусть каждый из 9 роутеров связан по EBGP с external AS.

Confederation уменьшает количество peer'ов внутри AS. Это делается следующим образом: вы делите данную AS (AS500) на несколько автономных систем, и присваиваете каждой из них свой номер, номер mini-AS в "конфедерации". Каждая mini-AS является fully-meshed, то есть полностью взаимосвязанной (каждый из роутеров, входящих в данную mini-AS, связан по IBGP со всеми своими "соседями" в mini-AS). Каждая mini-AS связана с другой mini-AS в "конфедерации". Заметим, что несмотря на то, что mini-AS имеют EBGP peer'ы с другими AS внутри confederations, они обмениваются routing updates как если бы они использовали IBGP - т.е. next hop, MED, и local preference не меняются.

Для внешнего мира confederation выглядит как одна AS.

Следующие команды конфигурируют роутер "C":

```
!Router C
router bgp 65050
  bgp confederation identifier 500
  bgp confederation peers 65060 65070
  neighbor 128.213.10.1 remote-as 65050
  neighbor 128.213.20.1 remote-as 65050
  neighbor 129.210.11.1 remote-as 65060
  neighbor 135.212.14.1 remote-as 65070
  neighbor 5.5.5.5 remote-as 100
```

Здесь, команда конфигурации router bgp 65050 указывает, что роутер "C" принадлежит AS 65050. Команда bgp confederation identifier 500 указывает, что роутер "C" принадлежит "конфедерации" 500.

Первые две строчки, начинающиеся с neighbor, указывают на то, что роутер "C" устанавливает IBGP connection с двумя оставшимися роутерами в своей mini-AS 65050. Вторые две строчки устанавливают BGP connection с confederation peers 65060 и 65070. Последняя команда "neighbor" устанавливает EBGP connection с external AS100.

Аналогично, рассмотрим конфигурацию роутера "D":

```
!Router D
router bgp 65060
  confederation identifier 500
  bgp confederation peers 65050 65070
  neighbor 129.210.30.2 remote-as 65060
  neighbor 128.213.30.1 remote-as 65050
```

- роутер принадлежит AS65060  
 - роутер принадлежит конф-ии 500  
 - устанавливает IBGP connection внутри AS  
 - устанавливает BGP connection

`135.212.14.1 remote-as 65070` с peer'ами внутри "confederation"neighbor 6.6.6.6  
`remote-as 600` - устанавливает EBGP с внеш. AS 600.

Конфигурация роутера "А" выглядит следующим образом:

```
!Router Arouter bgp 100neighbor 5.5.5.4 remote-as 500
```

- то есть устанавливается EBGP connection с роутером "С".

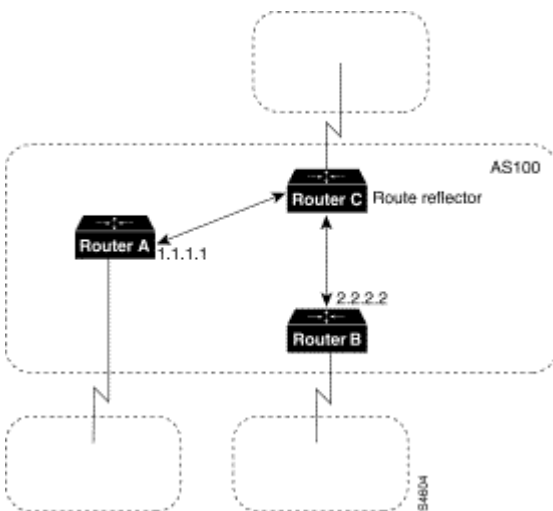
Заметим, что роутер "А" ничего не знает (и не должен знать) о делении AS500 на mini-AS, то есть он ничего не знает об AS65050, 65060, 65070. Он знает только об AS500.

### III.6 Route Reflectors

**Route Reflectors** - это другой способ для установления IBGP peering'a внутри AS.

Как ранее было сказано в секции Synchronization, BGP speaker'ы не передают маршруты, пришедшие от одного IBGP speaker'a - другому IBGP speaker'у. Route Reflectors позволяют обойти данное ограничение и позволяют роутеру анонсировать маршруты (reflect - в смысле отражать, отражать \_передавая дальше\_), полученные им от других speaker'ов по IBGP. При использовании Route Reflectors уменьшается количество IBGP speaker'ов внутри AS (вернее, взаимосвязей между ними - теперь они не должны быть связаны со всеми своими IBGP "соседями").

**Figure 12-29: Simple Route Reflector Example**



Итак, обратим внимание - все три роутера (А,В и С) принадлежат одной AS100.

Без использования Route Reflector'a для нормально функционирования сети, показанной на рисунке, требуется полная IBGP взаимосвязь (то есть роутер "А" должен быть связан еще и с роутером "В" по IBGP).

Но, если роутер "С" конфигурировать как Route reflector, то IBGP peering между роутерами "А" и "В" не требуется, поскольку роутер "С" будет передавать updates, пришедшие от роутера "А", на роутер "В", и наоборот.

Для конфигурирования роутера "С" в качестве Router Reflector'a используются следующие команды:

```
!Router Crouter bgp 100neighbor 1.1.1.1 remote-as 100neighbor 1.1.1.1 route-reflector-clientneighbor 2.2.2.2 remote-as 100neighbor 2.2.2.2 route-reflector-client
```

Итак, роутер, который включает в себя команду конфигурации neighborrouter-reflector-client - это Route Reflector.

Роутеры, ip адреса которых указаны в команде route-reflector-client - это клиенты Route Reflector'a.

Если рассматривать всю сеть целиком, то все вместе (Route Reflector и его client'ы) - называется "cluster".

IBGP peer'ы данной AS, которые не перечислены как clients, называются "nonclients".

В AS может быть несколько Route Reflect'оров. Когда это так, то каждый Route Reflect'ор общается с остальными Route Reflect'орами как с обычными IBGP speaker'ами.

В одном cluster'e может быть больше одного Route Reflect'ора, в одной AS может быть несколько cluster'ов.

*[рассмотрение более продвинутых конфигураций с использованием Route Reflector'ов, а так же способы защиты от зацикливаний при работе с Route Reflector'ами (Using an Originator*

ID, Using a Cluster List) пока опущены в настоящем рассмотрении. Причина: идея ясна, а рисунки слишком сложные для воспроизведения их в тексте; Да и времени маловато].

### III.7 Route Flap Dampening

*Route Flap Dampening - (начал использоваться в CISCO Internetwork Operating System [Cisco IOS] Release 11.0) - это механизм предотвращения нестабильности работы роутеров, основанный на flap dampening'e.*

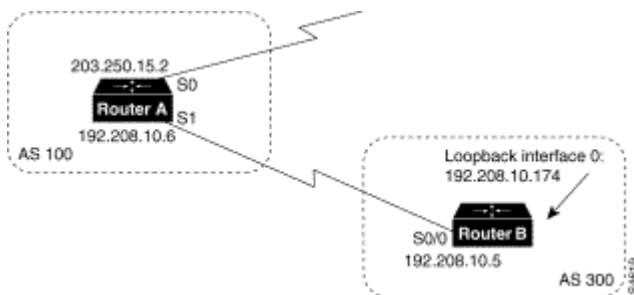
Для описания Route Flap Dampening'a используются следующие термины:

- Penalty** - Числовое значение, которое присваивается маршруту, когда роутер flaps
- Half-life time** - Числовое значение, которое устанавливается в конфигурации и означает время которое должно пройти с момента поднятия линка, и ведущее к уменьшению penalty вполнину.
- Suppress limit** - Числовое значение, с которым сравнивается penalty. Если penalty больше, чем suppress limit, то маршрут считается недоступным и отбрасывается.
- Suppressed** - маршрут, который не анонсируется, даже если он "UP". Маршрут считается Suppressed, если Penalty больше Suppress Limit.
- Reuse limit** - Числовое значение, устанавливаемое в процессе конфигурации. Это значение сравнивается с Penalty; Если Penalty меньше Reuse Limit, то Suppressed маршрут, если он UP в настоящий момент, больше не будет замораживаться (то есть считается размороженным).
- History entry** - запись, которая используется для сохранения flap information о маршрутах, которые down.

Маршруты, которые flapping, получают 1000 penalty за каждый flap. Когда накопленный таким образом penalty превысит установленный в конфигурации предел (Suppress Limit), BGP приостанавливает анонсирование данного маршрута, даже если в этот момент маршрут является рабочим (я его назвал UP). Накопленный penalty уменьшается вдвое по истечении half-life time. Если после этого оказывается, что penalty < reuse limit, маршрут начинает анонсироваться опять (если он UP в этот момент, конечно).

**Замечание:** Dampening не применяется к маршрутам, которые получены по IBGP. Это ограничение предотвращает forwarding loops и как бы защищает IBGP peer'ы от имеющих большие penalty у маршрутов, которые являются внешними к данной AS.

**Figure 12-33: Route Flap Dampening**



Рассмотрим команды конфигурации роутеров "A" и "B":

```
!Router Ahostname RouterA!interface serial0ip address 203.250.15.2
255.255.255.252interface serial1ip address 192.208.10.6 255.255.255.252!router bgp
100bgp dampening!network 203.250.15.0neighbor 192.208.10.5 remote-as 300!Router
Bhostname RouterB!interface loopback 0ip address 192.208.10.174
255.255.255.192!interface serial 0/0ip address 192.208.10.5 255.255.255.252!router bgp
300network 192.208.10.0neighbor 192.208.10.6 remote-as 100
```

Итак, роутер "A" сконфигурирован для route dampening'a. Предположим теперь, что EBGP линк к роутеру "B" стабилен. В этом случае BGP table на роутере A будет выглядеть так:

```
Q/ERROR: Опечатка в оригинальной ссылке на английском. см. след.строчку ниже
(там стоит RouterB#).RouterA# show ip bgp table version is 24, local router ID is
203.250.15.2Status codes: s suppressed, d damped, h history, * valid, > best, i -
internalOrigin codes: i - IGP, e - EGP, ? - incomplete Network Next
Hop Metric LocPrf Weight Path*> 192.208.10.0 192.208.10.5
0 0 0 300 i*> 203.250.15.0 0.0.0.0
0 32768 i
```

Для того, чтобы симитировать route flap, на роутере "B" введем команду: clear ip bgp 192.208.10.6  
Теперь таблица BGP маршрутов на роутере "A" будет выглядеть так:

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

Поскольку маршрут к сети 192.208.10.0 has flapped, запись в BGP table как бы удаляется, и к ней добавляется признак h - history.

Для более подробного просмотра некоторого отдельного маршрута (например, к сети 192.208.10.0) выполним команду:

```
RouterA# show ip bgp 192.208.10.0BGP routing table entry for 192.208.10.5
255.255.255.0, version 25Paths: (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, externalDampinfo:
penalty 1000, flapped 1 times in 0:02:03
```

Итак, мы видим, что данный маршрут имеет penalty 1000 из-за flapping, но penalty все еще меньше suppress limit'a (default 2000). Поскольку маршрут down, он помечен как "h" - history entry. Если маршрут flaps несколько раз, команда show ip bgp отображает следующее:

```
RouterA# show ip bgp table version is 32, local router ID is 203.250.15.2Status codes:
s suppressed, d damped, h history, * valid, > best, i - internalOrigin codes: i - IGP,
e - EGP, ? - incomplete      Network          Next Hop          Metric          LocPrf
Weight  Path*d 192.208.10.0    192.208.10.5    0
i*> 203.250.15.0    0.0.0.0        0                32768          i          300
```

И теперь вывод show ip bgp 192.208.10.0 будет следующий:

```
RouterA# show ip bgp 192.208.10.0BGP routing table entry for 192.208.10.5
255.255.255.0, version 32Paths: (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,
externalDampinfo: penalty 2615, flapped 3 times in 0:05:18, reuse in 0:27:00
```

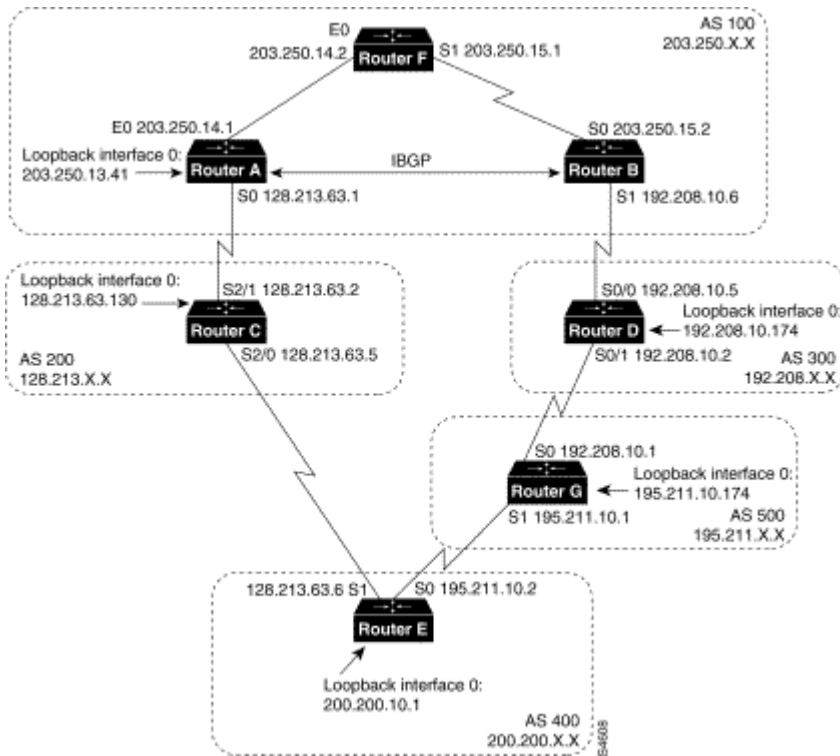
Итак, route is UP, но поскольку penalty > suppress limit, в данный момент маршрут suppressed. Этот маршрут будет восстановлен в таблице маршрутизации, когда penalty станет равным reuse limit (default 750).

Кроме того, dampening information будет очищена после того, как penalty станет меньшим, чем 1/2 от reuse limit (750/2).

## IV. Practical Design Example

На рисунке ниже представлена BGP сеть, на которой рассматриваются types of topologies, типичные для Internet Service Providers.

**Figure 12-34: Practical Design Example for ISPs**



Если AS связана с двумя ISPs через EBGP, IBGP должен использоваться между роутерами внутри данной AS для лучшего управления маршрутами.  
 Рассмотрим AS100, имеющую два EBGP connection (роутеры "A" и "B") с внешним миром (роутеры "C" и "D"). "A" и "B" общаются между собой по IBGP.  
 Между роутерами "A-F", "A-B" и "B-F" этой AS также используется OSPF (протокол семейства IGP). Приведенная ниже конфигурация роутеров - предварительная, поскольку она не полная. Это сделано для того, чтобы продемонстрировать методы BGP troubleshooting'a. Полная и правильная конфигурация расположена в главе "[Final Configurations](#)," ниже.

```
!Router Ahostname RouterA!interface loopback 0ip address 203.250.13.41
255.255.255.0!interface ethernet 0ip address 203.250.14.1 255.255.255.0!interface
serial 0ip address 128.213.63.1 255.255.255.252!router ospf 10network 203.250.0.0
0.0.255.255 area 0router bgp 100network 203.250.13.0 mask 255.255.255.0network
203.250.14.0 mask 255.255.255.0neighbor 128.213.63.2 update-source loopback 0 !Router
B!hostname RouterB!interface serial 0ip address 203.250.15.2 255.255.255.252!inetrface
serial 1ip address 192.208.10.6 255.255.255.252!router ospf 10network 203.250.0.0
0.0.255.255 area 0!router bgp 100network 203.250.15.0neighbor 192.208.10.5 remote-as
300neighbor 203.250.15.1 remote-as 100 !Router Chostname RouterC!interface loopback
0ip address 128.213.63.130 255.255.255.192!interface serial 2/0ip address 128.213.63.5
255.255.255.252!interface serial 2/1ip address 128.213.63.2 255.255.255.252!router bgp
200network 128.213.0.0neighbor 128.213.63.1 remote-as 100neighbor 128.213.63.6 remote-
as 400 !Router Dhostname RouterD!interface loopback 0ip address 192.208.10.174
255.255.255.192!interface serial 0/0ip address 192.208.10.5 255.255.255.252!interface
serail 0/1 (ERROR: здесь и строчка ниже в оригинале - с опечаткой)ip address
192.208.10.2 255.255.255.252!router bgp 300network 192.208.10.0neighbor 192.208.10.1
remote-as 500neighbor 192.208.10.6 remote-as 100 !Router Ehostname RouterE!interface
loopback 0ip address 200.200.10.1 255.255.255.0!interface serial 0ip address
195.211.10.2 255.255.255.252!interface serial 1ip address 128.213.63.6
255.255.255.252!router bgp 400network 200.200.10.0neighbor 128.213.63.5 remote-as
200neighbor 195.211.10.1 remote-as 500 !Router F!hostname RouterF!interface ethernet
0ip address 203.250.14.2 255.255.255.0!interface serial 1ip address 203.250.15.1
255.255.255.252!router ospf 10network 203.250.0.0 0.0.255.255 area 0 !Router Ghostname
RouterG!interface loopback 0ip address 195.211.10.174 255.255.255.192!interface serial
0ip address 192.208.10.0 255.255.255.252!interface serial 1ip address 195.211.10.1
255.255.255.252!router bgp 500network 195.211.10.0neighbor 192.208.10.2 remote-as
300neighbor 195.211.10.2 remote-as 400
```

Когда вы конфигурируете redistribute IGP routes в BGP, вам необходимо контролировать те маршруты, которые попадают в BGP. По этой причине всегда лучше использовать команду

конфигурации "network" или выполнять redistributing of static routes, как и показано на приведенной выше конфигурации. Эти методы так же предотвращают route flaps.

## IV.1 Determining the State of BGP

Предположим, что (см. приведенный выше рисунок) связь между роутерами "B" и "D" испортилась. Выполним на роутере "B" команду show ip bgp:

```
RouterB# show ip bgp table version is 4, local router ID is 203.250.15.2
Status codes: s
suppessed, 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
```

Символ "i" в начале строки означает, что о данном маршруте стало известно от IBGP peer'a.

Символ "i" в конце строки означает, что информация о данном пути пришла от IGP.

Первая строка читается так:

Информация о доступности сети 128.213.0.0 получена через AS\_path 200, и для того, чтобы с данного роутера достичь этой сети, в качестве Next hop'a следует использовать 128.213.63.2. Замечание: любой маршрут, который сгенерирован на данном роутере (см. 203.250.15.0) имеет next hop = 0.0.0.0.

Символ ">" означает, что BGP выбрал данный маршрут, как лучший. Процесс выбора наилучшего маршрута описан выше в главе "Summary of the BGP Path Selection Process". BGP всегда выбирает только один маршрут, как лучший. После чего он записывает этот маршрут в IP routing table и анонсирует этот путь другим BGP peer'am.

Заметим, что next hop attribute 128.213.63.2, имеющий место для части маршрутов, унаследован от EBGP.

Теперь проверим IP routing table на роутере "B":

```
RouterB# 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
not set          203.250.13.0 255.255.255.255 is subnetted, 1 subnetsO
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 subnetsC          203.250.15.0 is directly connected,
Serial00          203.250.14.0 [110/74] via 203.250.15.1, 02:40:46, Serial0
```

Заметим, что ни один BGP маршрут не появился в IP routing table. Это произошло потому, что в данной конфигурации мы имеем одну проблему: маршруты к некоторым сетям, содержащиеся в BGP route table на "B", имеют next hop = 128.213.63.2, который недоступен с "B".

Адрес 128.213.63.2 недоступен потому, что в таблице маршрутизации на "B" отсутствует запись о том, как достичь данный адрес через IGP (в данном случае, через OSPF). Итак, роутер "B" не знает о 128.213.63.0 из OSPF.

## IV.2 Correcting Next Hop Problems

В данном примере проблема с next hop может быть решена двумя способами:

\* Использование на роутере "A" команды "next-hop-self" для изменения значения next hop между роутерами "A" и "B".

\* На роутере "A" настроить OSPF \_на интерфейсе\_ Serial 0, указав его как passive. В этом случае роутер "B" будет знать, каким образом достичь next hop 128.213.63.2.

Итак, следующая конфигурация роутера "A" устанавливает OSPF на Serial 0 и делает его passive:

```
!Router A hostname RouterA ! interface loopback 0 ip address 203.250.13.41
255.255.255.0 ! interface ethernet 0 ip address 203.250.14.1 255.255.255.0 !
interface serial 0 ip address 128.213.63.1 255.255.255.252 ! router ospf 10
passive-interface serial 0 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 203.250.13.0 mask
255.255.255.0 network 203.250.14.0 mask 255.255.255.0 neighbor 128.213.63.2 remote-
```



```
as 200 neighbor 203.250.15.2 remote-as 100 neighbor 203.250.15.2 update-source
loopback 0
```

Теперь, BGP neighbor table на роутере "B" будет содержать следующие маршруты:

```
RouterB# show ip bgp table version is 4, local router ID is 203.250.15.2Status codes: s
suppesed, d damped, h history, * valid, > best, i internalOrigin 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
```

Как видно, символ ">" появился у всех записей о маршрутах, и это означает, что BGP удовлетворен наличием достижимого next hop'a в этих записях.

Теперь IP Routing Table на роутере "B" будет выглядеть по-другому:

```
RouterB# 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
not set 203.250.13.0 255.255.255.255 is subnetted, 1 subnetsO
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 subnetsC 203.250.15.0 is directly connected,
Serial00 203.250.14.0 [110/74] via 203.250.15.1, 02:40:46, Serial0
128.213.0.0 255.255.255.252 is subnetted, 1 subnetsO 128.213.63.0 [110/138]
via 203.250.15.1, 00:04:47, Serial0
```

Пока что мы добились лишь того, что сеть 128.213.63.0 стала доступной по OSPF. Заметим, что BGP entries все еще не появились в IP routing table.

Проблема заключается в synchronization: BGP не синхронизован с IGP, поэтому маршруты BGP не передались в IP routing table, и соответственно данные маршруты не включены в передаваемые далее BGP update'ы.

Роутер "F" не знает о сетях 192.208.10.0, 195.211.10.0 потому что BGP routes все еще не redistributed into OSPF.

### IV.3 Turning Off Synchronization

Если вы введете команду конфигурации "no synchronisation" на роутере "B", и потом проверите таблицу IP маршрутизации на нем же, то увидите в ней следующие маршруты:

```
RouterB# 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
not set B 200.200.10.0 [200/0] via 128.213.63.2, 00:01:07B 195.211.10.0 [200/0] via
128.213.63.2, 00:01:07B 192.208.10.0 [200/0] via 128.213.63.2, 00:01:07
203.250.13.0 is variably subnetted, 2 subnets, 2 masksO 203.250.13.41
255.255.255.255 [110/75] via 203.250.15.1, 00:12:37, Serial 0B
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 subnetsC 203.250.15.0 255.255.255.252 is directly
connected, Serial 00 203.250.14.0 [110/74] via 203.250.15.1, 00:12:37, Serial 0
128.213.0.0 is is variably subnetted, 2 subnets, 2 masksB 128.213.0.0 255.255.0.0
[200/0] via 128.213.63.2, 00:01:08O 128.213.63.0 255.255.255.252 [110/138]
via 203.250.15.1, 00:12:37, Serial 0
```

Итак, таблица маршрутизации на первый взгляд правильная, но достичь указанные в ней сети не представляется возможным из-за того, что роутер "F", расположенный на пути к ним, не знает маршрутов к этим сетям. Это видно в результатах выполнения команды show ip route на "F":

```
RouterF# 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
```

```
not set      203.250.13.0 255.255.255.255 is subnetted, 1 subnetsO
203.250.13.41 [110/11] via 203.250.14.1, 00:14:15      203.250.15.0 255.255.255.252 is
subnetted, 1 subnetsC      203.250.15.0 is directly connected, Serial1C
203.250.14.0 is directly connected, Ethernet0      128.213.0.0 255.255.255.252 is
subnetted, 1 subnetsO      128.213.63.0 [110/74] via 203.250.14.1, 00:14:15,
Ethernet0
```

Если пакеты, пришедшие из сети, роутеры которой обмениваются маршрутами по BGP, попадут на роутер "F", они будут потеряны. Таким образом, выключение synchronization не решает эту проблему. Мы видим, что OSPF необходимо сделать redistribute своих маршрутов в BGP на роутере "A"; таким образом, роутер "F" узнает о BGP маршрутах.

#### IV.4 Redistributing OSPF

Итак, следующая конфигурация роутера "A" модифицирована таким образом, что BGP маршруты передаются (redistributed) в OSPF:

```
!Router Ahostname RouterA!interface loopback 0ip address 203.250.13.41
255.255.255.0!interface ethernet 0ip address 203.250.14.1 255.255.255.0!interface
serial 0ip address 128.213.63.1 255.255.255.252!router ospf 10redistribute bgp 100
metric 2000 subnetspassive-interface serial 0network 203.250.0.0 0.0.255.255 area
0network 128.213.0.0 0.0.255.255 area 0!router bgp 100network 203.250.0.0 mask
255.255.0.0neighbor 128.213.63.2 remote-as 200neighbor 203.250.15.2 remote-as
100neighbor 203.250.15.2 update-source loopback 0
```

Теперь IP routing table будет выглядеть следующим образом:

```
RouterB# 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 defaultGateway of last resort not set O E2
200.200.10.0 [110/2000] via 203.250.15.1, 00:00:14, Serial0O E2 195.211.10.0 [110/2000]
via 203.250.15.1, 00:00:14, Serial0O 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 masksO
203.250.13.41 255.255.255.255      [110/75] via 203.250.15.1, 00:00:15, Serial0O
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 subnetsC
203.250.15.8 is directly connected, Loopback1 C      203.250.15.0 is directly
connected, Serial0O      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 masksO E2
128.213.0.0 255.255.0.0 [110/2000] via 203.250.15.1, 00:00:15, Serial0O
128.213.63.0 255.255.255.252      [110/138] via 203.250.15.1, 00:00:16, Serial0
```

Теперь записи о BGP маршрутах пропали, поскольку OSPF имеет лучшее значение Administrative Distance (110), чем IBGP (200).

Отключение синхронизации на роутере "A" означает, что роутер "A" анонсирует маршруты к сети 203.250.15.0; Это требуется потому, что роутер "A" не синхронизован с OSPF из-за mask differences. По той же самой причине, синхронизация должна быть отключена на роутере "B", чтобы этот роутер мог анонсировать сеть 203.250.13.0.

Добавим, что OSPF должен быть включен на интерфейсе Serial 1 роутера "B" и быть passive, таким образом роутер "A" узнает о next hop 192.208.10.5 через IGP.

Итак, новые конфигурации роутеров "A" и "B":

```
!Router Ahostname RouterA!interface loopback 0ip address 203.250.13.41
255.255.255.0!interface ethernet 0ip address 203.250.14.1 255.255.255.0!interface
serial 0ip address 128.213.63.1 255.255.255.252!router ospf 10redistribute bgp 100
metric 2000 subnetspassive-interface serial 0network 203.250.0.0 0.0.255.255 area
0network 128.213.0.0 0.0.255.255 area 0!router bgp 100no synchronizationnetwork
203.250.13.0 mask 255.255.255.0network 203.250.14.0 mask 255.255.255.0neighbor
128.213.63.2 remote-as 200neighbor 203.250.15.2 remote-as 100neighbor 203.250.15.2
update-source loopback 0
```

Конфигурация роутера "B":

```
!Router Bhostname RouterB!interface serial 0ip address 203.250.15.2
255.255.255.252!interface serial 1ip address 192.208.10.6 255.255.255.252!router ospf
10redistribute bgp 100 metric 1000 subnetspassive-interface serial 1network 203.250.0.0
```

```
0.0.255.255 area 0network 192.208.0.0 0.0.255.255 area 0!router bgp 100network
203.250.15.0neighbor 192.208.10.5 remote-as 300neighbor 203.250.13.41 remote-as 100
```

Теперь поднимем Serial 1 на роутере "B" и получим такую таблицу BGP маршрутизации на роутере "A":

```
RouterA# show ip bgp table version is 117, local router ID is 203.250.13.41Status
codes: s suppressed, d damped, h history, * valid, > best, i - internalOrigin codes: i
- IGP, e - EGP, ? - incomplete      Network          Next Hop          Metric LocPrf
Weight Path *> 128.213.0.0          128.213.63.2          0      100      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*> 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
```

Результаты выполнения команды show ip route на роутере "A":

```
RouterA# show ip routeCodes: 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 defaultGateway of last resort not set
192.208.10.0 is variably subnetted, 2 subnets, 2 masksO E2      192.208.10.0
255.255.255.0      [110/1000] via 203.250.14.2, 00:41:25, Ethernet0O      192.208.10
4 255.255.255.252      [110/138] via 203.250.14.2, 00:41:25, Ethernet0C
203.250.13.0 is directly connected, LoopbackO      203.250.15.0 is variably subnetted, 3
subnets, 3 masksO      203.250.15.10 255.255.255.255      [110/75] via
203.250.14.2, 00:41:25, Ethernet0O      203.250.15.0 255.255.255.252      [110/
74] via 203.250.14.2, 00:41:25, Ethernet0B      203.250.15.0 255.255.255.0 [200/0] via
203.250.15.2, 00:41:25C      203.250.14.0 is directly connected, Ethernet0
128.213.0.0 is variably subnetted, 2 subnets, 2 masksB      128.213.0.0 255.255.0.0
[20/0] via 128.213.63.2, 00:41:26C      128.213.63.0 255.255.255.252 is directly
connected, Serial0B*      200.200.0.0 255.255.0.0 [20/0] via 128.213.63.2, 00:02:38
```

Результаты выполнения команды show ip bgp на роутере "B":

```
RouterB# show ip bgp table version is 12, local router ID is 203.250.15.2Status codes: s
suppressed, d damped, h history, * valid, > best, i - internalOrigin 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*>
195.208.10.0          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
```

## IV.5 Managing Asymmetry

Существует несколько путей для соединения сетей AS100 с сетями ISP'ов в AS200 и AS300. Один путь это - подключение через primary ISP и наличие backup ISP. AS100 может получать часть маршрутов от одного ISP, и иметь default routes к обоим ISP. В этом случае, AS100 получает часть маршрутов от AS200 и только local routes от AS300. Оба роутера, "A" и "B", generate default routes в OSPF, с более предпочтительным маршрутом к роутеру "B", поскольку он имеет меньший MED attribute. Это позволяет управлять outgoing traffic'ом к двум граничным с вами ISP. Потенциальная асимметрия может случиться в том случае, если трафик, уходящий через роутер "A", вернется назад через роутер "B". Это может случиться в том случае, если вы анонсируете свои сети обоим ISP. Вне вашей AS ваши сети будут достижимы через любого из ваших провайдеров - то есть в конечном итоге к вашим сетям трафик может идти как через "A", так и через "B". В этом случае вы можете обнаружить, что весь incoming traffic в вашу AS идет только через одного провайдера, несмотря на то, что на самом деле их двое. Одна потенциальная причина этого явления - различная длина AS\_path для достижения вашей AS извне. Например, трафик с AS400 всегда будет приходиться через AS100 - потому что эта AS\_path короче. Вы можете изменить это с помощью команды route-map, используя ключевое слово

"prepend" для искусственного удлинения AS\_path. Но, если на AS400 каким-либо образом удастся установить exit point через AS200 (с помощью lpref, MED, weight), то вы уже ничего не сможете изменить.

## IV.5 Final Configurations

Итак, заключительная конфигурация роутера "A":

```
!Router Ahostname RouterA!interface loopback 0ip address 203.250.13.41
255.255.255.0!interface ethernet 0ip address 203.250.14.1 255.255.255.0!interface
serial 0ip address 128.213.63.1 255.255.255.252!router ospf 10redistribute bgp 100
metric 2000 subnetspassive-interface serial 0network 203.250.0.0 0.0.255.255 area
0network 128.213.0.0 0.0.255.255 area 0default-information originate metric 2000!router
bgp 100no synchronizationneighbor 128.213.63.2 remote-as 200neighbor 128.213.63.2
route-map setlocalpref inneighbor 203.250.15.2 remote-as 100neighbor 203.250.15.2
update-source loopback 0!ip default-network 200.200.0.0!route-map setlocalpref permit
10set local-preference 200
```

Заключительная конфигурация роутера "A" устанавливает lpref для маршрутов, пришедших от AS200, равным 200. В данной конфигурации также используется команда ip default-network 200.200.0.0 для указания сети 200.200.0.0 как кандидата в default route.

Команда конфигурации роутера default-information originate metric 2000 используется для добавления default route в OSPF domain. Для RIP, сеть 0.0.0.0 автоматически redistributed в RIP без дополнительных команд конфигурации. Для IGRP и Enhanced IGRP, default information добавляется в IGP domain после того, как BGP redistributed. Также, с IGRP и Enhanced IGRP, вы можете redistribute a static route для 0.0.0.0 в IGP domain.

Ниже приведена конфигурация роутера "B":

```
!Router Bhostname RouterB!interface serial 0ip address 203.250.15.2
255.255.255.252!interface serial 1ip address 192.208.10.6 255.255.255.252!router ospf
10redistribute bgp 100 metric 1000 subnetspassive-interface serial 1network 203.250.0.0
0.0.255.255 area 0network 192.208.0.0 0.0.255.255 area 0default-information originate
metric 1000!router bgp 100no synchronizationnetwork 203.250.15.0neighbor 192.208.10.5
remote-as 300neighbor 192.208.10.5 route-map LOCALONLY inneighbor 203.250.13.41 remote-
as 100!ip default-network 192.208.10.0ip as-path access-list 1 permit ^300 500$ip as-
path access-list 2 permit ^300$!route-map LOCALONLY permit 10match as-path 1set local-
preference 300!route-map LOCALONLY permit 20match as-path 2
```

В конфигурации роутера "B" устанавливается lpref=300 для update'ов, пришедших из AS300 и имеющих AS\_path=300,500 или AS\_path=300; Этот lpref больше, чем lpref у IBGP update'ов, пришедших от роутера "A". Поэтому AS100 будет выбирать Router B для маршрутов к AS500. Все остальные маршруты на роутере "B" будут отправлены по внутренним маршрутам с lpref=100, который меньше, чем lpref 200 у маршрутов, пришедших с роутера "A". Все это позволяет использовать роутер "A", как наиболее предпочтительный. В дальнейшем, в связи с длиной AS\_path Attribute, роутер "B" используется для достижения сетей в AS300.

Заметим, что роутер "B" принимает только маршруты от AS300 и AS500. Update'ы, чьи AS\_path не содержат данный AS, удаляются. Если вы хотите анонсировать local routes и neighbor routes (клиентов ISP), вы можете использовать ^300\_[0-9]\* в качестве регулярного выражения.

```
RouterB# show bgp regexp ^300$ BGP table version is 14, local router ID is
203.250.15.2Status code: s suppressed, d damped, h history, * valid, > best, i -
internalOrigin codes: i - IGP, e - EGP, ? - incomplete Network Next
Hop Metric LocPrf Weight Path*> 192.208.10.0 192.28.10.5 0
300 0 300
```

Рассмотрим теперь окончательную конфигурацию роутера "C":

```
!Router Chostname RouterC!interface loopback 0ip address 128.213.63.130
255.255.255.192!interface serial 2/0ip address 128.213.63.5 255.255.255.252!interface
serial 2/1ip address 128.213.63.2 255.255.255.252!router bgp 200network
128.213.0.0aggregate-address 128.213.0.0 255.255.0.0 summary-onlyneighbor 128.213.63.1
remote-as 100neighbor 128.213.63.1 distribute-list 1 outneighbor 128.213.63.6 remote-as
400!access-list 1 deny 195.211.0.0 0.0.255.255access-list 1 permit any
```

Конфигурация роутера "C" анонсирует сеть 128.213.0.0/16 и указывает маршруты, которые будут переданы в AS100.

Окончательная конфигурация роутера "D":

```
!Router Dhostname RouterD!interface loopback 0ip address 192.208.10.174
255.255.255.192!interface serial 0/0ip address 192.208.10.5 255.255.255.252!interface
serial 0/1ip address 192.208.10.2 255.255.255.252!router bgp 300network
192.208.10.0neighbor 192.208.10.1 remote-as 500neighbor 192.208.10.6 remote-as 100
Конфигурация роутера "E":
```

```
!Router Ehostname RouterE!interface loopback 0ip address 200.200.10.1
255.255.255.0interface serial 0ip address 195.211.10.2 255.255.255.252!interface serial
1ip address 128.213.63.6 255.255.255.252!router bgp 400network 200.200.10.0aggregate-
address 200.200.0.0 255.255.0.0 summary-onlyneighbor 128.213.63.5 remote-as 200neighbor
195.211.10.1 remote-as 500
```

Как видно, роутер "E" анонсирует сеть 200.200.0.0/16.

Теперь - окончательная конфигурация роутеров "F" и "G":

```
!Router Fhostname RouterF!interface ethernet 0ip address 203.250.14.2
255.255.255.0!interface serial 1ip address 203.250.15.1 255.255.255.252!router ospf
10network 203.250.0.0 0.0.255.255 area 0 !Router Ghostname RouterG!interface loopback
0ip address 195.211.10.174 255.255.255.192!interface serial 0ip address 192.208.10.1
255.255.255.252!interface serial 1ip address 195.211.10.1 255.255.255.252!router bgp
500network 195.211.10.0aggregate-address 195.211.0.0 255.255.0.0 summary-onlyneighbor
192.208.10.2 remote-as 300neighbor 192.208.10.2 send-communityneighbor 192.208.10.2
route-map setcommunity outneighbor 195.211.10.2 remote-as 400!access-list 2 permit
anyaccess-list 101 permit ip 195.211.0.0 0.0.255.255 255.255.255.0 0.0.0.255!route-map
setcommunity permit 10match ip address 101set community no-export!route-map
setcommunity permit 20match ip address 2
```

В конфигурации роутера "G" демонстрируется использование community filtering'a с добавлением "no-export" community в more specific Class C routes 195.211.0.0/16, которые посылаются роутеру "D". Роутер "D" не передает маршруты роутеру "B".

В результате, таблица BGP маршрутизации на роутере "A":

```
RouterA# show ip bgp table version is 21, local router ID is 203.250.13.41Status codes:
s suppressed, d damped, h history, * valid, > best, i - internalOrigin 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
```

Теперь - таблица маршрутизации IP на роутере "A":

```
RouterA# 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 defaultGateway of last resort is
128.213.63.2 to network 200.200.0.0 192.208.10.0 is variably subnetted, 2 subnets,
2 masksO E2 192.208.10.0 255.255.255.0 [110/1000] via 203.250.14.2,
00:41:25, Ethernet0O 192.208.10.4 255.255.255.252 [110/138] via
203.250.14.2, 00:41:25, Ethernet0C 203.250.13.0 is directly connected, Loopback0
203.250.15.0 is variably subnetted, 3 subnets, 3 masksO 203.250.15.10
255.255.255.255 [110/75] via 203.250.14.2, 00:41:25, Ethernet0O
203.250.15.0 255.255.255.252 [110/74] via 203.250.14.2, 00:41:25,
Ethernet0B 203.250.15.0 255.255.255.0 [200/0] via 203.250.15.2, 00:41:25C
203.250.14.0 is directly connected, Ethernet0 128.213.0.0 is variably subnetted, 2
subnets, 2 masksB 128.213.0.0 255.255.0.0 [20/0] via 128.213.63.2,
00:41:26C 128.213.63.0 255.255.255.252 is directly connected, Serial0B*
200.200.0.0 255.255.0.0 [20/0] via 128.213.63.2, 00:02:38
```

Таблица маршрутизации на роутере "F":

```
RouterF# 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 defaultGateway of last resort is
203.250.15.2 to network 0.0.0.0 192.208.10.0 is variably subnetted, 2 subnets, 2
```

```

masksO E2 192.208.10.0 255.255.255.0 [110/1000] via 203.250.15.2,
00:48:50, Serial10 192.208.10.4 255.255.255.252 [110/128] via
203.250.15.2, 01:12:09, Serial11 203.250.13.0 is variably subnetted, 2 subnets, 2
masksO 203.250.13.41 255.255.255.255 [110/11] via 203.250.14.1,
01:12:09, Ethernet0O 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
masksO 203.250.15.10 255.255.255.255 [110/65] via 203.250.15.2,
01:12:09, Serial1C 203.250.14.0 is directly connected, Ethernet0 128.213.0.0
255.255.0.0 is variably subnetted, 2 subnets, 2 masksO E2 128.213.0.0
255.255.0.0 [110/2000] via 203.250.14.1, 00:45:01, Ethernet0O E2 200.200.0.0
255.255.0.0 [110/1000] via 203.250.14.1, 00:03:47, Ethernet0O*E2 0.0.0.0 0.0.0.0 [110/
1000] via 203.250.15.2, 00:03:33, Serial1

```

Заметим, что на роутере "F" указывается, что сети локальные к AS300, такие как 192.208.10.0, достигаются через роутер "B". Другие известные сети, такие как 200.200.0.0, достижимы через роутер "A". Шлюз для последнего обращения - роутер "B". Если что-либо случится с соединением между роутерами "B" и "D", to default advertised by Router "A" will kick in with MED attribute of 2000.

Таблица BGP маршрутизации на роутере "B":

```

RouterB# show ip bgp table version is 14, local router ID is 203.250.15.10Status codes:
s suppressed, d damped, h history, * valid, > best, i - internalOrigin 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

```

IP routing table на роутере "B":

```

RouterB# 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 defaultGateway of last resort is
203.250.15.2 to network 192.208.10.0* 192.208.10.0 is variably subnetted, 2 subnets,
2 masksB* 192.208.10.0 255.255.255.0 [20/0] via 192.208.10.5, 00:50:46C
192.208.10.4 255.255.255.252 is directly connected, Serial1 203.250.13.0 is
variably subnetted, 2 subnets, 2 masksO 203.250.13.41 255.255.255.255
[110/75] via 203.250.15.1, 01:20:33, Serial0O 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, 2 masksO 203.250.15.10
255.255.255.255 [110/65] via 203.250.15.2, 01:12:09, Serial1C 203.250.14.0
is directly connected, Ethernet0 128.213.0.0 255.255.0.0 is variably subnetted, 2
subnetsC 203.250.15.8 id directly connected, Loopback1C 203.250.15.0 is
directly connected, Serial0O 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 masksO E2 128.213.0.0
255.255.0.0 [110/2000] via 203.250.15.1, 00:46:55,Serial0

```

## V. Summary

Основная функция BGP - обмен информацией о путях к сетям с другими BGP системами. Эта информация использует graph of AS connectivity, к котором устранены зацикливания и с которым используется AS-level policy decision. BGP предоставляет механизм управления потоком передаваемых маршрутов, это route, path и community Filtering. Он также содержит методы объединения маршрутов, такие как CIDR aggregation, Confederations, и Route Reflectors. BGP - это высокоэффективное средство обеспечения маршрутизации без зацикливаний между Автономными системами и в пределах одной Автономной Систсемы.

## **(c) DERIVED FROM CISCO UNIVER CD. For education only**

Автор перевода - Олег Семенюк, olegs@relcom.net.

Редактор - Алексей Руднев, alex@relcom.net.

Попытка перевести все полностью на русский, к сожалению, показывает, что начинает теряться смысл. Поэтому в тексте и частично - в оглавлении оставлены на английском те термины, которые прямо соответствуют командам или понятиям протокола.

---

[\(source\)](#)