

# EIGRP (Enhanced Interior Gateway Routing Protocol)

## Описание

Если в сеть небольшая, то настроить маршруты можно и вручную на каждом роутере.

Но когда сеть растёт, настроить маршруты на каждом маршрутизаторе, а потом ещё и постоянно их актуализировать становится сложно.

Для решения этой проблемы существует динамическая маршрутизация.

### **Погружение в динамическую маршрутизацию.**

Начинаем с протокола EIGRP (Enhanced Interior Gateway Routing Protocol): обзор, базовая конфигурация и проверка

# Оглавление

1. Понимание EIGRP: обзор, базовая конфигурация и проверка
2. Пример настройки EIGRP
3. Про соседство и метрики EIGRP
4. Установка K-значений в EIGRP
5. Конвергенция EIGRP – настройка таймеров
6. Пассивные интерфейсы в EIGRP
7. Настройка статического соседства в EIGRP
8. EIGRP: идентификатор роутера и требования к соседству
9. Траблшутинг EIGRP
10. Поиск и устранение неисправностей протокола EIGRP

# 1. Понимание EIGRP: обзор, базовая конфигурация и проверка

Есть большое количество крупных компаний с сетью, содержащих более 500 маршрутизаторов Cisco (и тысячи коммутаторов Cisco Catalyst). Какой используется протокол маршрутизации, поддерживающий все эти маршрутизаторы в согласии о доступных маршрутах? Это усовершенствованный протокол маршрутизации внутреннего шлюза (EIGRP). Именно этому посвящена данная статья, которая является первой из серии статей, посвященных **EIGRP (Enhanced Interior Gateway Routing Protocol)**.

## ОСНОВЫ EIGRP

Существует давняя дискуссия о фундаментальной природе EIGRP. По своей сути, является ли EIGRP протоколом маршрутизации состояния канала или протоколом маршрутизации вектора расстояния? Или же это гибридный протокол маршрутизации (то есть комбинация того и другого)? Вы найдете много литературы, поддерживающей идею о том, что EIGRP является гибридным протоколом маршрутизации, утверждая, что соседи EIGRP изначально обмениваются своей полной таблицей маршрутизации, во многом похожей на протокол маршрутизации вектора расстояния, и EIGRP отправляет только обновления маршрутизации на основе сетевых изменений, во многом напоминающие протокол маршрутизации состояния канала.

Многие сетевые инженеры пришли к убеждению, что EIGRP-это "продвинутый протокол маршрутизации вектора расстояния". Их рассуждения по этому поводу: рассмотрим фундаментальную характеристику протокола маршрутизации состояния канала, которая заключается в том, что маршрутизаторы поддерживают таблицу топологии, указывающую, как маршрутизаторы связаны между собой. Эти маршрутизаторы (говоря о протоколах маршрутизации, таких как OSPF и IS-IS) затем запускают алгоритм Дейкстры на этой топологии, чтобы определить "кратчайший" путь к целевой сети с точки зрения конкретного маршрутизатора. EIGRP не поддерживает представление о топологии сети и не выполняет алгоритм Дейкстры. Скорее всего, таблица топологии EIGRP содержит список доступных сетей, а также информацию о "расстоянии" до этих сетей.

## ХАРАКТЕРИСТИКИ EIGRP

Давайте начнем наш обзор EIGRP, рассмотрением нескольких основных характеристиках EIGRP:

- Быстрая конвергенция: если пропадает связь в сети, во многих случаях EIGRP может быстро перенаправить поток данных, обойдя место сбоя связи. Обычно это происходит не более чем за 3 секунды. Эта быстрая конвергенция становится возможной благодаря тому, что EIGRP имеет резервный маршрут к сети, и этот резервный маршрут готов взять на себя управление в случае сбоя основного маршрута.
- Высокая масштабируемость: в то время как протокол маршрутизации, такой как RIP, имеет ограничение в пятнадцать переходов маршрутизатора, EIGRP может масштабироваться для поддержки очень крупных корпоративных сетей.
- Балансировка нагрузки с использованием каналов с разной метрикой: по умолчанию и EIGRP, и OSPF балансируют трафик нагрузки по нескольким каналам, ведущим к определенной целевой сети, если стоимость (то есть значение метрики протокола маршрутизации) одинакова. Однако EIGRP может быть настроен для балансировки нагрузки между каналами с неравными стоимостями. Это стало возможным благодаря функции дисперсии.
- Поддержка маски подсети переменной длины (VLSM): в отличие от RIP версии 1, EIGRP отправляет информацию о маске подсети как часть [объявления](#) маршрута.
- Коммуникации через мультикаст: в EIGRP спикер маршрутизатор взаимодействует с другими EIGRP-спикер маршрутизаторами через мультикаст. В частности, EIGRP для IPv4 использует адрес многоадресной рассылки 224.0.0.10, в то время как EIGRP для IPv6 использует адрес многоадресной рассылки ff02::a.
- Больше не проприетарный протокол: в то время как Cisco первоначально представила EIGRP как Cisco-proprietary протокол маршрутизации, в последние годы EIGRP был открыт для других поставщиков. В частности, EIGRP стал открытым стандартом в 2013 году, а информационный RFC EIGRP (RFC 7868) был опубликован в 2016 году.
- Поддержка нескольких протоколов: EIGRP изначально был разработан для поддержки маршрутизации нескольких протоколов, включая IPv4, IPX и AppleTalk. Хотя современные сети редко используют IPX или AppleTalk, EIGRP теперь может поддерживать IPv6, который набирает популярность. Данная поддержка нескольких протоколов становится возможной благодаря Protocol-Dependent Modules (PDM), где существует отдельный PDM, обрабатывающий решения о маршрутизации для каждого маршрутизируемого протокола (например, IPv4 и IPv6).

- Алгоритм диффузионного обновления (DUAL): алгоритм EIGRP, используемый для отслеживания маршрутов, известных соседним маршрутизаторам. DUAL также используется для определения наилучшего пути к целевой сети (то есть к маршруту-преемнику) и любых приемлемых резервных путей к этой целевой сети (то есть к возможным маршрутам-преемникам).
- Суммирование: чтобы уменьшить количество записей в таблице топологии EIGRP (или таблице IP-маршрутизации маршрутизатора), EIGRP имеет возможность суммировать несколько сетевых объявлений в одно сетевое объявление. Это обобщение можно настроить вручную. Однако EIGRP имеет функцию автоматического суммирования маршрутов, которая суммирует сети на классовых границах сети.
- Обновления: полные обновления таблицы топологии EIGRP отправляются при обнаружении новых соседей. В противном случае будут отправлены частичные обновления.

## ОБЗОР НАСТРОЙКИ

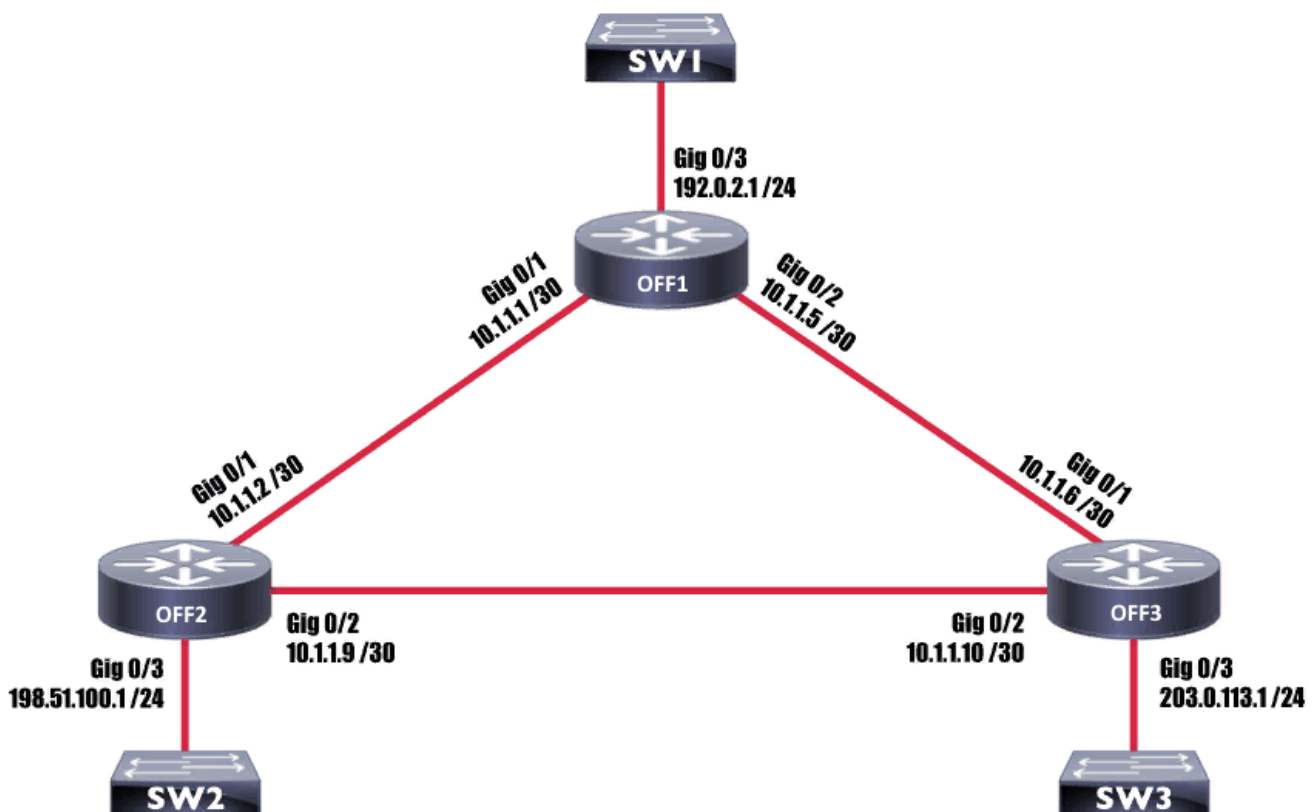
Базовая конфигурация EIGRP очень проста в настройке. На самом деле, для этого требуется только две команды:

```
router eigrp asn
network net-id wildcard-mask
```

Команда **router eigrp asn** запускает процесс маршрутизации EIGRP на маршрутизаторе для автономной системы (AS), заданной переменной **asn**. Эта команда также переводит вас в режим настройки маршрутизатора. Оттуда вы можете выполнить вторую команду, **network net-id wildcard-mask**. Эта вторая команда использует комбинацию сетевого адреса и маски подсети для указания диапазона одного или нескольких IP-адресов, и любой интерфейс маршрутизатора, чей IP-адрес принадлежит этому диапазону IP-адресов, затем участвует в процессе маршрутизации EIGRP. Тем не менее, существуют некоторые правила и модели поведения, которые следует учитывать при выполнении этих команд:

- EIGRP-спикер маршрутизаторы должны быть такими же, как и для формирования соседства.
- После того как маршрутизатор включает EIGRP на интерфейсах, соответствующих команде **network** EIGRP, он пытается обнаружить соседей с помощью многоадресной рассылки приветственных сообщений EIGRP.
- Если в команде **network** не указана маска подсети, то указанный сетевой адрес должен быть классовым сетевым адресом.
- Если в команде **network** не указана маска подсети, а указан классовый сетевой адрес, то все интерфейсы, IP-адреса которых подпадают под классovou сеть (например, 172.16.1.1 /24 подпадает под 172.16.0.0 /16), будут участвовать в процессе маршрутизации EIGRP.

Чтобы проиллюстрировать эти понятия, рассмотрим следующий пример:



```
! Router OFF1
router eigrp 1
network 10.1.1.0 0.0.0.3
network 10.1.1.5 0.0.0.0
network 192.0.2.0
! Router OFF2
router eigrp 1
network 10.0.0.0
network 198.51.100.0
! Router OFF3
router eigrp 1
network 0.0.0.0
```

Конфигурация EIGRP на маршрутизаторах OFF1, OFF2 и OFF3 начинается с команды **router eigrp 1**. Эта команда говорит каждому маршрутизатору начать процесс маршрутизации EIGRP в автономной системе 1. Поскольку номера автономной системы должны совпадать между EIGRP-спикер-соседями, все три маршрутизатора используют один и тот же номер автономной системы 1. Кроме того, обратите внимание, как меняется конфигурация при использовании команды network:

### КОМАНДА NETWORK 10.1.1.0 0.0.0.3 НА РОУТЕРЕ OFF1

На маршрутизаторе OFF1 команда **network 10.1.1.0 0.0.0.3** задает сетевой адрес 10.1.1.0 с обратной маской 0.0.0.3, которая соответствует 30-битной маске подсети (то есть маске подсети 255.255.255.252). Поскольку IP-адрес интерфейса Gig 0/1 маршрутизатора OFF1 10.1.1.1 / 30 попадает в эту подсеть, этот интерфейс проинструктирован участвовать в процессе EIGRP.

### КОМАНДА NETWORK 10.1.1.5 0.0.0.0 НА РОУТЕРЕ OFF1

Команда **network 10.1.1.5 0.0.0.0** указывает конкретный IP-адрес, а не всю подсеть (или можно утверждать, что это подсеть, содержащая один IP-адрес). Мы знаем, что он указывает только один IP-адрес из-за маски подсети 0.0.0.0. Напомним, что в маске подсети мы имеем ряд непрерывных нулей, за которыми следует ряд непрерывных единиц (в двоичном коде). Двоичные нули соответствуют позиции битов в IP-адресе, определяющие адрес сети, а бинарные единицы соответствуют позиции битов в IP-адресе, который указывает адрес узла. Однако в том случае, когда у нас все нули, как в нашем случае, у нас есть сеть с одним и только одним IP-адресом (то есть маска подсети равна /32). Поскольку IP-адрес совпадает с IP-адресом интерфейса Gig 0/2 маршрутизатора OFF1, этот интерфейс также участвует в процессе маршрутизации EIGRP.

### КОМАНДА NETWORK 192.0.2.0 НА РОУТЕРЕ OFF1

Последняя команда **network** на маршрутизаторе OFF1 - это network 192.0.2.0. Интересно, что эта команда фактически была введена как сеть 192.0.2.0 0.0.0.255, но поскольку 0.0.0.255 является обратной маской, соответствующей маске подсети по умолчанию сети класса C (в данном случае 192.0.2.0 /24), она подразумевается, но не показывается. IP-адрес интерфейса Gig 0/3 маршрутизатора OFF1 192.0.2.1 / 24 действительно попадает в подсеть класса C, заданную командой **network**. Таким образом, Gig 0/3 также начинает участвовать в процессе маршрутизации EIGRP маршрутизатора OFF1.

### КОМАНДА NETWORK 10.0.0.0 НА РОУТЕРЕ OFF2

Команда network 10.0.0.0 на маршрутизаторе OFF2, не имеет обратной маски. Однако помните, что из ранее обсуждавшейся команды network (на маршрутизаторе OFF1) обратная маска подсети не отображается, если она отражает естественную маску заданной подсети. Основываясь на этой логике, мы можем заключить, что если мы намеренно опустим аргумент обратной маски из команды network, то предполагаемая обратная маска будет маской подсети, соответствующей классовой маске подсети сети, указанной в команде network. В этом случае первый октет сети, указанный в команде network address, равен 10. 10 в первом октете адреса указывает, что мы имеем дело с адресом класса A, который имеет маску подсети по умолчанию 255.0.0.0 и, следовательно, обратную маску по умолчанию 0.0.0.255. Поскольку интерфейсы Gig 0/1 и Gig 0/2 маршрутизатора OFF2 подпадают под этот классовой сетевой оператор, оба интерфейса участвуют в процессе маршрутизации EIGRP маршрутизатора OFF2.

### КОМАНДА NETWORK 198.51.100.0 НА РОУТЕРЕ OFF2

Как и предыдущая команда network, команда маршрутизатора OFF2 network 198.51.100.0 была введена без указания обратной маски. Поскольку первый октет адреса равен 198, мы можем заключить, что у нас есть сеть класса C, чья маска подсети по умолчанию равна 255.255.255.0, а обратная маска по умолчанию равна 0.0.0.255. IP-адрес (198.51.100.1 /24) интерфейсного Gig

0/3 на маршрутизаторе OFF2 живет в пределах указанной подсети 198.51.100.0 /24. Таким образом, интерфейс участвует в процессе маршрутизации EIGRP.

### КОМАНДА NETWORK 0.0.0.0 НА РОУТЕРЕ OFF3

Напомним, что оператор network EIGRP, вопреки распространенному мнению, не указывает сеть для [объявления](#). Скорее, он определяет диапазон одного или нескольких IP-адресов, и любой интерфейс с IP-адресом в этом диапазоне проинструктирован участвовать в процессе маршрутизации EIGRP. Это означает, что, если мы хотим, чтобы все интерфейсы на маршрутизаторе участвовали в одном и том же процессе маршрутизации EIGRP, мы могли бы дать команду network 0.0.0.0, чтобы указать все возможные IP-адреса. Поскольку IP-адрес каждого отдельного интерфейса подпадает под категорию "все возможные IP-адреса", все интерфейсы на маршрутизаторе OFF3 проинструктированы участвовать в процессе маршрутизации EIGRP. Кроме того, сетевые адреса этих участвующих интерфейсов (вместе с информацией о подсети для этих сетевых адресов) затем объявляются через EIGRP.

### ПРОВЕРКА

Процесс проверки EIGRP - это нечто большее, чем просто проверка того, что между всеми маршрутизаторами сформировались соседские отношения и что все маршрутизаторы изучили все маршруты в сети. Процесс верификации должен помочь нам убедиться в том, что наши изначальные требования были выполнены. Например, нам нужно найти соответствующие маршруты, определенные интерфейсы и конкретных соседей, которые будут отображаться в таблицах EIGRP. Как только определимся с нашими изначальными целями проектирования и ожидаемыми результатами, мы можем применить команды проверки EIGRP, показанные в таблице ниже:

### КЛЮЧЕВЫЕ КОМАНДЫ ПРОВЕРКИ EIGRP

Команда	Информация
<b>show ip route</b>	Перечисляет содержимое таблицы IP-маршрутизации маршрутизатора, причем EIGRP-изученные маршруты появляются с кодом D в левой части выходных данных.
<b>show ip protocols</b>	Перечисляет содержимое команд конфигурации сети для каждого процесса протокола маршрутизации и список соседних IP-адресов.
<b>show ip eigrp interfaces</b>	Перечисляет рабочие интерфейсы, на которых включен EIGRP (на основе команды network), опуская пассивные интерфейсы.
<b>show ip eigrp neighbors</b>	Перечисляет известных соседей EIGRP, исключая EIGRP-спикер маршрутизаторы, препятствующим установлению соседства EIGRP.
<b>show ip eigrp topology [all-links]</b>	Перечисляет все маршруты-посредники и возможные маршруты-посредники, известные маршрутизатору. Добавление ключевого слова all-links приводит к тому, что выходные данные включают любые дополнительные известные маршруты, которые не являются ни посредниками, ни возможными посредниками маршрутов.

В следующих примерах показаны результаты выполнения каждой из этих команд после их выполнения на маршрутизаторе OFF1, показанном в предыдущей топологии.

Вывод результатов команды show ip route на маршрутизаторе OFF1:

OFF1#show ip route

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP  
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area  
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2  
E1 - OSPF external type 1, E2 - OSPF external type 2  
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2  
ia - IS-IS inter area, \* - candidate default, U - per-user static route  
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP  
a - application route  
+ - replicated route, % - next hop override, p - overrides from Pfr

Gateway of last resort is not set

```
10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C      10.1.1.0/30 is directly connected, GigabitEthernet0/1
L      10.1.1.1/32 is directly connected, GigabitEthernet0/1
C      10.1.1.4/30 is directly connected, GigabitEthernet0/2
L      10.1.1.5/32 is directly connected, GigabitEthernet0/2
D      10.1.1.8/30 [90/3072] via 10.1.1.6, 01:23:48, GigabitEthernet0/2
        [90/3072] via 10.1.1.2, 01:23:48, GigabitEthernet0/1
192.0.2.0/24 is variably subnetted, 2 subnets, 2 masks
C      192.0.2.0/24 is directly connected, GigabitEthernet0/3
L      192.0.2.1/32 is directly connected, GigabitEthernet0/3
D      198.51.100.0/24 [90/3072] via 10.1.1.2, 01:23:48, GigabitEthernet0/1
D      203.0.113.0/24 [90/3072] via 10.1.1.6, 01:23:48, GigabitEthernet0/2
```

Обратите внимание, как маршруты, изученные с помощью EIGRP, показаны с литерой D в левом столбце. Этот код D указывает на маршрут, изученный через EIGRP. Эти маршруты включают 10.1.1.8/30, 198.51.100.0/24 и 203.0.113.0 /24. Также обратите внимание на выделенные числовые значения 90 в каждом EIGRP-изученном маршруте. 90 - это административное расстояние EIGRP (то есть его правдоподобность по сравнению с другими источниками маршрутизации), где более низкие значения административного расстояния предпочтительны по сравнению с более высокими значениями.

**Вывод из команды show ip protocols на маршрутизаторе OFF1**



```
OFF1#show ip protocols
```

```
*** IP Routing is NSF aware ***
```

```
Routing Protocol is "application"
```

```
Sending updates every 0 seconds
```

```
Invalid after 0 seconds, hold down 0, flushed after 0
```

```
Outgoing update filter list for all interfaces is not set
```

```
Incoming update filter list for all interfaces is not set
```

```
Maximum path: 32
```

```
Routing for Networks:
```

```
Routing Information Sources:
```

```
Gateway Distance Last Update
```

```
Distance: (default is 4)
```

```
Routing Protocol is "eigrp 1"
```

```
Outgoing update filter list for all interfaces is not set
```

```
Incoming update filter list for all interfaces is not set
```

```
Default networks flagged in outgoing updates
```

```
Default networks accepted from incoming updates
```

```
EIGRP-IPv4 Protocol for AS(1)
```

```
Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
```

```
Soft SIA disabled
```

```
NSF-aware route hold timer is 240
```

```
Router-ID: 192.0.2.1
```

```
Topology : 0 (base)
```

```
Active Timer: 3 min
```

```
Distance: internal 90 external 170
```

```
Maximum path: 4
```

```
Maximum hopcount 100
```

```
Maximum metric variance 1
```

```
Automatic Summarization: disabled
```

```
Maximum path: 4
```

```
Routing for Networks:
```

```
10.1.1.0/30
```

```
10.1.1.5/32
```

```
192.0.2.0
```

```
Routing Information Sources:
```

```
Gateway Distance Last Update
```

```
10.1.1.2 90 01:23:54
```

```
10.1.1.6 90 01:23:54
```

```
Distance: internal 90 external 170
```

Вывод информации команды show ip protocols на EIGRP-спикер маршрутизаторе, как видно выше, предлагает нам несколько точек данных. Например, в разделе Routing for Networks: вы видите список сетей, указанных командой network в режиме конфигурации EIGRP. В разделе Routing Information Sources: вы можете видеть IP-адреса соседей EIGRP, которые являются 10.1.1.2 (то есть маршрутизатором OFF2) и 10.1.1.6 (то есть маршрутизатором OFF3) нашей топологии. Также в этом разделе вы можете увидеть административное расстояние (AD) до наших соседей. Поскольку эти соседи являются EIGRP-спикер маршрутизаторами, у них есть EIGRP AD по умолчанию 90. Наконец, обратите внимание на метрический вес K1=1, K2=0, K3=1, K4=0, K5=0 части выходного сигнала. В следующей статье мы узнаем, как EIGRP вычисляет свою метрику и как этот расчет включает в себя K-значения.

**Вывод из команды show ip eigrp interfaces на маршрутизаторе OFF1**



OFF1#show ip eigrp interfaces

EIGRP-IPv4 Interfaces for AS(1)

Interface	Peers	Xmit Queue Un/Reliable	PeerQ Un/Reliable	Mean SRTT	Pacing Time Un/Reliable	Multicast Flow Timer	Pending Routes
Gi0/1	1	0/0	0/0	323	0/0	1616	0
Gi0/2	1	0/0	0/0	5	0/0	50	0
Gi0/3	0	0/0	0/0	0	0/0	0	0

Выходные данные **show ip eigrp interfaces**, рассмотренные выше, указывают на то, что Gig 0/1, Gig 0/2 и Gig 0/3 маршрутизатора OFF1 участвуют в процессе маршрутизации EIGRP. В частности, этот процесс предназначен для EIGRP AS 1. Также обратите внимание, что соседство EIGRP было установлено с другим маршрутизатором, подключенным от интерфейса Gig 0/1 маршрутизатора OFF1, и другим от интерфейса Gig 0/2. Доказательством этих соседских отношений является наличие числа, превышающего 0 в колонке Peers. Поскольку интерфейс Gig 0/3 маршрутизатора OFF1 не формировал соседство с любыми другими маршрутизаторами, говорящими на EIGRP, в его столбце Peers стоит 0.

Вывод из команды show ip eigrp neighbors на маршрутизаторе OFF1:

OFF1#show ip eigrp neighbors

EIGRP-IPv4 Neighbors for AS(1)

H	Address	Interface	Hold (sec)	Uptime	SRTT (ms)	RTO	Q Cnt	Seq Num
0	10.1.1.6	Gi0/2	12	01:24:06	5	100	0	18
1	10.1.1.2	Gi0/1	11	01:25:17	323	1938	0	17

В то время как выводимые данные из команды **show ip eigrp interfaces** указывали, что у нас было несколько соседей EIGRP, выходные данные из команды **show ip eigrp neighbors**, как видно выше, предлагают более подробную информацию об этих соседях. В частности, сосед, связанный с интерфейсом маршрутизатора OFF1 по Gig 0/1, имеет IP-адрес 10.1.1.2, а сосед соединен с интерфейсом маршрутизатора OFF1 по Gig0/2 имеет IP-адрес 10.1.1.6.

Вывод из команды show ip eigrp topology [all-links] на маршрутизаторе OFF1:

```
OFF1#show ip eigrp topology
```

```
EIGRP-IPv4 Topology Table for AS(1)/ID(192.0.2.1)
```

```
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,  
r - reply Status, s - sia Status
```

```
P 192.0.2.0/24, 1 successors, FD is 2816  
  via Connected, GigabitEthernet0/3  
P 10.1.1.8/30, 2 successors, FD is 3072  
  via 10.1.1.2 (3072/2816), GigabitEthernet0/1  
  via 10.1.1.6 (3072/2816), GigabitEthernet0/2  
P 203.0.113.0/24, 1 successors, FD is 3072  
  via 10.1.1.6 (3072/2816), GigabitEthernet0/2  
P 10.1.1.4/30, 1 successors, FD is 2816  
  via Connected, GigabitEthernet0/2  
P 198.51.100.0/24, 1 successors, FD is 3072  
  via 10.1.1.2 (3072/2816), GigabitEthernet0/1  
P 10.1.1.0/30, 1 successors, FD is 2816  
  via Connected, GigabitEthernet0/1
```

```
OFF1#show ip eigrp topology all-links
```

```
EIGRP-IPv4 Topology Table for AS(1)/ID(192.0.2.1)
```

```
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,  
r - reply Status, s - sia Status
```

```
P 192.0.2.0/24, 1 successors, FD is 2816, serno 17  
  via Connected, GigabitEthernet0/3  
P 10.1.1.8/30, 2 successors, FD is 3072, serno 14  
  via 10.1.1.2 (3072/2816), GigabitEthernet0/1  
  via 10.1.1.6 (3072/2816), GigabitEthernet0/2  
P 203.0.113.0/24, 1 successors, FD is 3072, serno 15  
  via 10.1.1.6 (3072/2816), GigabitEthernet0/2  
  via 10.1.1.2 (3328/3072), GigabitEthernet0/1  
P 10.1.1.4/30, 1 successors, FD is 2816, serno 13  
  via Connected, GigabitEthernet0/2  
P 198.51.100.0/24, 1 successors, FD is 3072, serno 8  
  via 10.1.1.2 (3072/2816), GigabitEthernet0/1  
  via 10.1.1.6 (3328/3072), GigabitEthernet0/2  
P 10.1.1.0/30, 1 successors, FD is 2816, serno 1  
  via Connected, GigabitEthernet0/1
```

Одной из наиболее распространенных команд, используемых для проверки EIGRP и устранения неполадок, является `show ip eigrp topology`, как показано в приведенном выше примере. Выходные данные этой команды показывают маршруты-преемники (то есть предпочтительные маршруты) и возможные маршруты-преемники (то есть резервные маршруты), известные процессу маршрутизации EIGRP. Пожалуйста, имейте в виду, что появление маршрута в таблице топологии EIGRP не гарантирует его присутствия в таблице IP-маршрутизации маршрутизатора. В частности, маршруты-преемники, присутствующие в таблице топологии EIGRP, являются только кандидатами для попадания в таблицу IP-маршрутизации маршрутизатора. Например, маршрутизатор может обладать более достоверной информацией о маршрутизации для сети, такой как статически настроенный маршрут с административным расстоянием 1. Если EIGRP действительно является наиболее правдоподобным источником маршрутизации для конкретной сети, то эта сеть будет введена в таблицу IP-маршрутизации маршрутизатора. Кроме того, обратите внимание, как добавление аргумента `all-links` в приведенном выше примере показывает еще больше маршрутов (они выделены). Разница заключается в том, что аргумент `all-links` предписывает команде `show ip eigrp topology` отображать все изученные EIGRP маршруты, даже если некоторые из маршрутов не считаются маршрутами-преемниками или возможными маршрутами-преемниками.

## 2. Пример настройки EIGRP

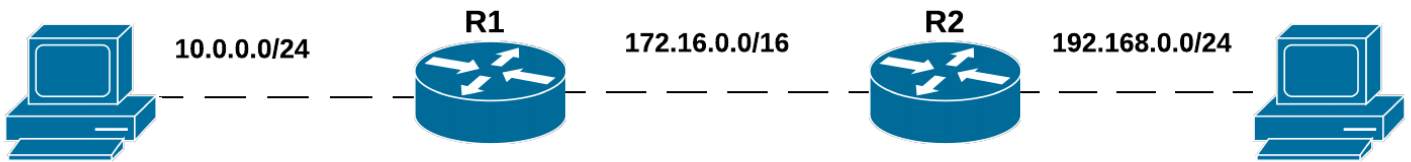
Настройка **EIGRP** состоит из двух шагов:

1. включения протокола глобальной командой `router eigrp ASN_NUMBER`;
2. выбора сетей, которые протокол будет «вещать», для чего используется команда(ы) `network`;

Первая команда включения говорит сама за себя, но поясним про **ASN\_NUMBER** – это номер автономной системы, и для установления сетевой связности между несколькими маршрутизаторами, использующими EIGRP, данный номер должен быть одинаковым. Вторая команда работает также, как и в RIP по умолчанию – то есть включение протокола на интерфейсе и указание классовых сетей.

### Пример настройки EIGRP

В нашей топологии у маршрутизаторов R1 и R2 есть напрямую подключенные подсети.



Нам нужно включить данные подсети в процесс динамической маршрутизации EIGRP. Для этого нам сначала нужно включить EIGRP на обоих маршрутизаторах и затем «вещать» данные сети с помощью команды **network**. На маршрутизаторе R1 переходим в глобальный режим конфигурации и вводим следующие команды:

```
router eigrp 1
network 10.0.0.0
network 172.16.0.0
```

Немного пояснений – сначала мы включаем протокол динамической маршрутизации, затем меняем версию на вторую, затем используем команду `network 10.0.0.0` для включения интерфейса Fa0/1 на маршрутизаторе R1. Как мы уже говорили, команда `network` берет **классовую** сеть, так что каждый интерфейс с подсетью, начинающейся на 10 будет добавлен в EIGRP процесс. Также нам необходимо получить маршрут между двумя роутерами через EIGRP, для этого добавляем еще одну команду `network` – с адресом 172.16.0.0.

IP-адреса начинающиеся на 10, по умолчанию принадлежат к классу «А» и имеют стандартную маску подсети 255.0.0.0.

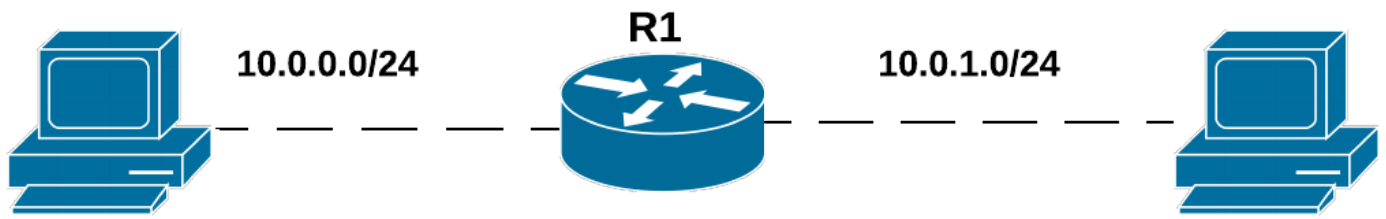
На R2 настройка выглядит похожей, только с другой подсетью – т.к к маршрутизатору R2 напрямую подключена подсеть 192.168.0.0.

```
router eigrp 1
network 192.168.0.0
network 172.16.0.0
```

Вот и все – также просто, как и настроить RIP: главное не забывать указывать одинаковый номер автономной системы. Для проверки работоспособности EIGRP введите команду `show ip eigrp neighbors` на обоих маршрутизаторах и убедитесь, что там указан адрес другого маршрутизатора. Данная команда показывает список всех EIGRP «соседей», с разнообразной информацией – номером локального интерфейса и т.д Также проверить сетевую связность можно с помощью команды вывода таблицы маршрутизации `sh ip route`. Маршруты, получаемые по EIGRP будут отмечены буквой «D».

### Пример настройки EIGRP 2

Как мы уже говорили, по умолчанию команда `network` использует классовую сеть, т.е все интерфейсы внутри это классовой сети будут участвовать в процессе маршрутизации. Для включения EIGRP только на нужном вам интерфейсе, необходимо использование обратной маски. То есть команда будет выглядеть следующим образом: `network ОБРАТНАЯ_МАСКА`



В нашем примере у маршрутизатора R1 есть две напрямую подключенные подсети, 10.0.0.0/24 и 10.0.1.0/24. Наша цель – включить EIGRP только на подсети, подключенной к интерфейсу Fa0/0. Если просто использовать команду `network` – обе подсети будут добавлены в EIGRP процесс, т.к будет использоваться классовая сеть. Для настройки EIGRP только на интерфейсе Fa0/0, нужно использовать команду `network 10.0.0.0 0.0.0.255`. Она включит EIGRP только на интерфейсах 10.0.0.X. Проверить можно с помощью команды `sh ip protocols`, что только сеть 10.0.0.0/24 добавлена в EIGRP процесс.

### 3. Про соседство и метрики EIGRP

Метрические веса TOS K1 K2 K3 K4 K5, выданные командой в режиме конфигурации маршрутизатора EIGRP, может быть использована для установки K-значений, используемых EIGRP в своем расчете. Параметр TOS был предназначен для использования маркировки качества обслуживания (где TOS обозначает тип служебного байта в заголовке IPv4). Однако параметр TOS должен быть равен 0. На самом деле, если вы введете число в диапазоне 1 - 8 и вернетесь назад, чтобы изучить свою текущую конфигурацию, вы обнаружите, что Cisco IOS изменила это значение на 0. Пять оставшихся параметров в команде **metric weights** - это пять K-значений, каждое из которых может быть задано числом в диапазоне от 0 до 255.

Например, представьте, что в нашем проекте мы обеспокоены тем, что нагрузка на наши линии может быть высокой в разы, и мы хотим, чтобы EIGRP учитывал уровень насыщения линии при расчете наилучшего пути. Изучая полную формулу расчета метрики EIGRP, мы замечаем, что наличие ненулевого значения для K2 приведет к тому, что EIGRP будет учитывать нагрузку. Поэтому мы решили установить K2 равным 1, в дополнение к K1 и K3, которые уже установлены в 1 по умолчанию. Значения K4 и K5 сохранятся на уровне 0. В приведенном ниже примере показано, как можно настроить такой набор K-значений.

```
OFF1#conf term
Enter configuration commands, one per line. End with CNTL/Z .
OFF1(config)#router eigrp 1
OFF1(config-router)#metric weights 0 1 1 1 0 0
OFF1(config-router)#end

OFF1#
*Oct 19 17:43:58.625: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 10.1.1.2 (GigabitEthernet0/1) is down: metric changed
*Oct 19 17:43:58.626: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 10.1.1.6 (GigabitEthernet0/2) is down: metric changed
*Oct 19 17:43:59.389: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 10.1.1.2 (GigabitEthernet0/1) is down: K-value mismatch
*Oct 19 17:44:00.869: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 10.1.1.6 (GigabitEthernet0/2) is down: K-value mismatch
```

Первый 0 в команде `metric weights 0 1 1 1 0 0`, показанной в приведенном выше примере, задает значение TOS равное 0. Следующие пять чисел задают наши пять K-значений: K1 = 1, K2 = 1, K3 = 1, K4 = 0, K5 = 0. Этот набор K-значений теперь будет учитывать не только пропускную способность и задержку, но и нагрузку при выполнении расчета метрики. Однако есть проблема. Обратите внимание на сообщения консоли, появляющиеся после нашей конфигурации. Оба наших соседства были разрушены, потому что маршрутизатор OFF1 теперь имеет другие K-значения, чем маршрутизаторы OFF2 и OFF3. Напомним, что соседи EIGRP должны иметь соответствующие K-значения, а это означает, что при изменении K-значений на одном EIGRP-спикер маршрутизаторе, вам нужен идентичный набор K-значений на каждом из его соседей EIGRP. Как только вы настроите соответствующие K-значения на этих соседях, то каждый из этих соседей должен соответствовать K-значениям. Как вы можете видеть, в большой топологии может возникнуть значительная административная нагрузка, связанная с манипуляцией K-значением.

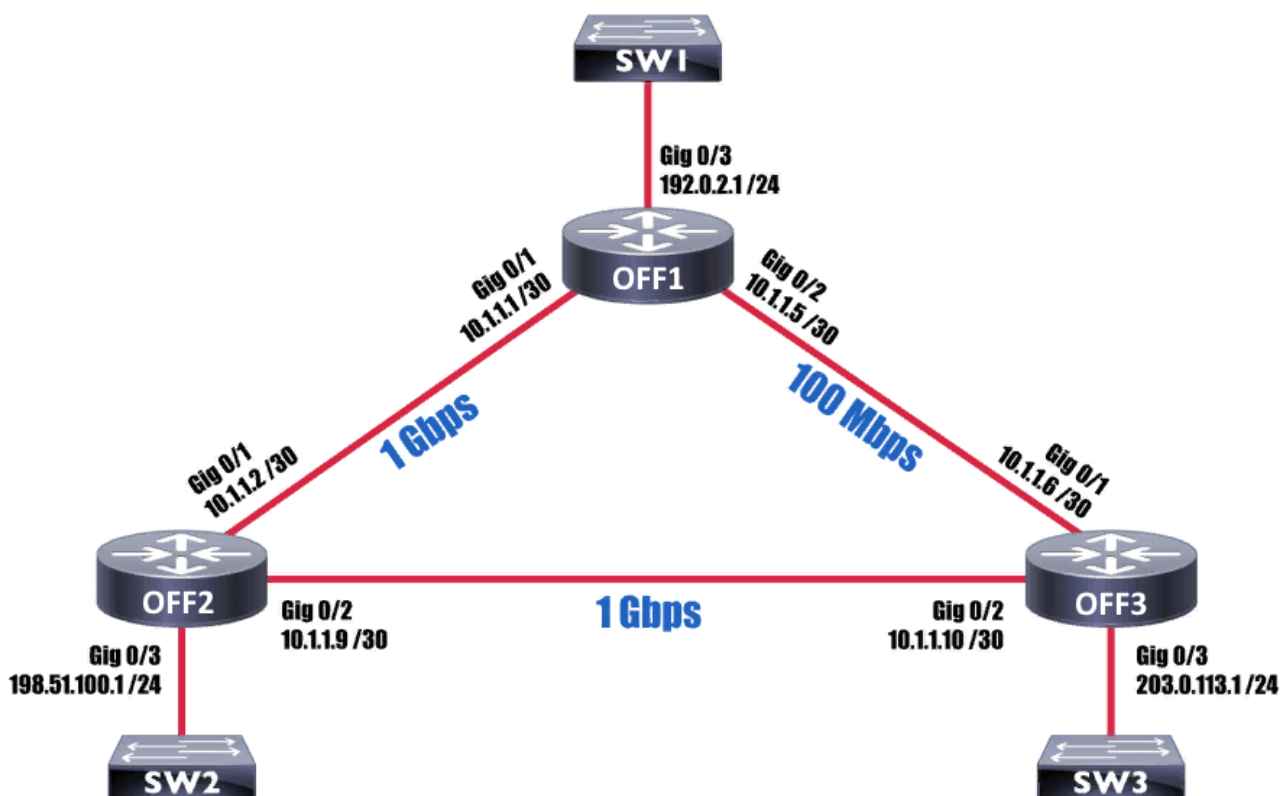
### ПРЕЕМНИК И ВОЗМОЖНЫЕ МАРШРУТЫ ПРЕЕМНИКОВ

Одна из причин, по которой EIGRP быстро восстанавливает соединения в случае сбоя маршрута, заключается в том, что EIGRP часто имеет резервный маршрут, готовый взять на себя управление, если основной маршрут уходит в down. Чтобы убедиться, что резервный маршрут не зависит от основного маршрута, EIGRP тщательно проверяет резервный маршрут, убедившись, что он соответствует условию осуществимости EIGRP. В частности, условие осуществимости гласит:

Маршрут EIGRP является возможным маршрутом-преемником, если его сообщенное расстояние (RD) от нашего соседа меньше возможного расстояния (FD) маршрута-преемника.

Например, рассмотрим топологию, показанную на следующем рисунке, и соответствующую конфигурацию, приведенную ниже. Обратите внимание, что сеть 10.1.1.8/30 (между маршрутизаторами OFF2 и OFF3) доступна из OFF1 через OFF2 или через OFF3. Если маршрутизатор OFF1 использует маршрут через OFF2, он пересекает канал связи 1 Гбит/с, чтобы достичь целевой сети. Однако маршрут через OFF3 заставляет трафик пересекать более медленное соединение со скоростью 100 Мбит/с. Поскольку EIGRP учитывает пропускную способность и задержку по умолчанию, мы видим, что предпочтительный маршрут проходит через маршрутизатор OFF2. Однако, что делать, если связь между маршрутизаторами OFF1 и OFF2

обрывается? Есть ли возможный преемственный маршрут, который может почти сразу заработать? Опять же, мы видим, что маршрутизатор OFF1 будет использовать возможный маршрут преемника через маршрутизатор OFF3. Однако, прежде чем мы убедимся в этом, мы должны подтвердить, что путь через OFF3 соответствует условию осуществимости.



Возможное условие преемника выполнено на маршрутизаторе OFF1

```
OFF1#show ip eigrp topology
```

```
EIGRP-IPv4 Topology Table for AS(1)/ID(192.0.2.1)
```

```
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status
```

```
P 192.0.2.0/24, 1 successors, FD is 2816
   via Connected, GigabitEthernet0/3
P 10.1.1.8/30, 1 successors, FD is 3072
   via 10.1.1.2 (3072/2816), GigabitEthernet0/1
   via 10.1.1.6 (28416/2816), GigabitEthernet0/2
P 203.0.113.0/24, 1 successors, FD is 3328
   via 10.1.1.2 (3328/3072), GigabitEthernet0/1
   via 10.1.1.6 (28416/2816), GigabitEthernet0/2
P 10.1.1.4/30, 1 successors, FD is 28160
   via Connected, GigabitEthernet0/2
   via 10.1.1.2 (3328/3072), GigabitEthernet0/1
P 198.51.100.0/24, 1 successors, FD is 3072
   via 10.1.1.2 (3072/2816), GigabitEthernet0/1
P 10.1.1.0/30, 1 successors, FD is 2816
   via Connected, GigabitEthernet0/1
```

```
OFF1#
```

Просто в силу того, что маршрут через маршрутизатор OFF3 (то есть через 10.1.1.6) появляется в выходных данных команды **show ip eigrp topology**, выполненной на маршрутизаторе OFF1, мы делаем вывод, что путь через OFF3 действительно является возможным маршрутом-преемником. Однако давайте рассмотрим выходные данные немного более внимательно, чтобы определить, почему это возможный маршрут-преемник.



Во-первых, рассмотрим запись из выходных данных в приведенном выше примере, идентифицирующую последующий маршрут (то есть предпочтительный маршрут):

`via 10.1.1.2 (3072/2816), GigabitEthernet0/1`

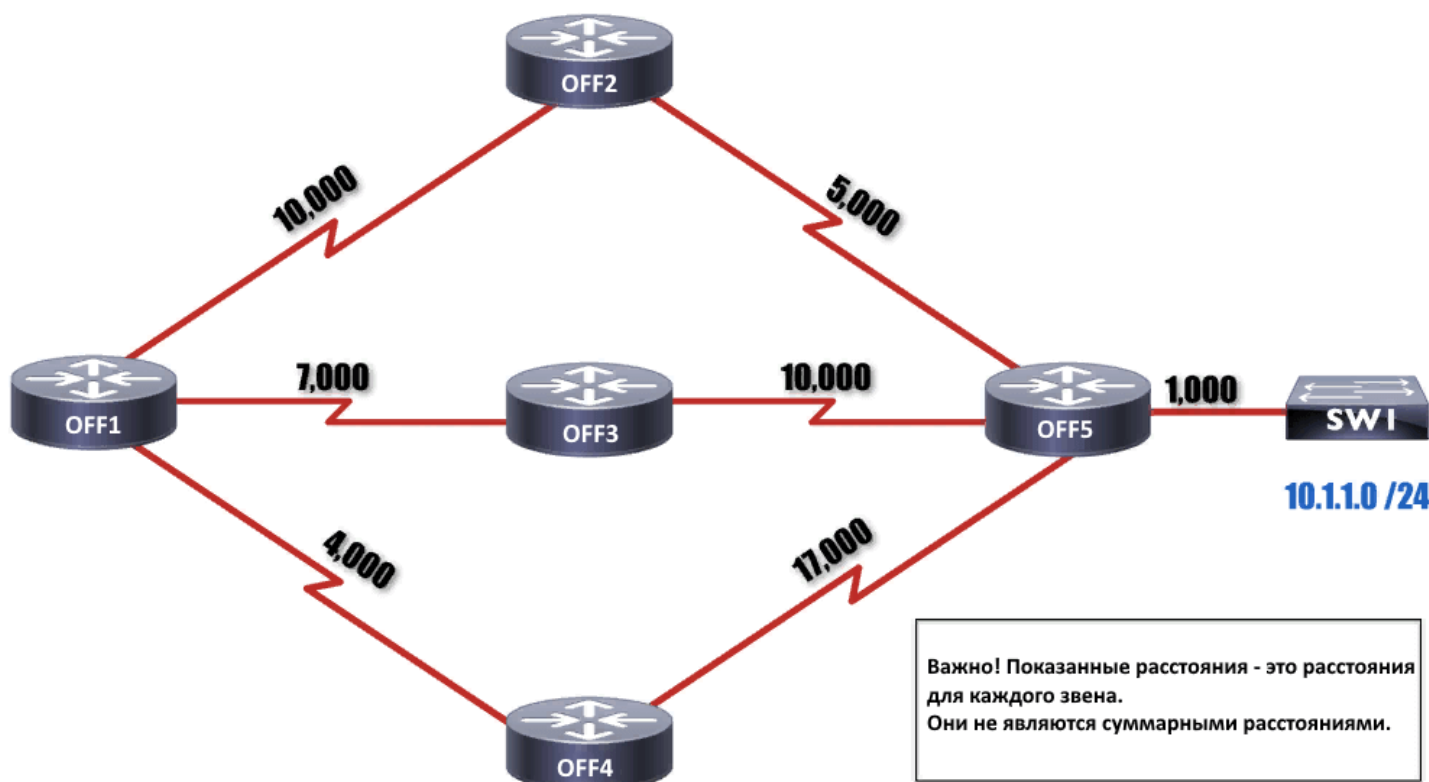
Часть выходных данных **via 10.1.1.2** говорит, что этот маршрут указывает на адрес следующего прыжка 10.1.1.2, который является маршрутизатором OFF2. На интерфейсе GigabitEthernet0/1 часть выходных данных указывает, что мы выходим из маршрутизатора OFF1 через интерфейс Gig0/1 (то есть выходной интерфейс). Теперь давайте рассмотрим эти два числа в скобках: (3072/2816). Стоимость 2816 называется зафиксированная дистанция (reported distance (RD)). В некоторых литературных источниках это значение также называется advertised distance (AD). Эти термины, синонимы, относятся к метрике EIGRP, сообщенной (или объявленной) нашим соседом по EIGRP. В данном случае значение 2816 говорит нам, что метрика маршрутизатора OFF2 (то есть расстояние) до сети 10.1.1.8/30 равна 2816. Значение 3072 на выходе - это допустимое расстояние маршрутизатора OFF1 (FD). FD вычисляется путем добавления RD нашего соседа к метрике, необходимой для достижения нашего соседа. Поэтому, если мы добавим метрику EIGRP между маршрутизаторами OFF1 и OFF2 к RD маршрутизатора OFF2, мы получим FD (то есть общее расстояние), необходимое для того, чтобы OFF1 добрался до 10.1.1.8/30 через маршрутизатор OFF2. Кстати, причина, по которой маршрутизатор OFF1 определяет наилучший путь к сети 10.1.1.8/30, - это **via** **via** router OFF2 (то есть 10.1.1.2) В отличие от маршрутизатора OFF3 (то есть 10.1.1.6), потому что FD пути через OFF1 (3072) меньше, чем FD пути через OFF2 (28,416).

Далее рассмотрим запись для возможного последующего маршрута из приведенного выше примера:

`via 10.1.1.6 (28416/2816), GigabitEthernet0/2`

Часть выходных данных **via 10.1.1.6** говорит, что этот маршрут указывает на адрес следующего прыжка 10.1.1.6, который является маршрутизатором OFF3. На интерфейсе GigabitEthernet0/2 часть результатов показывает, что мы выходим из маршрутизатора OFF1 через интерфейс Gig0/2. Эта запись имеет FD 28 416 и RD 2816. Однако прежде, чем EIGRP просто слепо сочтет этот резервный путь возможным приемником, он проверяет маршрут на соответствие условию осуществимости. В частности, процесс EIGRP на маршрутизаторе OFF1 запрашивает, является ли RD от маршрутизатора OFF3 меньше, чем FD последующего маршрута. В этом случае RD от маршрутизатора OFF3 составляет 2816, что действительно меньше, чем FD приемника 3072. Поэтому маршрут через маршрутизатор OFF3 считается возможным приемником маршрута.

Чтобы утвердить эту важную концепцию, рассмотрим топологию, показанную ниже. Процесс EIGRP на маршрутизаторе OFF1 изучил три пути для достижения сети 10.1.1.0/24. Однако далее EIGRP должен определить, какой из этих путей является маршрутом-приемником, какие (если таковые имеются) пути являются возможными маршрутами-приемниками, а какие (если таковые имеются) пути не являются ни приемником, ни возможным маршрутом-приемником. Результаты расчетов EIGRP приведены в таблице ниже.





Neighbor	Reported Distance (RD)	Feasible Distance (FD)	(Feasible) Successor?
OFF2	6,000	16,000	Yes: Successor
OFF3	11,000	18,000	Yes: Feasible Successor
OFF4	18,000	22,000	No

Используя приведенную выше таблицу в качестве рассмотрения, сначала рассмотрим путь маршрутизатора OFF1 к сети 10.1.1.0/24 через маршрутизатор OFF2. С точки зрения маршрутизатора OFF2, расстояние до сети 10.1.1.0/24 - это расстояние от OFF2 до OFF5 (которое равно 5000) плюс расстояние от OFF5 до сети 10.1.1.0/24 (которое равно 1000). Это дает нам в общей сложности 6000 для расстояния от маршрутизатора OFF2 до сети 10.1.1.0/24. Это расстояние, которое маршрутизатор OFF2 сообщает маршрутизатору OFF1. Таким образом, маршрутизатор OFF1 видит RD 6000 от маршрутизатора OFF2. Маршрутизатор OFF1, затем добавляет расстояние между собой и маршрутизатором OFF2 (который равен 10 000) к RD от OFF2 (который равен 6000), чтобы определить его FD для достижения сети 10.1.1.0/24 составляет 16 000 (то есть  $10\,000 + 6000 = 16\,000$ ). Процесс EIGRP на маршрутизаторе OFF1 выполняет аналогичные вычисления для путей к сети 10.1.1.0/24 через маршрутизаторы OFF3 и OFF4. Ниже приведены расчеты, которые привели к значениям, приведенным в таблице.

**Path via OFF2:** OFF2's RD =  $5,000 + 1,000 = 6,000$ . Distance to OFF2 = 10,000. FD = Distance to OFF2 + OFF2's RD =  $10,000 + 6,000 = 16,000$ .  
**Path via OFF3:** OFF3's RD =  $10,000 + 1,000 = 11,000$ . Distance to OFF3 = 7,000. FD = Distance to OFF3 + OFF3's RD =  $7,000 + 11,000 = 18,000$ .  
**Path via OFF4:** OFF4's RD =  $17,000 + 1,000 = 18,000$ . Distance to OFF4 = 4,000. FD = Distance to OFF4 + OFF4's RD =  $4,000 + 18,000 = 22,000$ .

Затем маршрутизатор OFF1 проверяет результаты этих вычислений и определяет, что кратчайшее расстояние до сети 10.1.1.0/24 проходит через маршрутизатор OFF2, поскольку путь через OFF2 имеет самый низкий FD (16 000). Этот путь, определяемый как кратчайший, считается следующим маршрутом. Затем маршрутизатор OFF1 пытается определить, соответствует ли любой из других маршрутов условию выполнимости EIGRP. В частности, маршрутизатор OFF1 проверяет, чтобы увидеть, что RD от маршрутизаторов OFF3 или OFF4 меньше, чем FD последующего маршрута. В случае OFF3 его RD в 11 000 действительно меньше, чем FD последующего маршрута (который составляет 16 000). Таким образом, путь к сети 10.1.1.0 /24 через OFF3 квалифицируется как возможный маршрут-преемник. Однако маршрут через OFF4 не подходит, потому что RD OFF4 из 18 000 больше, чем 16 000 (FD последующего маршрута). В результате путь к сети 10.1.1.0/24 через маршрутизатор OFF4 не считается возможным маршрутом-преемником.

## 4. Установка К-значений в EIGRP

В первой статье серии EIGRP мы познакомились с функциями EIGRP, рассмотрели пример базовой конфигурации и набор команд проверки. Сегодня, в этой статье, мы углубимся в понимание того, как EIGRP устанавливает соседство, изучает маршрут к сети, определяет оптимальный маршрут к этой сети, и пытается ввести этот маршрут в таблицу IP-маршрутизации маршрутизатора.

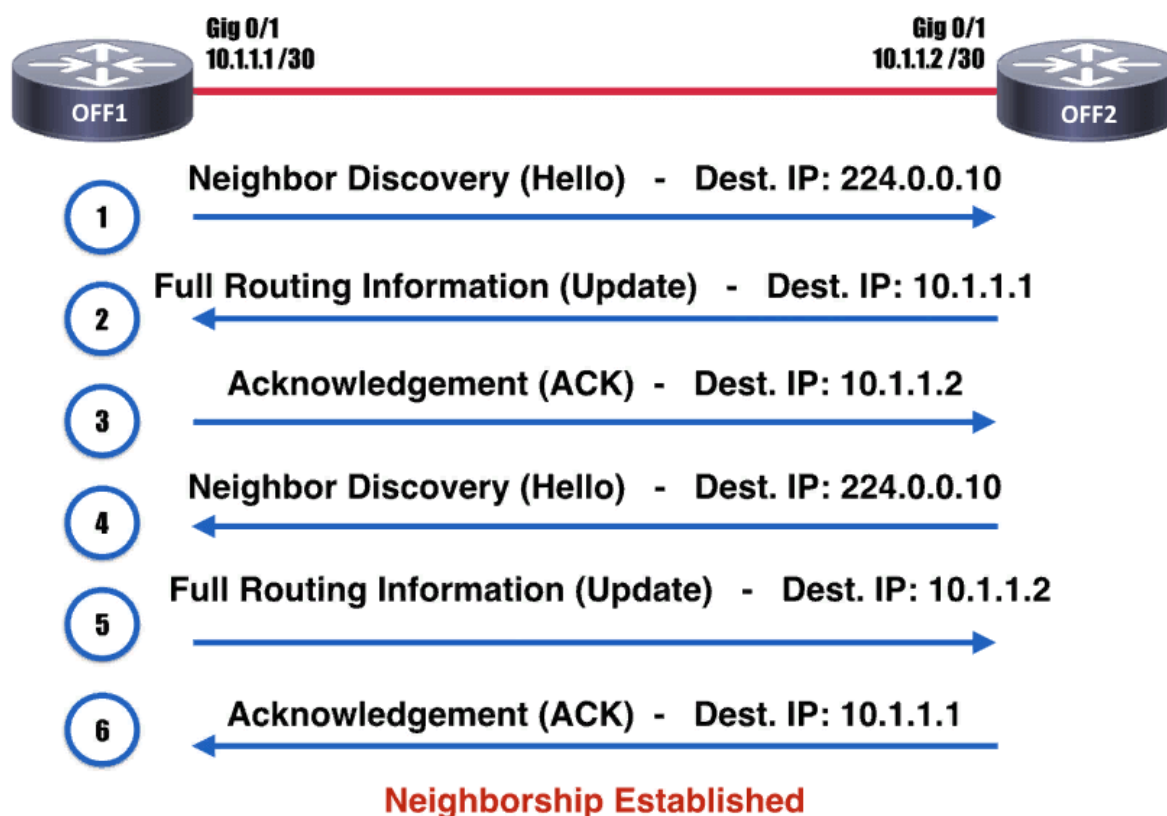
Операции EIGRP могут быть концептуально упрощены в три основных этапа:

- **Этап 1. Обнаружение соседей:** посредством обмена приветственными сообщениями EIGRP-спикер маршрутизаторы обнаруживают друг друга, сравнивают параметры (например, номера автономной системы, К-значения и сетевые адреса) и определяют, должны ли они образовывать соседство.
- **Этап 2. Обмен топологиями:** если соседние EIGRP маршрутизаторы решают сформировать соседство, они обмениваются своими полными таблицами топологии друг с другом. Однако после установления соседства между маршрутизаторами передаются только изменения существующей топологии. Этот подход делает EIGRP намного более эффективным, чем протокол маршрутизации, такой как RIP, который объявляет весь свой список известных сетей через определенные интервалы времени.
- **Этап 3. Выбор маршрутов:** как только таблица топологии EIGRP маршрутизатора заполнена, процесс EIGRP проверяет все изученные сетевые маршруты и выбирает лучший маршрут к каждой сети. EIGRP считает, что сетевой маршрут с самой низкой метрикой является лучшим маршрутом к этой сети.

Очень важно, что в когда вы читаете вышеописанные этапы, подробно описывающее обнаружение соседей EIGRP, обмен топологией и выбор маршрута, должны понимать, что в EIGRP, в отличие от OSPF, нет понятия назначенного маршрутизатора (DR) или резервного назначенного маршрутизатора (BDR).

### ОБНАРУЖЕНИЕ СОСЕДЕЙ И ОБМЕН ТОПОЛОГИЯМИ

Чтобы лучше понять, как маршрутизатор EIGRP обнаруживает своих соседей и обменивается информацией о топологии с этими соседями, рассмотрим рисунок ниже.



Шесть шагов, изображенных на рисунке выше, выполняются следующим образом:

- **Шаг 1.** Маршрутизатор OFF1 хочет видеть, есть ли какие-либо EIGRP-спикер маршрутизаторы вне его интерфейса Gig 0/1, с которым он мог бы, возможно, сформировать соседство. Таким образом, он осуществляет многоадресную рассылку приветственного сообщения EIGRP (EIGRP Hello) на хорошо известный EIGRP multicast-адрес 224.0.0.10 с просьбой к любым EIGRP-спикер маршрутизаторам, идентифицировать себя.
- **Шаг 2.** После получения приветственного сообщения маршрутизатора OFF1 маршрутизатор OFF2 отправляет одноадресное сообщение обновления (unicast Update message) обратно на IP-адрес маршрутизатора OFF1 10.1.1.1. Это сообщение обновления содержит полную таблицу топологии EIGRP маршрутизатора OFF2.
- **Шаг 3.** Маршрутизатор OFF1 получает обновление маршрутизатора OFF2 и отвечает одноадресным сообщением подтверждения (Acknowledgement (ACK), отправленным на IP-адрес маршрутизатора OFF2 10.1.1.2.
- **Шаг 4.** Затем процесс повторяется, и роли меняются местами. В частности, маршрутизатор OFF2 отправляет приветственное сообщение на адрес многоадресной рассылки EIGRP 224.0.0.10.
- **Шаг 5.** Маршрутизатор OFF1 отвечает на приветственное сообщение маршрутизатора OFF2 одноадресным обновлением (unicast Update), содержащим полную таблицу топологии EIGRP маршрутизатора OFF1. Это unicast Update достигается IP-адрес маршрутизатора OFF2 10.1.1.2.
- **Шаг 6.** Маршрутизатор OFF2 получает информацию о маршрутизации маршрутизатора OFF1 и отвечает одноадресным сообщением ACK, отправленным на IP-адрес маршрутизатора OFF1 10.1.1.1.

На этом этапе было установлено соседство EIGRP между маршрутизаторами OFF1 и OFF2. Маршрутизаторы будут периодически обмениваться приветственными сообщениями, чтобы подтвердить, что сосед каждого маршрутизатора все еще присутствует. Однако это последний раз, когда маршрутизаторы обмениваются своей полной информацией о маршрутизации. Последующие изменения топологии объявляются через частичные обновления, а не полные обновления, используемые во время создания соседства. Кроме того, обратите внимание, что сообщения обновления во время установления соседа были отправлены как одноадресные сообщения. Однако будущие сообщения обновления отправляются как многоадресные сообщения, предназначенные для 224.0.0.10. Это гарантирует, что все EIGRP-спикер маршрутизаторы на сегменте получают сообщения об обновлении.

EIGRP имеет преимущество перед OSPF в том, как он отправляет свои сообщения об обновлении. В частности, сообщения об обновлении EIGRP отправляются с использованием надежного транспортного протокола (Reliable Transport Protocol (RTP)). Это означает, что, в отличие от OSPF, если сообщение обновления будет потеряно при передаче, он будет повторно отправлен.

Примечание: аббревиатура RTP также относится к Real-time Transport Protocol (RTP), который используется для передачи голосовых и видеопакетов.

## ВЫБОР МАРШРУТА

Маршруты, показанные в таблице топологии EIGRP, содержат метрическую информацию, указывающую, насколько "далеко" она находится от конкретной целевой сети. Но как именно рассчитывается эта метрика? Расчет метрики EIGRP немного сложнее, чем с RIP или OSPF. В частности, метрика EIGRP по умолчанию является целочисленным значением, основанным на пропускной способности и задержке. Также, вычисление метрики может включать и другие компоненты. Рассмотрим формулу вычисления метрики EIGRP:

$$\text{Metric} = \left[ \left( K_1 * BW_{\min} + \frac{K_2 * BW_{\min}}{256 - \text{load}} + K_3 * \text{delay} \right) * \frac{K_5}{K_4 + \text{reliability}} \right] * 256$$

$$\text{where } BW_{\min} = \frac{10^7}{\text{least-bandwidth}}$$

Обратите внимание, что расчет метрики включает в себя набор K-значений, которые являются константами, принимающие нулевые значения или некоторые положительные целые числа. Расчет также учитывает пропускную способность, задержку, нагрузку и надежность (bandwidth, delay, load, reliability). Интересно, что большая часть литературы по EIGRP утверждает, что

метрика также основана на Maximum Transmission Unit (MTU). Однако, как видно из формулы расчета метрики, MTU отсутствует. Так в чем же дело? Учитывает ли EIGRP MTU интерфейса или нет?

В самом начале разработки EIGRP, MTU был обозначен как Тай-брейкер, если два маршрута имели одинаковую метрику, но разные значения MTU. В такой ситуации был бы выбран маршрут с более высоким MTU. Таким образом, хотя сообщение об обновлении EIGRP действительно содержит информацию MTU, эта информация непосредственно не используется в расчетах метрик.

Далее, давайте рассмотрим каждый компонент расчета метрики EIGRP и tiebreaking MTU:

- **Bandwidth (Пропускная способность):** значение пропускной способности, используемое в расчете метрики EIGRP, определяется путем деления 10 000 000 на пропускную способность (в Кбит / с) самого медленного канала вдоль пути к целевой сети.
- **Delay (Задержка):** в отличие от полосы пропускания, которая представляет собой "самое слабое звено", значение задержки является кумулятивным. В частности, это сумма всех задержек, связанных со всеми интерфейсами, которые используются чтобы добраться до целевой сети. Выходные данные команды `show interfaces` показывают задержку интерфейса в микросекундах. Однако значение, используемое в расчете метрики EIGRP, выражается в десятках микросекунд. Это означает, что вы суммируете все задержки выходного интерфейса, как показано в выводе `show interfaces` для каждого выходного интерфейса, а затем делите на 10, чтобы получить единицу измерения в десятки микросекунд.
- **Reliability (Надежность):** надежность-это значение, используемое в числителе дроби, с 255 в качестве ее знаменателя. Значение дроби указывает на надежность связи. Например, значение надежности 255 указывает на то, что связь надежна на 100 процентов (то есть  $255/255 = 1 = 100$  процентов).
- **Load (Нагрузка):** как и надежность, нагрузка-это значение, используемое в числителе дроби, с 255 в качестве ее знаменателя. Значение дроби указывает, насколько занята линия. Например, значение нагрузки 1 указывает на то, что линия загружена минимально (то есть  $1/255 = 0,004$  1%)
- **MTU:** хотя он не отображается в Формуле вычисления метрики EIGRP, значение MTU интерфейса (которое по умолчанию составляет 1500 байт) переносится в сообщение обновления EIGRP, которое будет использоваться в случае привязки (например, два маршрута к целевой сети имеют одну и ту же метрику, но разные значения MTU), где предпочтительно более высокое значение MTU.

Для улучшения запоминания используйте следующий алгоритм **Big Dogs Really Like Me**. Где **B** в слове **Big** ассоциируется с первой буквой в слове **B**andwidth. Буква **D** в слове **Dogs** соответствует первой букве **D** в слове **D**elay, и так далее.

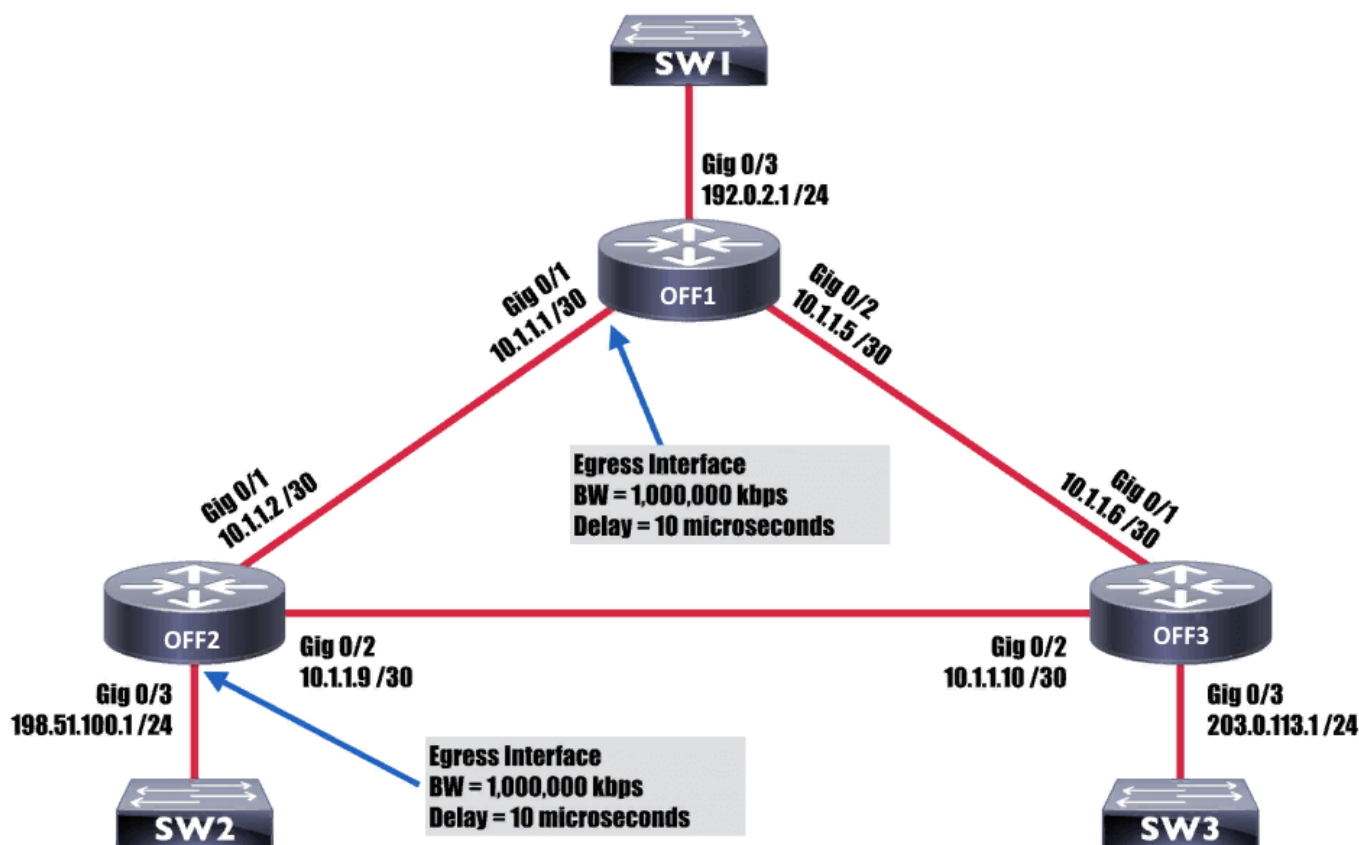
Однако по умолчанию EIGRP имеет большинство своих K-значений равными нулю, что значительно упрощает расчет метрики, учитывая только пропускную способность и задержку. В частности, значения K по умолчанию являются:

- **K1 = 1**
- **K2 = 0**
- **K3 = 1**
- **K4 = 0**
- **K5 = 0**

Если мы подставим эти дефолтные значения K в расчет метрики EIGRP, то значение каждой дроби будет равно нулю, что сводит формулу к следующему:

$$\text{Metric} = \left[ \left( \frac{10^7}{\text{least-bandwidth}} \right) + \text{cumulative-delay} \right] * 256$$

Чтобы закрепить знания по вычислению метрики, давайте проведем расчет метрики и посмотрим, соответствует ли она нашей таблице топологии EIGRP. Рассмотрим топологию, показанную на рисунке ниже.



Предположим, что мы хотим вычислить метрику для сети 198.51.100.0/24 от роутера OFF1 для маршрута, который идет от OFF1 до OFF2, а затем выходит в целевую сеть. Из топологии мы можем определить, что нам нужно будет выйти с двух интерфейсов маршрутизатора, чтобы добраться от маршрутизатора OFF1 до сети 198.51.100.0 /24 через маршрутизатор OFF2. Эти два выходных интерфейса являются интерфейсами Gig0/1 на маршрутизаторе OFF1 и интерфейсом Gig0/3 на маршрутизаторе OFF2. Мы можем определить пропускную способность и задержку, связанные с каждым интерфейсом, изучив выходные данные команд `show interfaces`, приведенных в следующем примере.

Определение значений пропускной способности и задержки интерфейса на маршрутизаторах OFF1 и OFF2

```
OFF1#show interfaces gig 0/1
GigabitEthernet0/1 is up, line protocol is up
  Hardware is iGbE, address is fa16.3e90.386f (bia fa16.3e90.386f)
  Internet address is 10.1.1.1/30
  MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
    reliability 255/255, txload 1/255, rxload 1/255
.. OUTPUT OMITTED ..
```

```
OFF2#show interfaces gig 0/3
GigabitEthernet0/3 is up, line protocol is up
  Hardware is iGbE, address is fa16.3ec7.46e6 (bia fa16.3ec7.46e6)
  Internet address is 198.51.100.1/24
  MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
    reliability 255/255, txload 1/255, rxload 1/255
```

Из приведенного выше примера мы видим, что оба выходных интерфейса имеют пропускную способность 1 000 000 Кбит/с (то есть 1 Гбит/с). Кроме того, мы видим, что каждый выходной интерфейс имеет задержку в 10 микросекунд. Значение пропускной способности, которое мы вводим в нашу формулу вычисления метрики EIGRP, - это пропускная способность

самого медленного канала на пути к целевой сети, измеряемая в Кбит/с. В нашем случае оба выходных интерфейса имеют одинаковую скорость соединения, то есть мы говорим, что наша "самая медленная" связь составляет 1 000 000 Кбит/с.

Для примера ниже показаны общие значения по умолчанию для пропускной способности и задержки на различных типах интерфейсов маршрутизатора Cisco.

Общие значения по умолчанию для пропускной способности и задержки интерфейса:

Interface Type	Bandwidth (kbps)	Delay (microseconds)
Serial	1544	20,000
Gigabit Ethernet	1,000,000	10
Fast Ethernet	100,000	100
Ethernet	10,000	1000

Наше значение задержки может быть вычислено путем сложения задержек выходного интерфейса (измеренных в микросекундах) и деления на 10 (чтобы дать нам значение, измеренное в десятках микросекунд). Каждый из наших двух выходных интерфейсов имеет задержку в 10 микросекунд, что дает нам суммарную задержку в 20 микросекунд. Однако мы хотим, чтобы наша единица измерения была в десятках микросекунд. Поэтому мы делим 20 микросекунд на 10, что дает нам 2 десятка микросекунд. Теперь у нас есть два необходимых значения для нашей формулы: **пропускная способность = 1 000 000 Кбит/с** и **задержка = 2 десятка микросекунд**. Теперь давайте добавим эти значения в нашу формулу:

$$\begin{aligned} \text{Metric} &= \left[ \left( \frac{10^7}{1,000,000} \right) + 2 \right] * 256 \\ &= [10 + 2] * 256 \\ &= 12 * 256 \\ &= 3072 \end{aligned}$$

Вычисленное значение показателя EIGRP составляет 3072. Теперь давайте посмотрим, является ли это фактической метрикой, появляющейся в таблице топологии EIGRP маршрутизатора OFF1. Выходные данные команды `show ip eigrp topology`, выведенные на маршрутизаторе OFF1, показаны в следующем примере.

Проверка метрики EIGRP для сети 198.51.100.0/24 на маршрутизаторе OFF1



OFF1#show ip eigrp topology

EIGRP-IPv4 Topology Table for AS(1)/ID(192.0.2.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,  
r - reply Status, s - sia Status

```
P 192.0.2.0/24, 1 successors, FD is 2816
    via Connected, GigabitEthernet0/3
P 10.1.1.8/30, 2 successors, FD is 3072
    via 10.1.1.2 (3072/2816), GigabitEthernet0/1
    via 10.1.1.6 (3072/2816), GigabitEthernet0/2
P 203.0.113.0/24, 1 successors, FD is 3072
    via 10.1.1.6 (3072/2816), GigabitEthernet0/2
P 10.1.1.4/30, 1 successors, FD is 2816
    via Connected, GigabitEthernet0/2
P 198.51.100.0/24, 1 successors, FD is 3072
    via 10.1.1.2 (3072/2816), GigabitEthernet0/1
P 10.1.1.0/30, 1 successors, FD is 2816
    via Connected, GigabitEthernet0/1
```

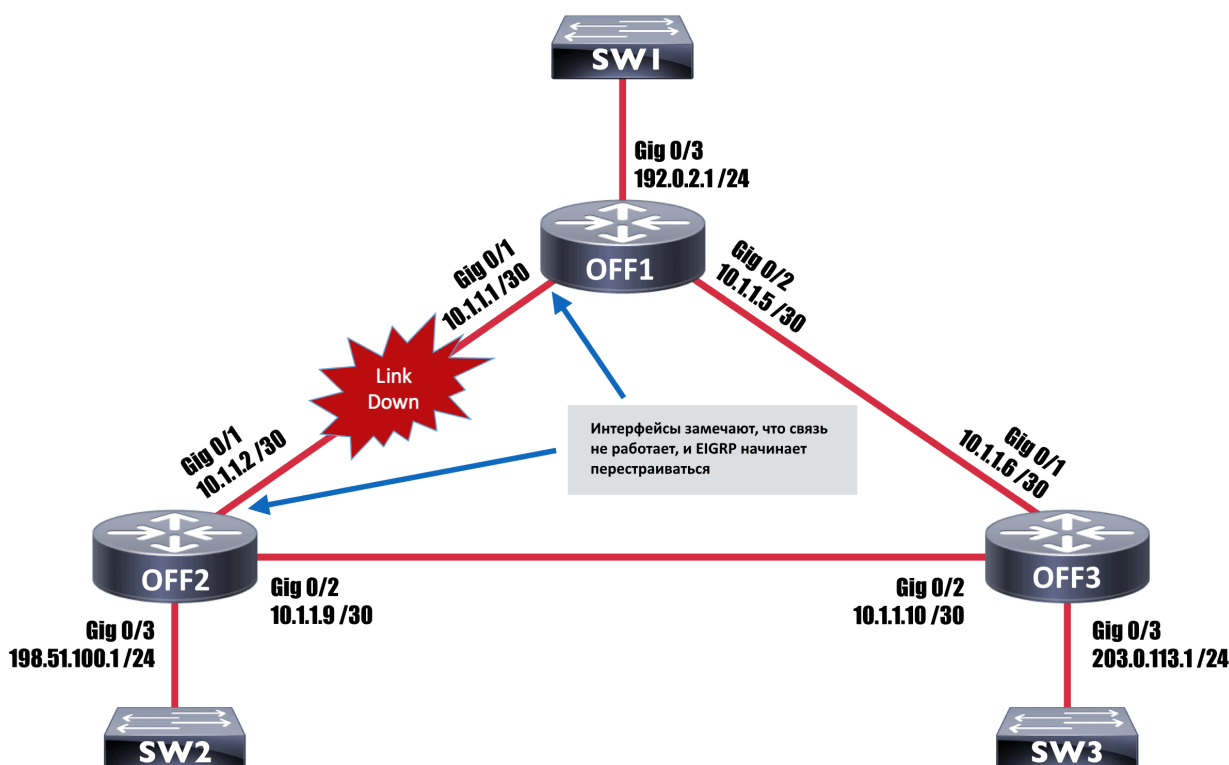
Как и предполагалось, метрика (также известная как допустимое расстояние) от маршрутизатора OFF1 до Сети 198.51.100.0 /24 через маршрутизатор OFF2 составляет 3072. Напомним, что в этом примере мы использовали значения К по умолчанию, что также является обычной практикой в реальном мире. Однако для целей проектирования мы можем манипулировать К-значениями. Например, если мы обеспокоены надежностью каналом связи или нагрузкой, которую мы могли бы испытать на линии, мы можем манипулировать нашими К-значениями таким образом, чтобы EIGRP начал бы рассматривать надежность и/или нагрузку в своем метрическом расчете.



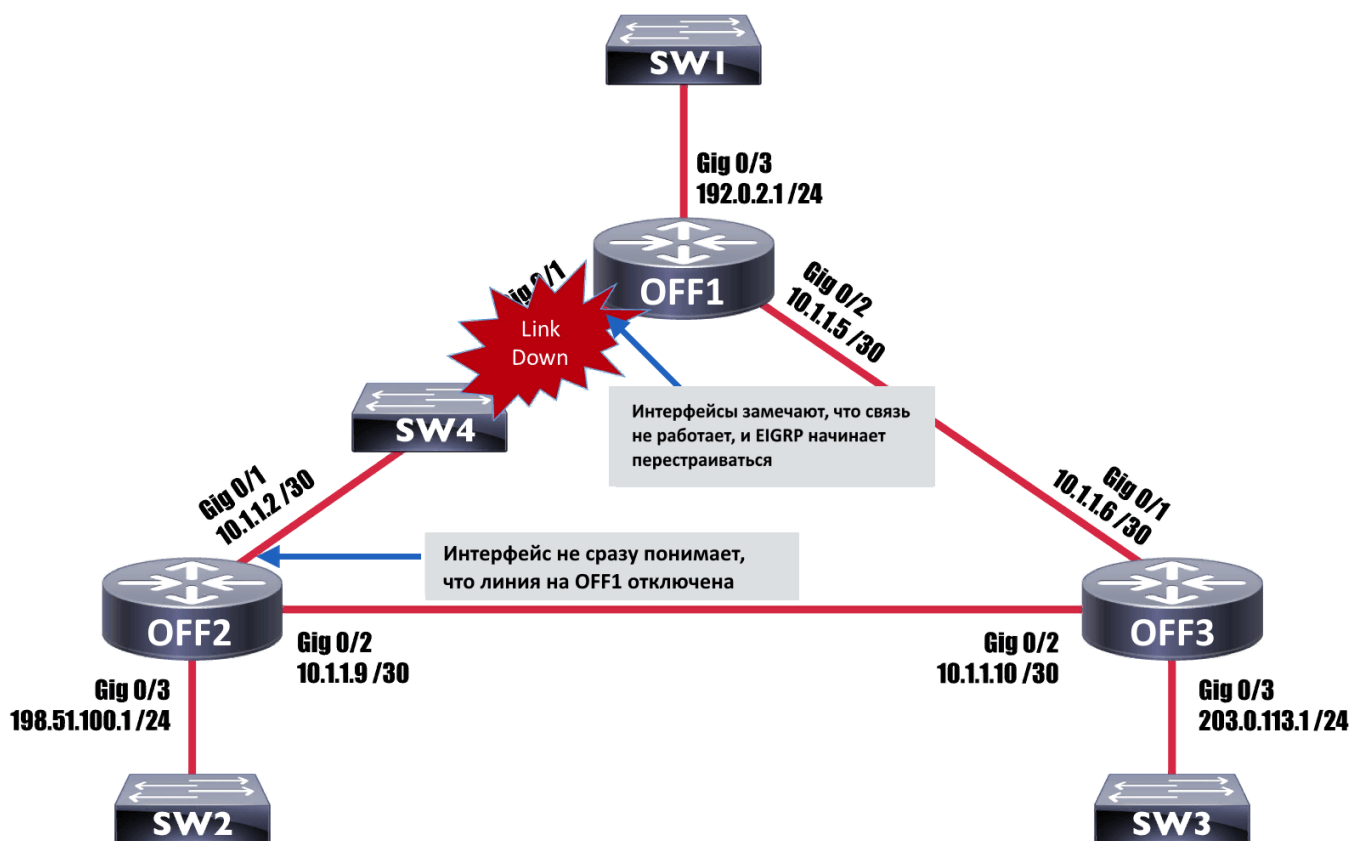
## 5. Конвергенция EIGRP – настройка таймеров

Одним из преимуществ и популярности EIGRP является его быстрая конвергенция в случае сбоя связи. Однако одно, что может замедлить эту конвергенцию, – это конфигурация таймера. Именно этому посвящена эта статья, которая является третьей в серии статей о понимании EIGRP.

Начнем наше обсуждение таймеров EIGRP с рассмотрения ситуации, когда два соседа EIGRP непосредственно связаны друг с другом. Если физическая связь между ними не работает, подключенный интерфейс каждого роутера отключается, и EIGRP может перейти на резервный путь (то есть возможный маршрут приемника). Такая ситуация показана на следующем рисунке:

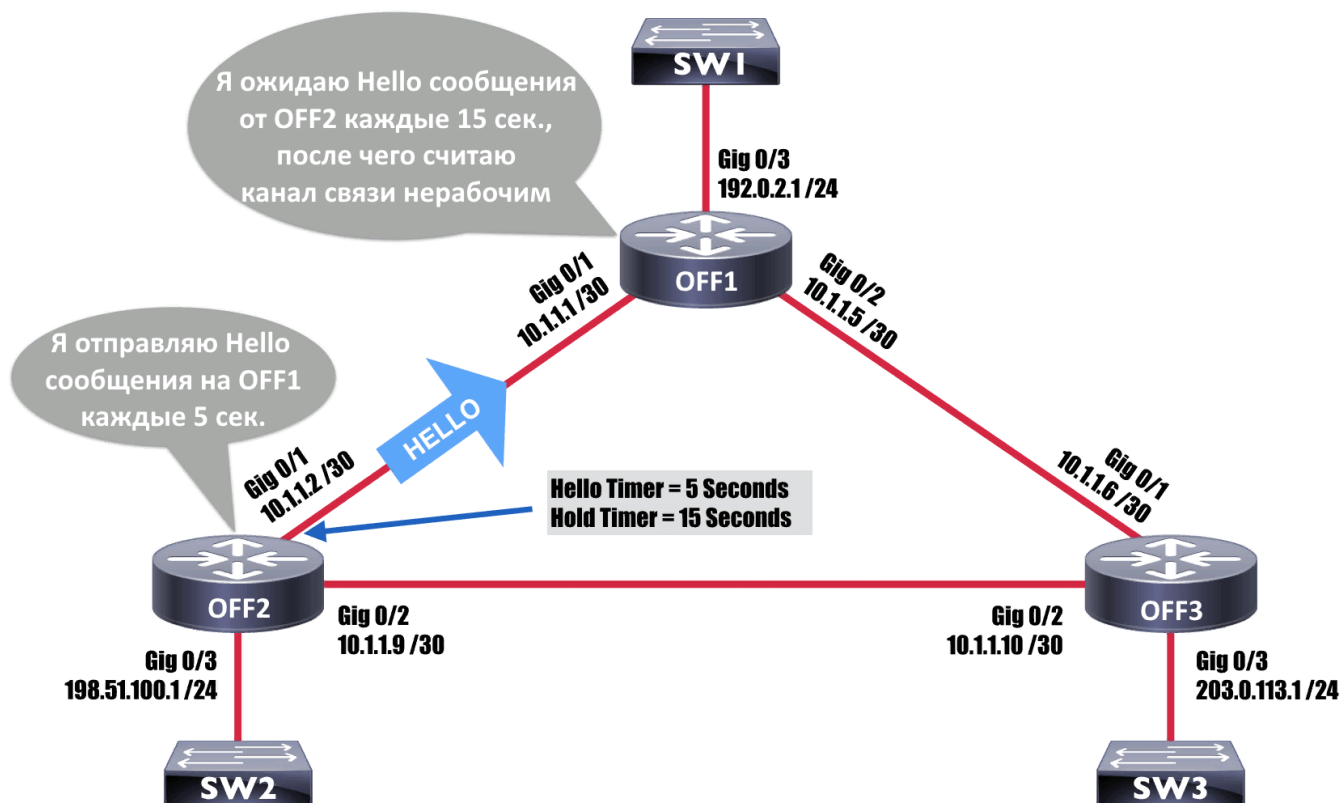


Роутеры OFF1 и OFF2, показанные на приведенном выше рисунке, соединены друг с другом. Поэтому, если кабель между ними обрывается, каждый из интерфейсов роутера, соединяющихся с этим звеном, отключаются, и EIGRP понимает, что он просто потерял соседа и начинает перестраиваться. Однако нарушение связи между несколькими соседями EIGRP не всегда так очевидно. Например, рассмотрим вариант предыдущей топологии, как показано ниже:



Обратите внимание, что между роутерами OFF1 и OFF2 был подключен коммутатор (SW4) на рисунке выше. Если происходит сбой соединения между коммутатором SW4 и роутером OFF1, роутер OFF2 не сразу осознает это, потому что его порт Gig0/1 все еще находится в состоянии up/up. В результате роутер OFF2 может продолжать считать, что роутер OFF1 - это наилучший путь для доступа к сети, такой как 192.0.2.0 /24. К счастью, EIGRP использует таймеры, чтобы помочь EIGRP-спикер роутерам определить, когда они потеряли связь с соседом по определенному интерфейсу.

Таймеры, используемые EIGRP, - это таймеры Hello и Hold. Давайте задержимся на мгновение, чтобы изучить их работу, потому что таймер Hold не ведет себя интуитивно. Во-первых, рассмотрим таймер Hello. Как вы можете догадаться, это определяет, как часто интерфейс роутера отправляет приветственные сообщения своему соседу. Однако таймер Hold интерфейса - это не то, как долго этот интерфейс ожидает получения приветственного сообщения от своего соседа, прежде чем считать этого соседа недоступным. Таймер Hold - это значение, которое мы посылаем соседнему роутеру, сообщая этому соседнему роутеру, как долго нас ждать, прежде чем считать нас недоступными. Эта концепция проиллюстрирована на рисунке ниже, где роутер OFF2 настроен с таймером Hello 5 секунд и таймером Hold 15 секунд.



Два больших вывода из этого рисунка таковы:

- Таймер Hello роутера OFF2 влияет на то, как часто он посылает приветствия, в то время как таймер Hold роутера OFF2 влияет на то, как долго роутер OFF1 будет ждать приветствий роутера OFF2.
- Указанное время Hello и Hold является специфичным для интерфейса Gig 0/1 роутера OFF2. Другие интерфейсы могут быть сконфигурированы с различными таймерами.

Поскольку таймер Hold, который мы отправляем, на самом деле является инструкцией, сообщаемой соседнему роутеру, как долго нас ждать, а не как долго мы ждем Hello-сообщения соседа, причем у каждого соседа может быть свой набор таймеров. Однако наличие совпадающих таймеров между соседями считается лучшей практикой для EIGRP (и является требованием для OSPF).

Чтобы проиллюстрировать конфигурацию и проверку таймеров EIGRP, допустим, что роутер OFF1 имел таймер Hello 1 секунду и таймер Hold 3 секунды на своем интерфейсе Gig 0/1 (подключение к OFF2). Затем мы захотели, чтобы роутер OFF2 имел таймер Hello 5 секунд и таймер Hold 15 секунд на своем интерфейсе Gig 0/1 (подключение к роутеру OFF1). Такая конфигурация укрепляет понятие того, что соседи EIGRP не требуют совпадающих таймеров (хотя лучше всего иметь совпадающие таймеры). В следующем примере показана эта конфигурация таймера для роутеров OFF1 и OFF2.

```

OFF1#conf term
Enter configuration commands, one per line. End with CNTL/Z.
OFF1(config)#int gig 0/1
OFF1(config-if) #ip hello-interval eigrp 1 1
OFF1(config-if) #ip hold-time eigrp 1 3
OFF1(config-if) #end
OFF1#

OFF2#conf term
Enter configuration commands, one per line. End with CNTL/Z.
OFF2(config)#int gig 0/1
OFF2 (config-if) #ip hello-interval eigrp 1 5
OFF2 (config-if) #ip hold-time eigrp 1 15
OFF2(config-if) #end
OFF2#

```

Команда `ip hello-interval eigrp asn h_intls` вводится на каждом роутере для установки таймеров Hello. Параметр `asn` определяет настроенную автономную систему EIGRP равным 1, и таймер Hello для роутера OFF1 настроен равным 1 секунде, в то время как таймер Hello для роутера OFF2 настроен равным 5 секундам. Аналогично, команда `ip hold-time eigrp asn ho_t` вводится на каждом роутере для установки таймеров Hold. Опять же, обе команды задают автономную систему 1. Таймер Hold роутера OFF1 настроен на 3 секунды, в то время как таймер Hold роутера OFF2 настроен на 15 секунд. В обоих случаях таймер Hold EIGRP был настроен таким образом, чтобы быть в три раза больше таймера Hello. Хотя такой подход является обычной практикой, он не является обязательным требованием. Кроме того, вы должны быть осторожны, чтобы не установить таймер Hold на роутере со значением меньше, чем таймер Hello. Такая неверная конфигурация может привести к тому, что соседство будет постоянно "падать" и восстанавливаться. Интересно, что Cisco IOS действительно принимает такую неправильную конфигурацию, не сообщая ошибки или предупреждения.

EIGRP использует таймер Hello по умолчанию 5 секунд и таймер Hold по умолчанию 15 секунд на LAN интерфейсах. Однако в некоторых ситуациях на интерфейсах, настроенных для Frame Relay, таймеры по умолчанию будут больше.

Далее, посмотрим, как мы можем проверить настройки таймера EIGRP. Команда `show ip eigrp neighbors`, как показано в примере ниже, показывает оставшееся время удержания для каждого соседа EIGRP.

OFF2#show ip eigrp neighbors

EIGRP-IPv4 Neighbors for AS(1)

H	Address	Interface	Hold (sec)	Uptime	SRTT (ms)	RTO	Q Cnt	Seq Num
0	10.1.1.1	Gi0/1	2	00:05:51	7	100	0	14
1	10.1.1.10	Gi0/2	13	02:07:22	86	516	0	8

Обратите внимание в приведенном выше примере, что значение в столбце Hold равно 2 секундам для роутера OFF1 (то есть 10.1.1.1) и 13 секундам для роутера OFF3 (то есть 10.1.1.10). Эти цифры говорят нам о не настроенных таймерах Hold. Они говорят нам, сколько времени остается до того, как роутер OFF2 отключит этих соседей, в отсутствие приветственного сообщения от этих соседей. Роутер OFF2 перезапускает свой обратный отсчет времени Hold для роутера OFF3 до 15 секунд (таймер Hold роутера OFF3) каждый раз, когда он получает Hello сообщение от OFF3 (которое OFF3 отправляет каждые 5 секунд на основе своего таймера Hello). Поэтому, если вы повторно выполните команду `show ip eigrp neighbors` на роутере OFF2, вы, вероятно, увидите оставшееся время Hold для роутера OFF3 где - то в диапазоне 10-14 секунд. Однако, поскольку роутер OFF1 настроен с таймером Hold 3 секунды и таймером Hello 1 секунды, оставшееся время Hold, зафиксированно на роутере OFF2 для его соседства с роутером OFF1, обычно должно составлять 2 секунды.

Мы можем видеть настроенные значения таймера Hello и Hold для интерфейса роутера, выполнив команду `show ip eigrp interfaces detail interface_id`, как показано в примере ниже. Вы можете видеть в выходных данных, что интерфейс Gig 0/1 на роутере OFF2 имеет таймер Hello 5 секунд и таймер Hold 15 секунд.

OFF2#show ip eigrp interfaces detail gig 0/1

EIGRP-IPv4 Interfaces for AS(1)

Interface	Peers	Xmit Queue Un/Reliable	PeerQ Un/Reliable	Mean SRTT	Pacing Time Un/Reliable	Multicast Flow Timer	Pending Routes
Gi0/1	1	0/0	0/0	7	0/0	50	0

Hello-interval is 5, Hold-time is 15  
 Split-horizon is enabled  
 Next xmit serial <none>  
 Packetized sent/expedited: 4/0  
 Hello's sent/expedited: 1880/3  
 Un/reliable mcasts: 0/5 Un/reliable ucasts: 5/3  
 Mcast exceptions: 0 CR packets: 0 ACKs suppressed: 0  
 Retransmissions sent: 1 Out-of-sequence rcv'd: 3  
 Topology-ids on interface - 0  
 Authentication mode is not set  
 Topologies advertised on this interface: base  
 Topologies not advertised on this interface:

OFF2 #

## 6. Пассивные интерфейсы в EIGRP

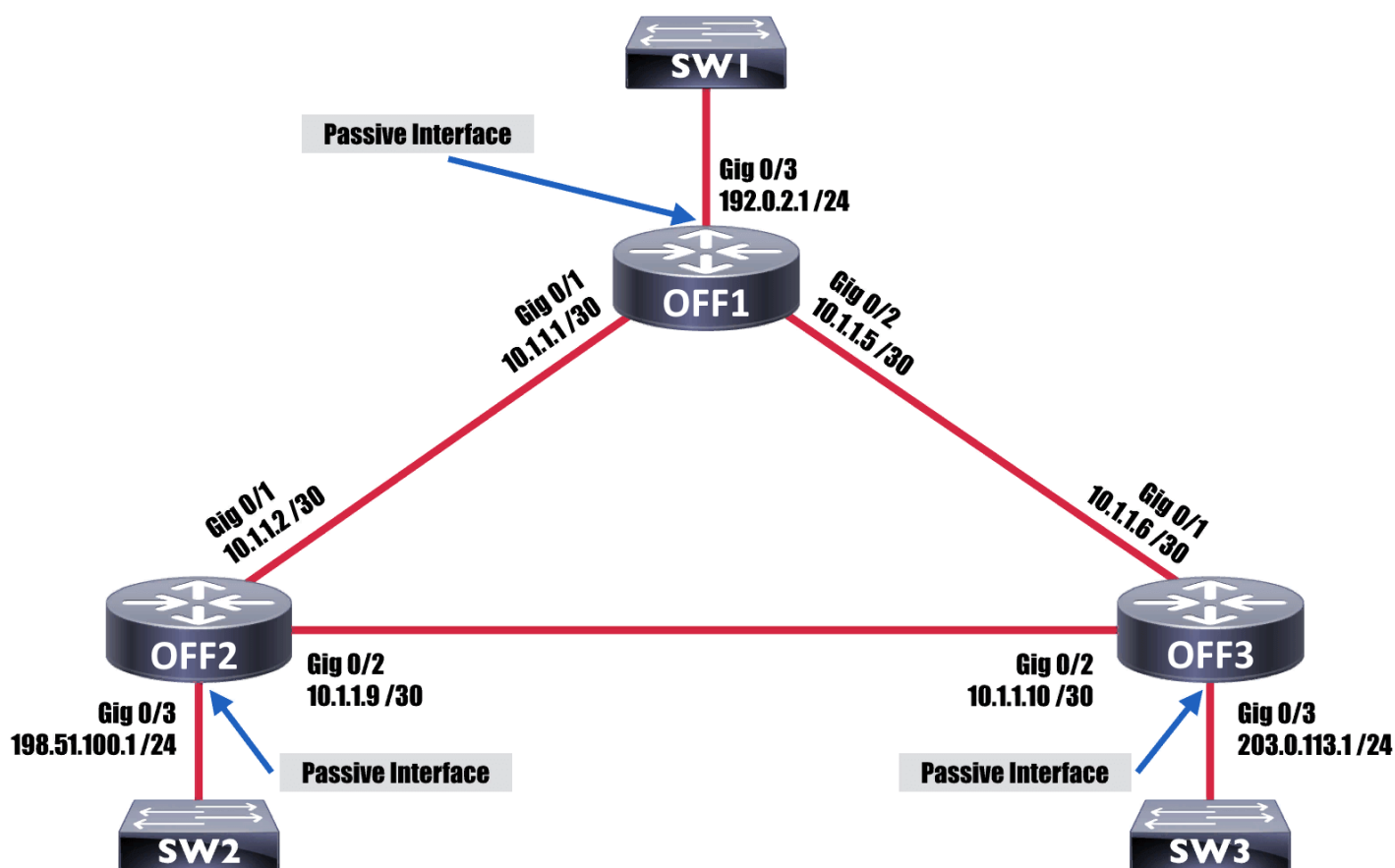
Иногда нам хочется, чтобы интерфейс роутера участвовал в процессе маршрутизации EIGRP, но без отправки Hello сообщений EIGRP с этого интерфейса. Именно об этом мы и поговорим в этой статье.

Ранее мы говорили о команде `Network net-id wildcard-mask`, вводимой в режиме конфигурации роутера EIGRP. Эта команда вызывает два основных действия:

1. Отправляет EIGRP Hello multicast сообщения с любого интерфейса, чей IP-адрес попадает в сетевое адресное пространство, указанное командой `network`.
2. Объявляет подсеть любого интерфейса, IP-адрес которого попадает в сетевое адресное пространство, заданное командой `network`.

Однако в некоторых случаях нам нет необходимости в том, чтобы команда `network` выполняла первое действие, указанное выше. Например, если интерфейс подключается к хостам в локальной сети, а не к другим EIGRP-спикер роутерам. В этом случае нет необходимости отправлять Hello сообщения с этого интерфейса. К счастью, мы можем выборочно отключать отправку приветствий с интерфейса, все еще объявляя подсеть этого интерфейса нашим соседям EIGRP. Это стало возможным благодаря функции пассивного интерфейса.

Рассмотрим топологию ниже:



Обратите внимание, что каждый роутер имеет интерфейс, указывающий на сегмент локальной сети (то есть интерфейс, подключенный к коммутатору). Мы действительно хотим, чтобы подсети этих интерфейсов объявлялись через EIGRP, но нам не надо отправлять Hello сообщения с этого интерфейса (поскольку они не подключаются ни к каким другим EIGRP - спикер роутерам). Это делает эти интерфейсы (то есть интерфейс Gig0/3 на роутерах OFF1, OFF2 и OFF3) отличными кандидатами на роль пассивных интерфейсов. В следующем примере показано, как использовать команду `passive-interface interface_id`.

```

OFF1# configurationterminal
Enter configuration commands, one per line. End with CNTL/Z.
OFF1(config)#router eigrp 1
OFF1(config-router)#passive-interface gig0/3
OFF1(config-router)#end
OFF1#

OFF2# configurationterminal
Enter configuration commands, one per line. End with CNTL/Z.
OFF2 (config)#router eigrp 1
OFF2 (config-router)#passive-interface gig0/3
OFF2 (config-router)#end
OFF2#

OFF3# configurationterminal
Enter configuration commands, one per line. End with CNTL/Z.
OFF3 (config)#router eigrp 1
OFF3 (config-router)#passive-interface default
OFF3 (config-router)#no passive-interface gig0/1
OFF3 (config-router)#no passive-interface gig0/2
OFF3 (config-router)#end
OFF3#

```

В приведенном примере команда `passive-interface gig0/3` была введена на роутерах OFF1 и OFF2, чтобы сообщить, что эти роутеры должны блокировать отправку Hello сообщений со своих интерфейсов Gig0/3 (то есть интерфейсов, соединяющихся с сегментами локальной сети). Однако конфигурация на роутере OFF3 использует несколько иной подход. Вместо указания интерфейсов, которые должны быть пассивными, дается команда `passive-interface default`, которая делает все интерфейсы пассивными. Затем были даны команды `no passive-interface gig 0/1` и `no passive-interface gig 0/2`, чтобы выборочно сообщить, что эти интерфейсы не должны быть пассивными (так как эти интерфейсы используются для подключения к соседям EIGRP). Этот подход может быть полезен на роутерах с несколькими интерфейсами LAN и только несколькими интерфейсами, соединяющимися с соседями EIGRP.

Как только мы выполняем команду `passive-interface interface_id` для определенного интерфейса, этот интерфейс больше не появляется в выходных данных команды `show ip eigrp interfaces`, как показано в примере ниже. Обратите внимание, что интерфейс Gig0/3, который был настроен как пассивный интерфейс, не отображается в списке. Однако EIGRP все еще объявляет подсеть, к которой принадлежит интерфейс Gig0/3.

```

OFF2#show ip eigrp interfaces
EIGRP-IPv4 Interfaces for AS(1)

```

Interface	Peers	Xmit Queue Un/Reliable	PeerQ Un/Reliable	Mean SRTT	Pacing Time Un/Reliable	Multicast Flow Timer	Pending Routes
Gi0/1	1	0/0	0/0	4	0/0	50	0
Gi0/2	1	0/0	0/0	6	0/0	50	0

```

OFF2 #

```

Мы можем определить, какие интерфейсы на роутере действуют в качестве пассивных интерфейсов, выполнив команду `show ip protocols`. В отображаемых данных этой команды, как видно в примере ниже, обратите внимание, что интерфейс Gig0/3 на роутере OFF2 является пассивным интерфейсом, в то время как его подсеть (198.51.100.0/24) объявляется EIGRP.

## OFF2#show ip protocols

\*\*\* IP Routing is NSF aware \*\*\*

Routing Protocol is "application"

Sending updates every 0 seconds

Invalid after 0 seconds, hold down 0, flushed after 0

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Maximum path: 32

Routing for Networks:

Routing Information Sources:

Gateway	Distance	Last Update
---------	----------	-------------

Distance: (default is 4)

Routing Protocol is "eigrp 1"

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Default networks flagged in outgoing updates

Default networks accepted from incoming updates

EIGRP-IPv4 Protocol for AS(1)

Metric weight K1=1, K2=0, K3=1, K4=0, K5=0

Soft SIA disabled

NSF-aware route hold timer is 240

Router-ID: 198.51.100.1

Topology : 0 (base)

Active Timer: 3 min

Distance: internal 90 external 170

Maximum path: 4

Maximum hopcount 100

Maximum metric variance 1

Automatic Summarization: disabled

Maximum path: 4

Routing for Networks:

10.0.0.0

198.51.100.0

Passive Interface(s):

GigabitEthernet0/3

Routing Information Sources:

Gateway	Distance	Last Update
---------	----------	-------------

10.1.1.10	90	00:13:07
-----------	----	----------

10.1.1.1	90	00:13:07
----------	----	----------

Distance: internal 90 external 170

OFF2#



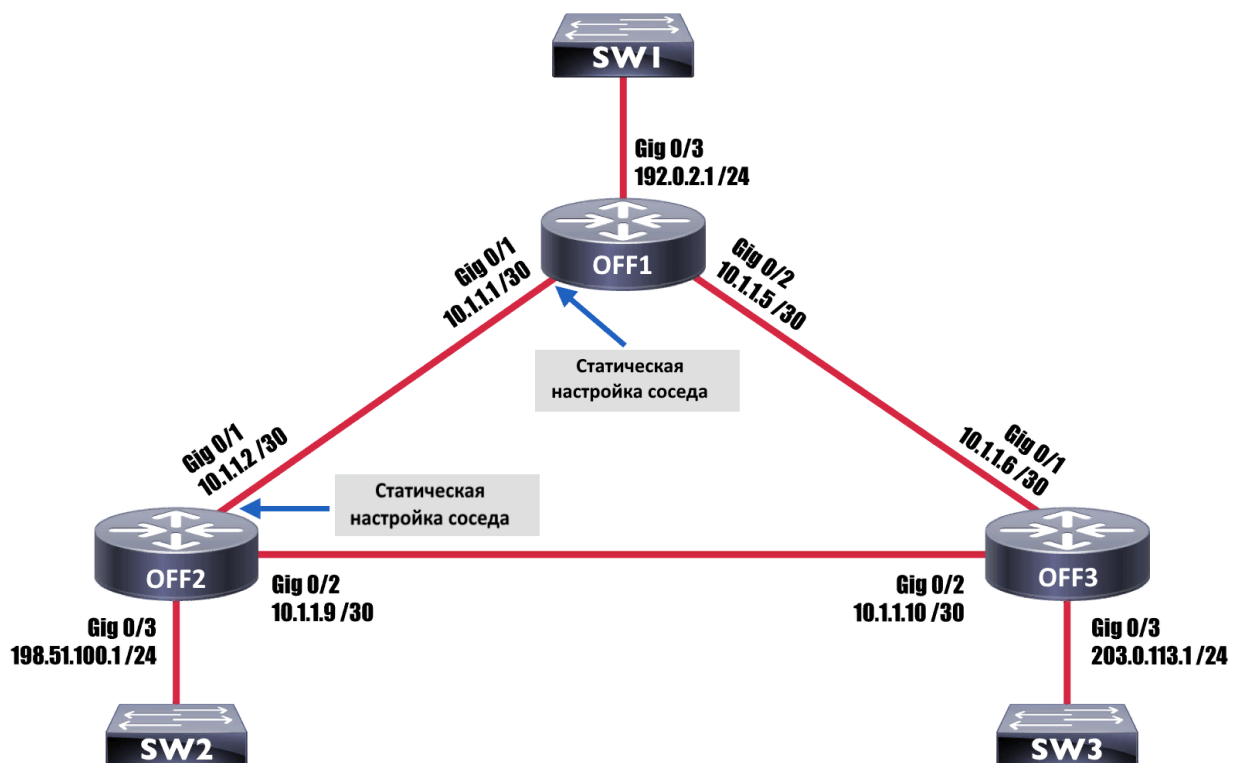
## 7. Настройка статического соседства в EIGRP

Как правило, EIGRP-спикер роутер динамически обнаруживает своих соседей, отправляя multicast Hello сообщения. Однако есть возможность статически настроить этих соседей и общаться с ними с помощью unicast сообщений. Это делается крайне редко, но в таких случаях может оказаться полезным.

Рассмотрим для примера Frame Relay WAN. Представьте себе, что роутер **A** имеет интерфейс, настроенный на десять постоянных виртуальных каналов Frame Relay (PVC). На другом конце двух этих PVC каналов находятся EIGRP-спикер роутеры. Однако другие восемь PVC каналов не подключены к EIGRP-спикер роутерам. В данной топологии, если бы WAN-интерфейс роутера A участвовал в EIGRP, то роутер A должен был бы реплицировать свое приветственное сообщение EIGRP и отправить копию всем десяти PVC, что привело бы к увеличению нагрузки на роутер A и увеличило использование полосы пропускания на других восьми PVC, не подключающихся к EIGRP роутеру. Это ситуация, при которой выигрыш состоит в статической настройке соседей EIGRP, а не от использования процесса обнаружения на основе многоадресной рассылки. Давайте рассмотрим вариант конфигурации статического соседства EIGRP в этой статье.

### СТАТИЧЕСКАЯ КОНФИГУРАЦИЯ СОСЕДСТВА

Команда `neighbor ip_address outgoing_interface` вводится в режиме конфигурации роутера EIGRP для статического указания соседства EIGRP. Обратите внимание, что эта настройка должна быть выполнена на обоих соседях. Кроме того, имейте в виду, что IP-адрес, указанный в команде **neighbor**, принадлежит той же подсети, что и указанный исходящий интерфейс. На основе топологии, показанной ниже, следующие примеры настроек показывают, как роутеры OFF1 и OFF2 статически указывают друг на друга, в отличие от использования динамического обнаружения.



```
OFF1#conf term
Enter configuration commands, one per line. End with CNTL/Z.
OFF1(config)#router eigrp 1
OFF1(config-router)#neighbor 10.1.1.2 gig 0/1
OFF1(config-router)#end
OFF1#

OFF2#conf term
Enter configuration commands, one per line. End with CNTL/Z.
OFF2(config)#router eigrp 1
OFF2(config-router)#neighbor 10.1.1.1 gig 0/1
OFF2(config-router)#end
OFF2#
```

На роутере OFF1 команда `neighbor 10.1.1.2 gig 0/1` введенная в режиме конфигурации роутера EIGRP, дает команду процессу EIGRP прекратить отправку многоадресных сообщений из интерфейса Gig 0/1 и вместо этого начать использовать одноадресные сообщения. Он также инструктирует процесс маршрутизации EIGRP попытаться установить соседство с EIGRP-спикер роутером, по IP-адресу 10.1.1.2 (то есть IP-адрес интерфейса Gig 0/1 роутера OFF2). Поскольку статическая конфигурация соседа должна выполняться на обоих концах канала, роутер OFF2 аналогично настроен для отправки одноадресных сообщений EIGRP со своего интерфейса Gig 0/1 и для установления соседства с EIGRP-спикер роутером с IP-адресом 10.1.1.1 (то есть IP-адресом интерфейса gig 0/1 роутера OFF1).

## ПРОВЕРКА СТАТИЧЕСКОГО СОСЕДСТВА

Чтобы определить, какие интерфейсы на роутере статически настроены с соседом EIGRP, можно использовать команду `show ip eigrp neighbors detail`. В приведенном ниже примере показано, что эта команда выполняется на роутере OFF1. Обратите внимание, что выходные данные идентифицируют 10.1.1.2 как статически настроенного соседа.

```
OFF1#show ip eigrp neighbors detail
```

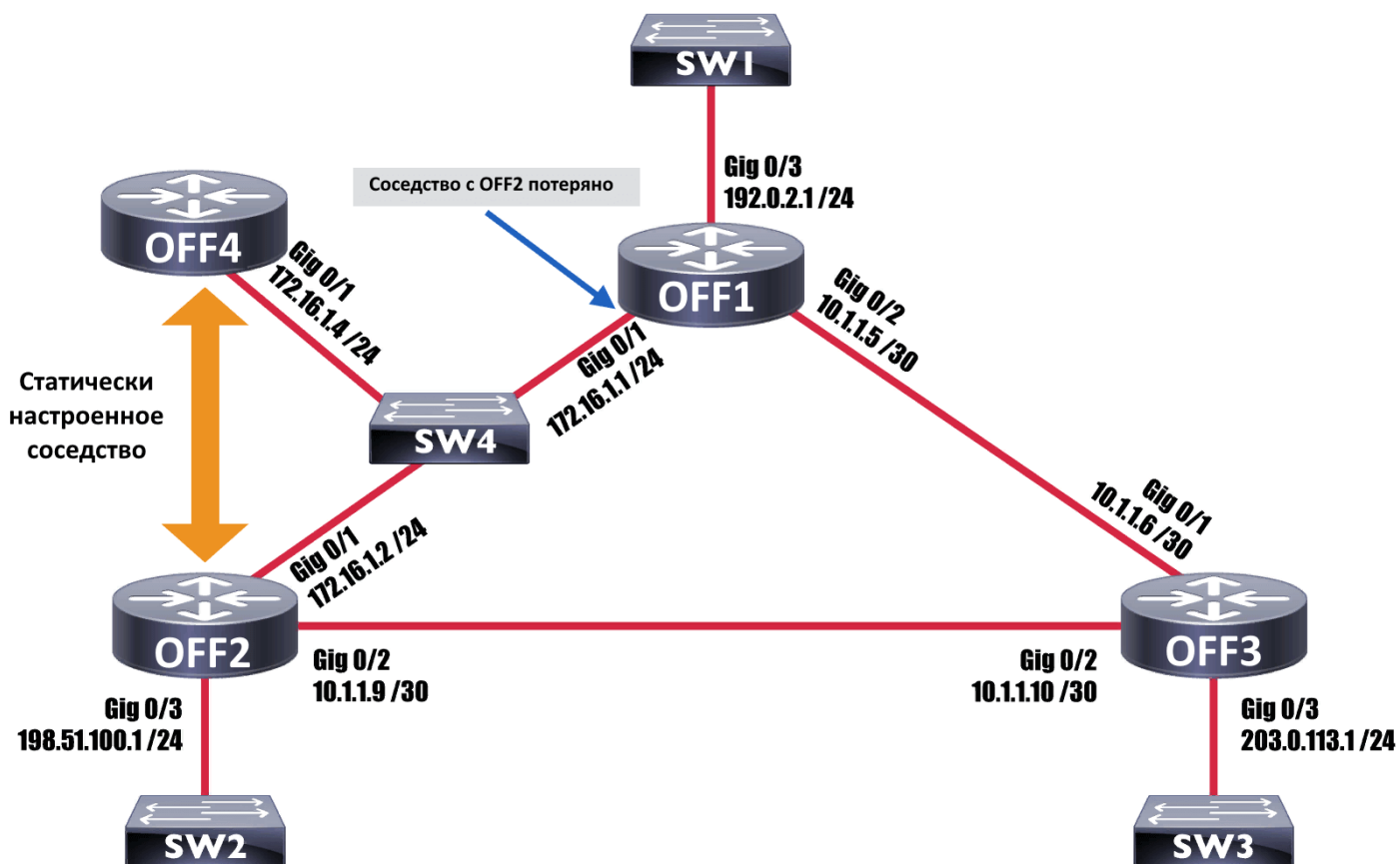
```
EIGRP-IPv4 Neighbors for AS(1)
H   Address                Interface          Hold Uptime    SRTT   RTO  Q  Seq
                               (sec)          (ms)                Cnt  Num
1   10.1.1.2                Gi0/1              12 00:18:18     7    100  0   12
  Static neighbor
  Version 20.0/2.0, Retrans: 0, Retries: 0, Prefixes: 3
  Topology-ids from peer - 0
  Topologies advertised to peer:   base

0   10.1.1.6                Gi0/2              12 00:20:45    87    522  0   8
  Version 20.0/2.0, Retrans: 1, Retries: 0, Prefixes: 3
  Topology-ids from peer - 0
  Topologies advertised to peer:   base

Max Nbrs: 0, Current Nbrs: 0
OFF1#
```

## ПРЕДОСТЕРЕЖЕНИЕ ПО ПРИМЕНЕНИЮ СТАТИЧЕСКОГО СОСЕДСТВА

Рассмотрим роутер, который должен установить более чем одно соседство EIGRP с одного интерфейса, например роутер OFF2 на рисунке ниже. В этой топологии роутеры OFF1 и OFF2 динамически сформировали соседство EIGRP. Позже был добавлен роутер OFF4, и роутеры OFF2 и OFF4 были настроены как соседи EIGRP статически. Однако после того, как была сделана статическая настройка, роутер OFF2 потерял свое соседство с роутером OFF1. Причина заключается в том, что роутер OFF2 отправляет только одноадресные сообщения EIGRP со своего интерфейса Gig0/1 и хочет получать только одноадресные сообщения EIGRP, поступающие на этот интерфейс. Однако роутер OFF1 все еще настроен (с настройками по умолчанию) для отправки и ожидания многоадресных сообщений EIGRP на своем интерфейсе Gig0/1. Итак, мораль этой истории заключается в том, что если вы настраиваете интерфейс роутера для установления соседства EIGRP статически, убедитесь, что все соседи EIGRP вне этого интерфейса также настроены для соседства статически.



## 8. EIGRP: идентификатор роутера и требования к соседству

Начнем мы наше обсуждение с рассмотрения идентификатора роутера EIGRP.

### EIGRP ROUTER ID

Каждый EIGRP-спикер роутер имеет ассоциируемый router ID EIGRP (RID). RID - это 32-битное значение, записанное в десятичном формате с точками, например IPv4-адрес. RID EIGRP определяется, когда процесс EIGRP начинает выполняться. Интересно, что EIGRP использует те же шаги для определения RID, что и OSPF. Ниже показаны последовательные шаги определения RID:

1. Шаг 1. Применить заданное значение RID.
2. Шаг 2. Если RID не настроен, используйте самый старший IPv4-адрес на loopback интерфейсе, находящийся в состоянии up/up.
3. Шаг 3. Если ни один loopback интерфейс не настроен с IPv4-адресом, используйте самый высокий IPv4-адрес на non-loopback интерфейсе.

Интересно, что в то время, как EIGRP требует, чтобы роутер имел RID, значение RID играет очень тривиальную роль в процессе EIGRP. Соседи EIGRP могут дублировать RID и устанавливать соседство EIGRP между ними, хотя лучше всего назначать уникальные RID соседям EIGRP. Однако, прежде чем мы чрезмерно минимизируем RID, есть один очень важный момент, когда роутер нуждается в уникальном RID роутера. В частности, если мы вводим внешние маршруты в процесс маршрутизации EIGRP, роутер, выполняющий это перераспределение, нуждается в уникальном RID.

### НАСТРОЙКА И ПРОВЕРКА ROUTER ID EIGRP

Чтобы сделать схему сетевой адресации более интуитивно понятной, вы можете выбрать ручную настройку RID EIGRP на определенном роутере. Это можно сделать с помощью команды EIGRP **router-id rid**, как показано на роутере OFF1 и показано в следующем примере:

```
OFF1#conf term
Enter configuration commands, one per line. End with CNTL/Z.
OFF1(config)#router eigrp 1
OFF1(config-router)#eigrp router-id 1.1.1.1
OFF1(config-router)#end
OFF1#
```

Обратите внимание на выходные данные в приведенном выше примере, что мы вручную установили RID роутера OFF1 на 1.1.1.1. Команды проверки, которые позволяют нам просматривать RID роутера, включают: `show ip eigrp topology` и `show ip protocols`, как показано в следующих примерах:

```
OFF1#show ip eigrp topology
EIGRP-IPv4 Topology Table for AS(1)/ID(1.1.1.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 192.0.2.0/24, 1 successors, FD is 2816
  via Connected, GigabitEthernet0/3
P 10.1.1.8/30, 2 successors, FD is 3072
  via 10.1.1.2 (3072/2816), GigabitEthernet0/1
  via 10.1.1.6 (3072/2816), GigabitEthernet0/2
P 203.0.113.0/24, 1 successors, FD is 3072
  via 10.1.1.6 (3072/2816), GigabitEthernet0/2
P 10.1.1.4/30, 1 successors, FD is 2816
  via Connected, GigabitEthernet0/2
P 198.51.100.0/24, 1 successors, FD is 3072
  via 10.1.1.2 (3072/2816), GigabitEthernet0/1
P 10.1.1.0/30, 1 successors, FD is 2816
  via Connected, GigabitEthernet0/1
OFF1#
```

```
OFF1#show ip eigrp topology
EIGRP-IPv4 Topology Table for AS(1)/ID(1.1.1.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 192.0.2.0/24, 1 successors, FD is 2816
  via Connected, GigabitEthernet0/3
P 10.1.1.8/30, 2 successors, FD is 3072
  via 10.1.1.2 (3072/2816), GigabitEthernet0/1
  via 10.1.1.6 (3072/2816), GigabitEthernet0/2
P 203.0.113.0/24, 1 successors, FD is 3072
  via 10.1.1.6 (3072/2816), GigabitEthernet0/2
P 10.1.1.4/30, 1 successors, FD is 2816
  via Connected, GigabitEthernet0/2
P 198.51.100.0/24, 1 successors, FD is 3072
  via 10.1.1.2 (3072/2816), GigabitEthernet0/1
P 10.1.1.0/30, 1 successors, FD is 2816
  via Connected, GigabitEthernet0/1
OFF1#
```

# ТРЕБОВАНИЯ К СОСЕДСТВУ

Одной из основных проблем, возникающих при устранении неполадок в сети EIGRP, является установление соседства. EIGRP имеет несколько требований, как и OSPF. Однако EIGRP и OSPF немного отличаются по своим предпосылкам соседства. В таблице ниже перечислены и противопоставлены правила установления соседства как для EIGRP, так и для OSPF.

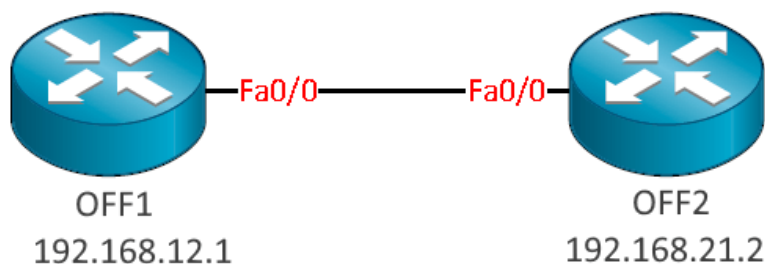
Требования	EIGRP	OSPF
иметь возможность отправлять пакеты на другой сервер	Да	Да
Первичный адрес интерфейса (не вторичный адрес) должен быть включен в ту же подсеть, что и сеть, сопоставляемая оператором network.	Да	Да
Интерфейс, соединенный с соседом не должен быть пассивным.	Да	Да
Необходимо использовать ту же автономную систему (для EIGRP) или process-ID (для OSPF) при настройке роутера.	Да	Нет
Таймер Hello и таймер Hold (для EIGRP) или Dead таймер (для OSPF) максимально совпадать.	Нет	Да
Соседи должны аутентифицироваться друг с другом, если аутентификация настроена.	Да	Да
Должно быть в той же зоне	N/A	Да
IP MTU совпадает.	Нет	Да
К-значения совпадают	Да	N/A
Идентификаторы роутеров (rid) должны быть уникальными	Нет	Да

## 9. Трешблшутинг EIGRP

В этой статье мы рассмотрим протокол маршрутизации Cisco EIGRP. EIGRP (Enhanced Interior Gateway Routing Protocol) - это протокол расширенной векторной маршрутизации, который должен устанавливать отношения соседства перед отправкой обновлений. Из-за этого первое, что нам нужно сделать, это проверить, правильно ли работает соседство. Если это так, мы можем продолжить, проверив, объявляются сети или нет. В этой статье рассмотрим все, что может пойти не так с EIGRP, как это исправить и в каком порядке. Давайте начнем с проверки соседства!

Существует ряд элементов, которые вызывают проблемы соседства EIGRP:

- **Неизвестная подсеть:** соседи EIGRP с IP-адресами, которые не находятся в одной подсети.
- **Несоответствие значений K:** по умолчанию пропускная способность и задержка включены для расчета метрики. Мы можем включить нагрузку и надежность, но мы должны сделать это на всех маршрутизаторах EIGRP.
- **Несоответствие AS:** номер автономной системы должен совпадать на обоих маршрутизаторах EIGRP, чтобы сформировать соседство.
- **Проблемы уровня 2:** EIGRP работает на уровне 3 модели OSI. Если уровни 1 и 2 не работают должным образом, у нас будут проблемы с формированием соседства.
- **Проблемы со списком доступа:** возможно, кто-то создал список доступа, который отфильтровывает многоадресный трафик. EIGRP по умолчанию использует 224.0.0.10 для связи с другими соседями EIGRP.
- **NBMA:** по умолчанию Non Broadcast Multi Access сети, такие как Frame Relay, не разрешают широковещательный или многоадресный трафик. Это может препятствовать тому, чтобы EIGRP формировал соседние отношения EIGRP.



```
OFF1(config)#int f0/0
OFF1(config-if)#ip address 192.168.12.1 255.255.255.0
OFF1(config-if)#router eigrp 12
OFF1(config-router)#network 192.168.12.0

OFF2(config)#int f0/0
OFF2(config-if)#ip address 192.168.21.2 255.255.255.0
OFF2(config)#router eigrp 12
OFF2(config-router)#network 192.168.21.0
```

Ошибка неверной подсети легко обнаружить. В приведенном выше примере у нас есть 2 маршрутизатора, и вы можете видеть, что были настроены разные подсети на каждом интерфейсе.

После включения EIGRP всплывают следующие ошибки:

```
OFF1# IP-EIGRP(Default-IP-Routing-Table:12): Neighbor 192.168.21.2 not on
common subnet for FastEthernet0/0
OFF2# IP-EIGRP(Default-IP-Routing-Table:12): Neighbor 192.168.12.1 not on
common subnet for FastEthernet0/0
```

Оба маршрутизатора жалуются, что находятся не в одной подсети.

```
OFF2(config-router)#int f0/0
OFF2(config-if)#ip address 192.168.12.2 255.255.255.0
OFF2(config)#router eigrp 12
OFF2(config-router)#no network 192.168.21.0
OFF2(config-router)#network 192.168.12.0
```

Мы изменили IP-адрес на OFF2 и убедились, что для **EIGRP** правильно настроена команда `network`.

```
OFF1# %DUAL-5-NBRCHANGE: IP-EIGRP(0) 12: Neighbor 192.168.12.2
(FastEthernet0/0) is up: new adjacency
OFF2# %DUAL-5-NBRCHANGE: IP-EIGRP(0) 12: Neighbor 192.168.12.1
(FastEthernet0/0) is up: new adjacency
```

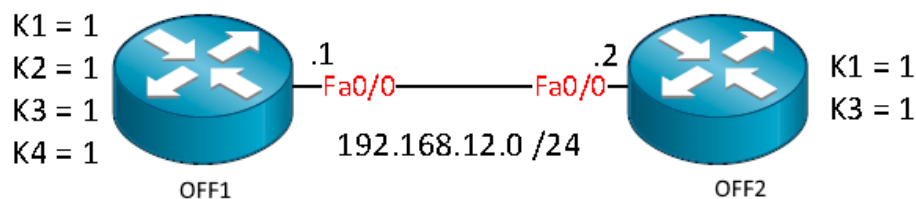
Вуаля! Теперь у нас есть соседство EIGRP.

```
OFF1#show ip eigrp neighbors
IP-EIGRP neighbors for process 12
H   Address                  Interface          Hold Uptime      SRTT    RTO    Q
Seq                                     (sec)            (ms)          Cnt
Num
0   192.168.12.2              Fa0/0              13 00:05:15      3       200    0    3
OFF2#show ip eigrp neighbors
IP-EIGRP neighbors for process 12
H   Address                  Interface          Hold Uptime      SRTT    RTO    Q
Seq                                     (sec)            (ms)          Cnt
Num
0   192.168.12.1              Fa0/0              11 00:05:32     1263    5000    0    3
```

Проверим это с помощью команды `show ip eigrp neighbors`.

**Извлеченный урок:** убедитесь, что оба маршрутизатора находятся в одной подсети.

## CASE #2



На этот раз IP-адреса верны, но мы используем разные значения К с обеих сторон. OFF1 включил пропускную способность, задержку, нагрузку и надежность. OFF2 использует только пропускную способность и задержку.

```
OFF1# %DUAL-5-NBRCHANGE: IP-EIGRP(0) 12: Neighbor 192.168.12.2
(FastEthernet0/0) is down: K-value mismatch

OFF1# %DUAL-5-NBRCHANGE: IP-EIGRP(0) 12: Neighbor 192.168.12.2
(FastEthernet0/0) is down: Interface Goodbye received
```

Эту ошибку легко обнаружить, поскольку сообщение в консоли гласит "**Несоответствие К-значений**" на обоих маршрутизаторах.

```
OFF1#show run | section eigrp
router eigrp 12
  network 192.168.12.0
  metric weights 0 1 1 1 1 0
  auto-summary
```



Мы можем проверить нашу конфигурацию, посмотрев ее на обоих маршрутизаторах. Как вы видите, что значения К были изменены на OFF1.

```
OFF2(config)#router eigrp 12
OFF2(config-router)#metric weights 0 1 1 1 1 0
```

Давайте убедимся, что значения К одинаковы на обоих маршрутизаторах, так как мы изменили их на OFF2.

```
OFF1# %DUAL-5-NBRCHANGE: IP-EIGRP(0) 12: Neighbor 192.168.12.2
(FastEthernet0/0) is up: new adjacency
```

```
OFF2# %DUAL-5-NBRCHANGE: IP-EIGRP(0) 12: Neighbor 192.168.12.1
(FastEthernet0/0) is up: new adjacency
```

После изменения значений К у нас появилось соседство EIGRP-соседей.

```
OFF1#show ip eigrp neighbors
IP-EIGRP neighbors for process 12
H      Address          Interface    Hold Uptime      SRTT  RTO   Q      Seq
sec          (ms)  Cnt    Num
0      192.168.12.2      Fa0/0       13 00:02:11      13    200   0      6
```

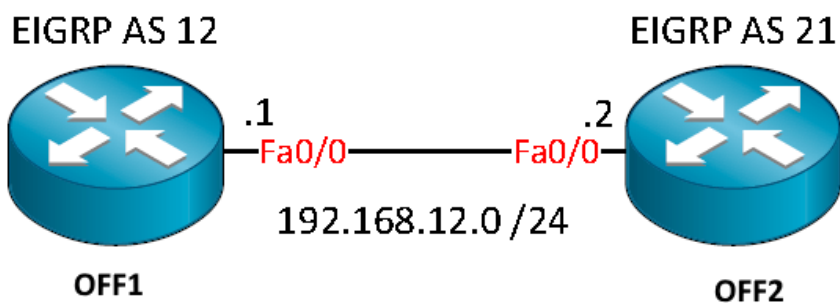
```
OFF2#show ip eigrp neighbors
IP-EIGRP neighbors for process 12
H      Address          Interface    Hold Uptime      SRTT  RTO   Q      Seq
sec          (ms)  Cnt    Num
0      192.168.12.1      Fa0/0 13    00:02:42      19    200   0      6
```

Еще одна проблема решена!

**Извлеченный урок:** убедитесь, что значения К одинаковы на всех маршрутизаторах EIGRP в одной и той же автономной системе.

## CASE #3

Давайте продолжим со следующей ошибкой ...



Вот еще один пример типичной проблемы. Несоответствие номера AS. Когда мы настраиваем EIGRP, мы должны ввести номер AS. В отличие от OSPF (который использует ID процесса) этот номер должен быть одинаковым на обоих маршрутизаторах.

```
OFF1#show ip eigrp neighbors
IP-EIGRP neighbors for process 12
```

```
OFF2#show ip eigrp neighbors
IP-EIGRP neighbors for process 21
```

В отличие от других неверных настроек конфигурации EIGRP, эта проблема не выдает сообщение об ошибке. Используем команду `show ip eigrp neighbors` и видим, что соседей нет. Внимательно изучите выходные данные, чтобы обнаружить различия, и вы увидите, что маршрутизаторы используют разные номера AS.

```

OFF1#show run | section eigrp
router eigrp 12
  network 192.168.12.0
  auto-summary

OFF2#show run | section eigrp
router eigrp 21
  network 192.168.12.0
  auto-summary

```

Если посмотреть на работающую конфигурацию, и мы увидим то же самое.

```

OFF2(config)#no router eigrp 21
router eigrp 12
  network 192.168.12.0
  metric weights 0 1 1 1 1 0
  auto-summary

```

Давайте изменим номер AS на OFF2.

```

OFF1# %DUAL-5-NBRCHANGE: IP-EIGRP(0) 12: Neighbor 192.168.12.2
(FastEthernet0/0) is up: new adjacency

OFF2# %DUAL-5-NBRCHANGE: IP-EIGRP(0) 12: Neighbor 192.168.12.1
(FastEthernet0/0) is up: new adjacency

```

После смены номера AS все заработало как положено.

```

OFF1#show ip eigrp neighbors
IP-EIGRP neighbors for process 12
H      Address           Interface           Hold Uptime  SRTT  RTO   Q      Seq
      (sec)             (ms)              Cnt  Num
0      192.168.12.2       Fa0/0 11           00:01:44    13    200    0      3

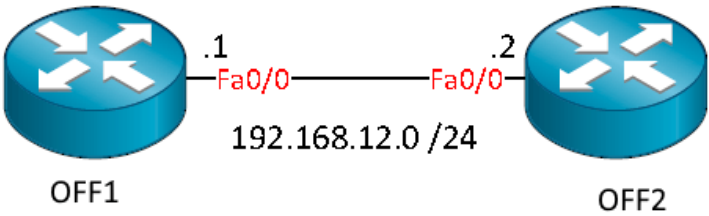
OFF2#show ip eigrp neighbors
IP-EIGRP neighbors for process 12
H      Address           Interface           Hold Uptime  SRTT  RTO   Q      Seq
      (sec)             (ms)              Cnt  Num
0      192.168.12.1       Fa0/0 12           00:01:53     7    200    0      9

```

**Извлеченный урок:** убедитесь, что номера AS одинаковые, если вы хотите соседства EIGRP.

## CASE #4

И последнее, но не менее важное: если вы проверили номер AS, значения K, IP-адреса и у вас все еще нет работающего соседства EIGRP, вам следует подумать о безопасности. Возможно, **access-list** блокирует EIGRP и/или многоадресный трафик.



**Следующая ситуация:** опять два маршрутизатора EIGRP и отсутствие соседства. Что здесь происходит?

```

OFF1#show ip eigrp neighbors
IP-EIGRP neighbors for process 12

OFF2#show ip eigrp neighbors
IP-EIGRP neighbors for process 12

```

Мы видим, что нет соседей ...

```
OFF1#show ip protocols
Routing Protocol is "eigrp 12"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 12
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    192.168.12.0
  Routing Information Sources:
    Gateway         Distance         Last Update
  Distance: internal 90 external 170

OFF2#show ip protocols
Routing Protocol is "eigrp 12"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 12
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    192.168.12.0
  Passive Interface(s):
    FastEthernet0/0
  Routing Information Sources:
    Gateway         Distance         Last Update
  Distance: internal 90 external 170
```

Если вы посмотрите на вывод команды **show ip protocols**, то увидите, что сеть была объявлена правильно. Если вы посмотрите внимательно на OFF2, вы увидите, что у нас есть пассивный интерфейс.

Удалим настройки пассивного интерфейса!

```
OFF2(config)#router eigrp 12
OFF2(config-router)#no passive-interface fastEthernet 0/0
```

Еще одна неправильная настройка создала нам проблемы, но мы ее решили.

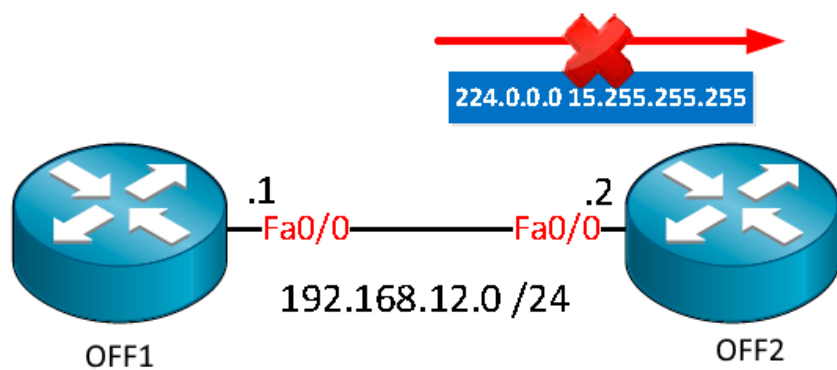
```
OFF1#show ip eigrp neighbors
IP-EIGRP neighbors for process 12
H Address          Interface          Hold Uptime        SRTT  RTO  Q      Seq
  (sec)            (ms)              Cnt  Num
0 192.168.12.2      Fa0/0 13          00:05:23          24   200   0      6

OFF2#show ip eigrp neighbors
IP-EIGRP neighbors for process 12
H Address          Interface          Hold Uptime        SRTT  RTO  Q      Seq
  (sec)            (ms)              Cnt  Num
0 192.168.12.1      Fa0/0 14          00:05:39          20   200   0      6
```

Задача решена!

**Извлеченный урок:** не включайте пассивный интерфейс, если вы хотите установить соседство EIGRP.

## CASE #5



В приведенном выше примере у нас есть те же 2 маршрутизатора, но на этот раз кто-то решил, что было бы неплохо настроить список доступа на OFF2, который блокирует весь входящий многоадресный трафик.

```
OFF1# %DUAL-5-NBRCHANGE: IP-EIGRP(0) 12: Neighbor 192.168.12.2
(FastEthernet0/0) is down: retry limit exceeded
%DUAL-5-NBRCHANGE: IP-EIGRP(0) 12: Neighbor 192.168.12.2 (FastEthernet0/0)
is up: new adjacency
%DUAL-5-NBRCHANGE: IP-EIGRP(0) 12: Neighbor 192.168.12.2 (FastEthernet0/0)
is down: retry limit exceeded
%DUAL-5-NBRCHANGE: IP-EIGRP(0) 12: Neighbor 192.168.12.2 (FastEthernet0/0)
is up: new adjacency
```

Здесь можно запутаться. На OFF1 мы видим, что он считает, что установил соседство EIGRP с OFF2. Это происходит потому, что мы все еще получаем пакеты EIGRP от OFF2.

```
OFF1#debug eigrp neighbors
EIGRP Neighbors debugging is on
EIGRP: Retransmission retry limit exceeded
EIGRP: Holdtime expired
```

Используем команду `debug eigrp neighbors`, чтобы посмотреть, что происходит. Очевидно, что OFF1 не получает ответ от своих **hello messages**, **holdtime** истекает, и это отбрасывает установление соседства EIGRP.

```
OFF1#ping 224.0.0.10
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.0.0.10, timeout is 2 seconds:
```

Быстрый способ проверить подключение - отправить эхо-запрос по адресу многоадресной рассылки 224.0.0.10, который использует EIGRP. Мы видим, что мы ответа нет от этого запроса. Рекомендуется проверить, есть ли в сети списки доступа.

```
OFF2#show ip interface fa0/0 | include access list
Outgoing access list is not set
Inbound access list is BLOCKMULTICAST
```

Так, так! Мы нашли что-то ...

```
OFF2#show ip access-lists
Extended IP access list BLOCKMULTICAST
10 deny ip any 224.0.0.0 15.255.255.255 (536 matches)
20 permit ip any any (468 matches)
```

Этот список доступа блокирует весь многоадресный трафик. Давайте сделаем настройку, которая разрешит EIGRP.

```
OFF2(config)#ip access-list extended BLOCKMULTICAST
OFF2(config-ext-nacl)#5 permit ip any host 224.0.0.10
```

Мы создаем специальное правило, которое будет разрешать трафик EIGRP.

```

OFF2#show access-lists
Extended IP access list BLOCKMULTICAST
 5 permit ip any host 224.0.0.10 (27 matches)
10 deny ip any 224.0.0.0 15.255.255.255 (569 matches)
20 permit ip any any (501 matches)

```

Как мы видим, что трафик EIGRP разрешен - это соответствует правилу, которое мы выше создали.

```

OFF1# %DUAL-5-NBRCHANGE: IP-EIGRP(0) 12: Neighbor 192.168.12.2
(FastEthernet0/0) is up: new adjacency
OFF2# %DUAL-5-NBRCHANGE: IP-EIGRP(0) 12: Neighbor 192.168.12.1
(FastEthernet0/0) is up: new adjacency

```

Оба маршрутизатора теперь показывают рабочее соседство EIGRP.

```

OFF1#ping 224.0.0.10

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.0.0.10, timeout is 2 seconds:

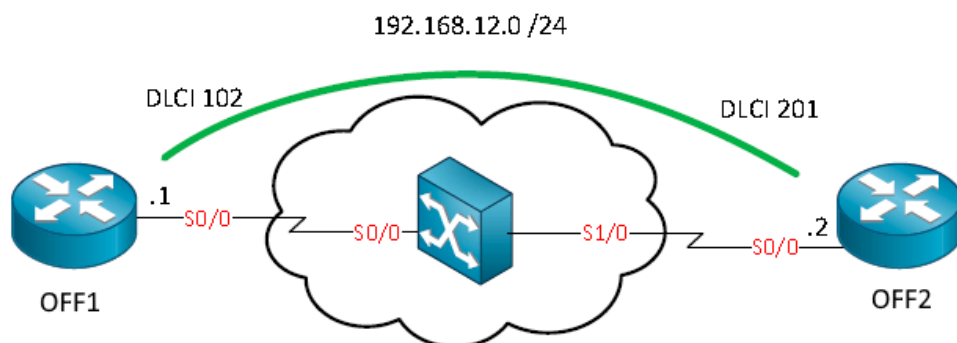
Reply to request 0 from 192.168.12.2, 24 ms

```

Эхо-запрос, который мы только что отправили, теперь работает.

**Извлеченный урок:** не блокируйте пакеты EIGRP!

## CASE #6



Рассмотрим очередную ситуацию, в которой нет соседства EIGRP. На картинке выше мы имеем сеть Frame Relay и один канал **PVC** между OFF1 и OFF2. Вот соответствующая конфигурация:

<pre> OFF1# interface Serial0/0  ip address 192.168.12.1 255.255.255.0  encapsulation frame-relay  serial restart-delay 0  frame-relay map ip 192.168.12.2 102  no frame-relay inverse-arp router eigrp 12  network 192.168.12.0  auto-summary </pre>	<pre> OFF2# interface Serial0/0  ip address 192.168.12.2 255.255.255.0  encapsulation frame-relay  serial restart-delay 0  frame-relay map ip 192.168.12.1 201  no frame-relay inverse-arp router eigrp 12  network 192.168.12.0  auto-summary </pre>
---	---

Оба маршрутизатора настроены для **Frame Relay**, а EIGRP настроен.

<pre> OFF1#show ip eigrp neighbors IP-EIGRP neighbors for process 12 </pre>	<pre> OFF2#show ip eigrp neighbors IP-EIGRP neighbors for process 12 </pre>
---	---

Видно, что нет соседей ... это не хорошо! Можем ли мы пропинговать другую сторону?

```
OFF1#ping 192.168.12.2
```

```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 192.168.12.2, timeout is 2 seconds:  
!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/8/24 ms
```

Пинг проходит, поэтому мы можем предположить, что PVC Frame Relay работает. EIGRP, однако, использует многоадресную передачу, а Frame Relay по умолчанию - **NBMA**. Можем ли мы пропинговать адрес многоадресной рассылки EIGRP 224.0.0.10?

```
OFF1#ping 224.0.0.10  
Type escape sequence to abort.  
Sending 1, 100-byte ICMP Echos to 224.0.0.10, timeout is 2 seconds:
```

Здесь нет ответа на наш вопрос, по крайней мере, теперь мы знаем, что unicast трафик работает, а multicast не работает. Frame Relay может быть настроен для point-to-point или point-to-multipoint соединения. Физический интерфейс всегда является интерфейсом **frame-relay point-to-multipoint**, и для него требуются frame-relay maps, давайте проверим это:

```
OFF1#show frame-relay map  
Serial0/0 (up): ip 192.168.12.2 dlci 102(0x66,0x1860), static,  
CISCO, status defined, active  
OFF2#show frame-relay map  
Serial0/0 (up): ip 192.168.12.1 dlci 201(0xC9,0x3090), static,  
CISCO, status defined, active
```

Мы видим, что оба маршрутизатора имеют DLCI-to-IP карты, поэтому они знают, как связаться друг с другом. Видим, что они используют ключевое слово "статический", а это говорит о том, что это сопоставление было кем-то настроено и не изучено с помощью Inverse ARP (в противном случае вы увидите "динамический"). Мы не видим ключевое слово "broadcast", которое требуется для пересылки широковещательного или многоадресного трафика. На данный момент у нас есть 2 варианта решения этой проблемы:

- Настроить EIGRP для использования одноадресного трафика вместо многоадресного.
- Проверить конфигурацию Frame Relay и убедиться, что многоадресный трафик не перенаправляется.

Давайте сначала сделаем unicast настройку EIGRP:

```
OFF1(config)#router eigrp 12  
OFF1(config-router)#neighbor 192.168.12.2 serial 0/0  
OFF2(config)#router eigrp 12  
OFF2(config-router)#neighbor 192.168.12.1 serial 0/0
```

Нам нужна команда **neighbor** для конфигурации EIGRP. Как только вы введете эту команду и нажмете enter, вы увидите это:

```
OFF1#  
%DUAL-5-NBRCHANGE: IP-EIGRP(0) 12: Neighbor 192.168.12.2 (Serial0/0) is up:  
new adjacency  
OFF2#  
%DUAL-5-NBRCHANGE: IP-EIGRP(0) 12: Neighbor 192.168.12.1 (Serial0/0) is up:  
new adjacency
```

Задача решена! Теперь давайте попробуем другое решение, где мы отправляем multicast трафик по PVC Frame Relay:

```
OFF1(config)#router eigrp 12  
OFF1(config-router)#no neighbor 192.168.12.2 serial 0/0  
OFF2(config)#router eigrp 12  
OFF2(config-router)#no neighbor 192.168.12.1 serial 0/0
```

Если это не работает ... не исправляйте это... , но не в этот раз! Пришло время сбросить соседство EIGRP.

```
OFF1(config)#interface serial 0/0  
OFF1(config-if)#frame-relay map ip 192.168.12.2 102 broadcast  
OFF2(config)#interface serial 0/0  
OFF2(config-if)#frame-relay map ip 192.168.12.1 201 broadcast
```

**Broadcast** - это ключевое волшебное слово здесь. Это разрешит широковещательный и многоадресный трафик.



```
OFF1#  
%DUAL-5-NBRCHANGE: IP-EIGRP(0) 12: Neighbor 192.168.12.2 (Serial0/0) is up:  
new adjacency  
  
OFF2#  
%DUAL-5-NBRCHANGE: IP-EIGRP(0) 12: Neighbor 192.168.12.1 (Serial0/0) is up:  
new adjacency
```

После изменения конфигурации frame-relay map появляется соседство EIGRP! Это все, что нужно сделать.

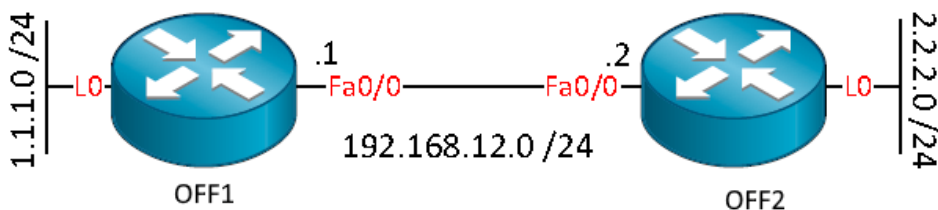
**Извлеченный урок:** проверьте, поддерживает ли ваша сеть Frame Relay broadcast или нет. Настройте EIGRP для использования unicast передачи или измените конфигурацию Frame Relay для поддержки широковещательного трафика.

## 10. Поиск и устранение неисправностей протокола EIGRP

Теперь мы можем продолжить поиск и устранение неисправностей. В большинстве случаев вы ожидаете увидеть определенную сеть в таблице маршрутизации, но ее там нет. Далее рассмотрим несколько сценариев неправильной (или полностью не рабочей) работы EIGRP и как исправить наиболее распространенные ошибки. Ниже перечислены часто встречающиеся ошибки:

- Кто-то настроил **distribute-list**, чтобы информация о маршрутах фильтровалась.
- Было настроено автосуммирование или кто-то настроил суммирование вручную
- **Split-horizon** блокирует объявление маршрутной информации.
- Перераспределение было настроено, но информация из EIGRP не используется.
- Перераспределение было настроено, но никакие внешние маршруты EIGRP не отображаются.

### CASE #1



Давайте начнем с простой топологии. OFF1 и OFF2 работают под управлением EIGRP, и каждый маршрутизатор имеет интерфейс обратной связи. Вот конфигурация обоих маршрутизаторов:

```
OFF1(config)#router eigrp 12
OFF1(config-router)#no auto-summary
OFF1(config-router)#network 1.1.1.0 0.0.0.255
OFF1(config-router)#network 192.168.12.0 0.0.0.255
OFF2(config)#router eigrp 12
OFF2(config-router)#no auto-summary
OFF2(config-router)#network 2.2.2.0 0.0.0.255
OFF2(config-router)#network 192.168.12.0 0.0.0.255
```

Все работает нормально, пока через пару недель один из пользователей не пожаловался на то, что ему не удалось подключиться к сети 2.2.2.0 / 24 из-за OFF1. Посмотрите на таблицу маршрутизации на OFF1, и вот что вы видите:

```
OFF1#show ip route

Gateway of last resort is not set

C    192.168.12.0/24 is directly connected, FastEthernet0/0
  1.0.0.0/24 is subnetted, 1 subnets
C    1.1.1.0 is directly connected, Loopback0
```

По какой-то причине нет сети 2.2.2.0 / 24 в таблице маршрутизации.

```
OFF1#show ip protocols | include filter
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
```

Видно, что на OFF1 не настроен **distribute lists**.

```
OFF2#show ip route
```

```
Gateway of last resort is not set
```

```
C    192.168.12.0/24 is directly connected, FastEthernet0/0
    1.0.0.0/24 is subnetted, 1 subnets
D    1.1.1.0 [90/156160] via 192.168.12.1, 00:14:01, FastEthernet0/0
    2.0.0.0/24 is subnetted, 1 subnets
C    2.2.2.0 is directly connected, Loopback0
```

OFF2 содержит сеть 1.1.1.0 / 24 в своей таблице маршрутизации. Давайте выполним быструю отладку, чтобы увидеть, что происходит.

```
OFF2#debug ip eigrp
```

```
IP-EIGRP Route Events debugging is on
```

```
IP-EIGRP(Default-IP-Routing-Table:12): 2.2.2.0/24 - denied by distribute
list
```

Отладка показывает нам, что происходит. Прежде чем вы увидите это сообщение, придется немного подождать, или вы можете сбросить соседство EIGRP, чтобы ускорить процесс. Как видите, в сети 2.2.2.0 / 24 отказано из-за distribute list.

```
OFF2#show ip protocols | include filter
```

```
Outgoing update filter list for all interfaces is 1
Incoming update filter list for all interfaces is not set
```

Другой быстрый способ проверить это - использовать команду `show ip protocol`.

```
OFF2#show run | section eigrp
```

```
router eigrp 12
  network 2.2.2.0 0.0.0.255
  network 192.168.12.0
  distribute-list 1 out
  no auto-summary
```

В этом случае использование `show run` могло бы быстрее обнаружить **distribute-list**.

```
OFF2#show access-lists
```

```
Standard IP access list 1
 10 deny 2.2.2.0, wildcard bits 0.0.0.255 (5 matches)
 20 permit any (5 matches)
```

Вот список доступа, доставляющий нам неприятности.

```
OFF2(config)#router eigrp 12
OFF2(config-router)#no distribute-list 1 out
```

Удалим distribute-list.

```
OFF1#show ip route
```

```
Gateway of last resort is not set
```

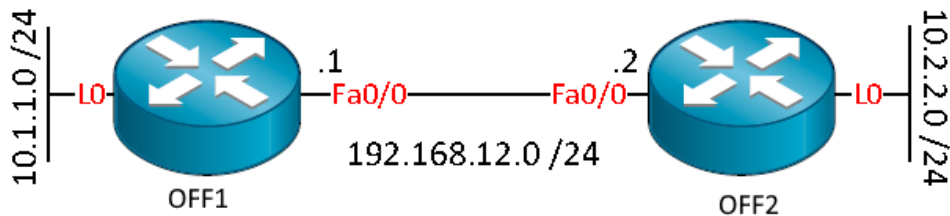
```
C    192.168.12.0/24 is directly connected, FastEthernet0/0
    1.0.0.0/24 is subnetted, 1 subnets
C    1.1.1.0 is directly connected, Loopback0
    2.0.0.0/24 is subnetted, 1 subnets
D    2.2.2.0 [90/156160] via 192.168.12.2, 00:00:13, FastEthernet0/0
```

Задача решена!

**Извлеченный урок:** если команды **network** верны, проверьте, есть ли у вас **distribute-list**, который запрещает объявлять префиксы или устанавливать их в таблицу маршрутизации.

Имейте в виду, distribute-list могут быть настроены как входящие или исходящие, как список доступа.

## CASE #2



В следующем сценарии те же 2 маршрутизатора, но разные сети в **loopback**. Вот конфигурация:

```
OFF1(config)#router eigrp 12
OFF1(config-router)#network 192.168.12.0
OFF1(config-router)#network 10.0.0.0
OFF2(config)#router eigrp 12
OFF2(config-router)#network 192.168.12.0
OFF2(config-router)#network 10.0.0.0
```

Как вы видите - это довольно базовая конфигурация.

```
OFF1#show ip route

Gateway of last resort is not set

C    192.168.12.0/24 is directly connected, FastEthernet0/0
    10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C    10.1.1.0/24 is directly connected, Loopback0
D    10.0.0.0/8 is a summary, 00:00:07, Null0

OFF2#show ip route

Gateway of last resort is not set
C    192.168.12.0/24 is directly connected, FastEthernet0/0
    10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C    10.2.2.0/24 is directly connected, Loopback0
D    10.0.0.0/8 is a summary, 00:00:15, Null0
```

Глядя на таблицы маршрутизации, не видно сети 10.1.1.0 / 24 или 10.2.2.0 / 24. Видна запись для сети 10.0.0.0/8, указывающую на интерфейс **null0**. Эта запись отображается только при настройке суммирования и используется для предотвращения циклов маршрутизации.

```
OFF2#debug ip eigrp
IP-EIGRP Route Events debugging is on
```

Давайте включим отладку и посмотрим, что мы можем найти.

```
OFF2#clear ip eigrp 12 neighbors
```

Этой командой мы сделаем сброс соседства EIGRP, чтобы ускорить процесс. Имейте в виду, что это, вероятно, не самое лучшее, что можно сделать в производственной сети, пока вы не узнаете, что не так, но это действительно помогает ускорить процесс.

```
OFF2#
IP-EIGRP(Default-IP-Routing-Table:12): 192.168.12.0/24 - do advertise out
FastEthernet0/0
IP-EIGRP(Default-IP-Routing-Table:12): 10.2.2.0/24 - don't advertise out
FastEthernet0/0
IP-EIGRP(Default-IP-Routing-Table:12): 10.0.0.0/8 - do advertise out
FastEthernet0/0
```

Вот наш ответ. Отладка говорит нам, что сеть 10.2.2.0 / 24 не следует объявлять, а сеть 10.0.0.0 / 8 нужно объявлять (это вкратце). Это может произойти по двум причинам:

- Суммирование было кем-то настроено

- Авто-суммирование включено для EIGRP.

```

OFF1#show run | section eigrp
router eigrp 12
  network 10.0.0.0
  network 192.168.12.0
  auto-summary
OFF2#show run | section eigrp
router eigrp 12
  network 10.0.0.0
  network 192.168.12.0
  auto-summary

```

Как вы видите, авто-суммирование включено для EIGRP. В зависимости от версии IOS авто-суммирование включено или отключено по умолчанию.

```

OFF1(config)#router eigrp 12
OFF1(config-router)#no auto-summary
OFF2(config)#router eigrp 12
OFF2(config-router)#no auto-summary

```

Отключение автоматического суммирования должно помочь.

```

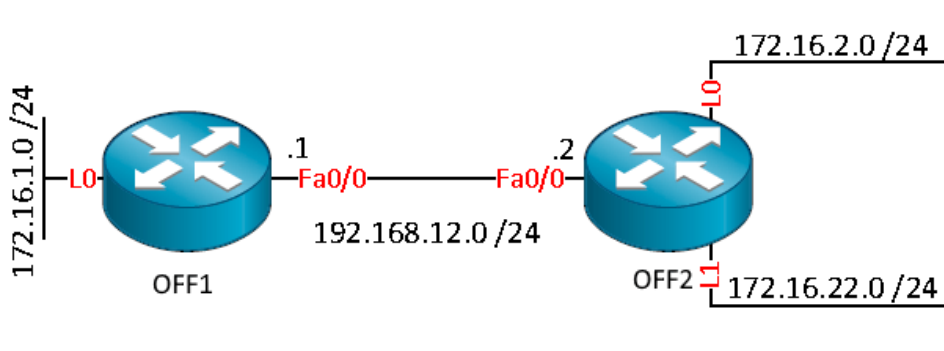
OFF1#show ip route
C    192.168.12.0/24 is directly connected, FastEthernet0/0
C    10.0.0.0/24 is subnetted, 2 subnets
D    10.2.2.0 [90/156160] via 192.168.12.2, 00:00:22, FastEthernet0/0
C    10.1.1.0 is directly connected, Loopback0
OFF2#show ip route
C    192.168.12.0/24 is directly connected, FastEthernet0/0
C    10.0.0.0/24 is subnetted, 2 subnets
C    10.2.2.0 is directly connected, Loopback0
D    10.1.1.0 [90/156160] via 192.168.12.1, 00:16:24, FastEthernet0/0

```

Ну что, наши сети появились в таблице маршрутизации.

**Извлеченный урок:** если включена автоматическое суммирование EIGRP, вы можете столкнуться с нестабильными сетями.

## CASE #3



Очередная проблема. В приведенном выше примере у нас есть 2 маршрутизатора, но разные сети. OFF1 содержит сеть 172.16.1.0 / 24 на интерфейсе обратной связи, а OFF2 содержит сеть 172.16.2.0 / 24 и 172.16.22.0 / 24 на своих интерфейсах обратной связи. Посмотрим конфигурацию EIGRP обоих маршрутизаторов:

```

OFF1#
router eigrp 12
  network 172.16.1.0 0.0.0.255
  network 192.168.12.0
  auto-summary
OFF2#
router eigrp 12
  network 172.16.2.0 0.0.0.255
  network 172.16.22.0 0.0.0.255
  network 192.168.12.0
  no auto-summary

```

Как вы видите, что все сети объявляются. Обратите внимание, что в OFF1 включено автоматическое суммирование, а в OFF2 отключено автоматическое суммирование.

```

OFF2#
interface FastEthernet0/0
ip summary-address eigrp 12 172.16.0.0 255.255.0.0 5

```

Кто-то настроил суммирование на OFF2 и отправляет ее на OFF1. Суммирование создана для сети 172.16.0.0 / 16.

```
OFF1#show ip route

C      192.168.12.0/24 is directly connected, FastEthernet0/0
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
D      172.16.0.0/16 is a summary, 00:30:07, Null0
C      172.16.1.0/24 is directly connected, Loopback0
```

Однако, если посмотреть на таблицу маршрутизации OFF1, она не появится. Мы видим запись для сети 172.16.0.0 / 16, но она указывает на интерфейс null0, а не на OFF2. Что здесь происходит?

```
OFF2#debug ip eigrp
IP-EIGRP Route Events debugging is on
```

OFF2#clear ip eigrp 12 neighbors

Давайте сделаем отладку на OFF2, чтобы увидеть, объявляется ли суммирование. Выполним команду `clear ip eigrp neighbors`, просто чтобы ускорить процесс.

```
OFF2#
IP-EIGRP(Default-IP-Routing-Table:12): 192.168.12.0/24 - do advertise out
FastEthernet0/0
IP-EIGRP(Default-IP-Routing-Table:12): 172.16.2.0/24 - don't advertise out
FastEthernet0/0
IP-EIGRP(Default-IP-Routing-Table:12): 172.16.22.0/24 - don't advertise out
FastEthernet0/0
IP-EIGRP(Default-IP-Routing-Table:12): 172.16.0.0/16 - do advertise out
FastEthernet0/0
```

Глядя на отладку, видно, что OFF2 работает правильно. Он объявляет сводный маршрут 172.16.0.0 / 16 так, как должен. Это означает, что проблема должна быть в OFF1.

```
OFF1#debug ip eigrp
IP-EIGRP Route Events debugging is on
```

Давайте проведем отладку OFF1.

```
OFF1#
IP-EIGRP(Default-IP-Routing-Table:12): Processing incoming UPDATE packet
IP-EIGRP(Default-IP-Routing-Table:12): Int 172.16.0.0/16 M 156160 - 25600
130560 SM 128256 - 256 128000
IP-EIGRP(Default-IP-Routing-Table:12): route installed for 172.16.0.0
(Summary)
IP-EIGRP(Default-IP-Routing-Table:12): 172.16.0.0/16 routing table not
updated thru 192.168.12.2
```

Мы можем видеть, что OFF1 получает сводный маршрут от OFF2, но решает не использовать его.



```

OFF1#show ip eigrp topology 172.16.0.0
IP-EIGRP (AS 12): Topology entry for 172.16.0.0/16
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 128256
  Routing Descriptor Blocks:
    0.0.0.0 (Null0), from 0.0.0.0, Send flag is 0x0
      Composite metric is (128256/0), Route is Internal
      Vector metric:
        Minimum bandwidth is 10000000 Kbit
        Total delay is 5000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1514
        Hop count is 0
    192.168.12.2 (FastEthernet0/0), from 192.168.12.2, Send flag is 0x0
      Composite metric is (156160/128256), Route is Internal
      Vector metric:
        Minimum bandwidth is 100000 Kbit
        Total delay is 5100 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1

```

Это хороший момент для проверки таблицы топологии EIGRP. Вы видите, что он имеет суммирование сети 172.16.0.0 / 16 от OFF2 в своей таблице топологии EIGRP, но OFF1 решает не использовать ее, потому что вход через интерфейс null0 является **лучшим** путем.

```

OFF1(config)#router eigrp 12
OFF1(config-router)#no auto-summary

```

Решение состоит в том, что нам нужно избавиться от записи **null0** в таблице маршрутизации. Единственный способ сделать это - отключить автоматическое суммирование.

```

OFF1#show ip route

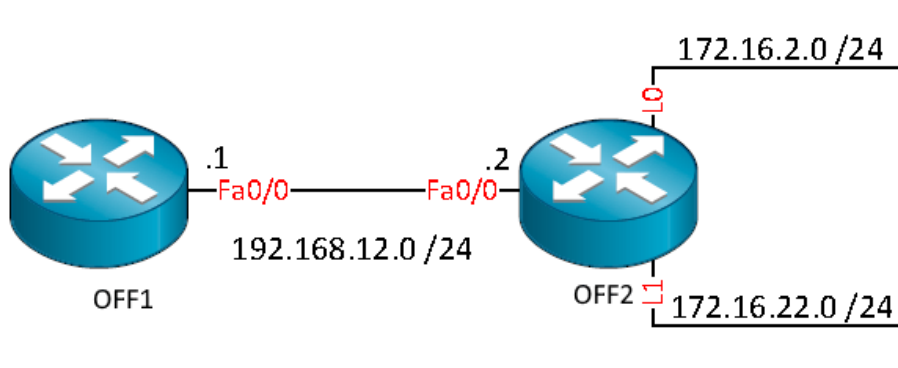
C       192.168.12.0/24 is directly connected, FastEthernet0/0
172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
D       172.16.0.0/16 [90/156160] via 192.168.12.2, 00:00:51, FastEthernet0/0
C       172.16.1.0/24 is directly connected, Loopback0

```

Отключение автоматического суммирования удаляет запись null0, и теперь суммирование OFF2 установлено проблема решена!

**Извлеченный урок:** автоматическое суммирование EIGRP создает запись через интерфейс null0, которая может помешать установке суммирования, которые вы получаете от соседних маршрутизаторов.

## CASE #4



Есть еще одна проблема с суммированием, которую сейчас и разберем. Мы используем топологию, которую вы видите выше, и ниже конфигурация EIGRP обоих маршрутизаторов.

```

OFF1#
router eigrp 12
  network 192.168.12.0
  no auto-summary

OFF2#
router eigrp 12
  network 172.16.2.0 0.0.0.255
  network 172.16.22.0 0.0.0.255
  network 192.168.12.0
  no auto-summary

```

Все сети объявлены, и автоматическое суммирование отключено на обоих маршрутизаторах.

```
OFF2#  
interface FastEthernet0/0  
ip summary-address eigrp 12 172.16.0.0 255.255.0.0 5
```

Суммирование было настроено на OFF2 и должно быть объявлено к OFF1.

```
OFF1#show ip route  
  
C    192.168.12.0/24 is directly connected, FastEthernet0/0
```

К сожалению, ничего не видно на OFF1. Давайте проверим OFF2, чтобы посмотреть, что не так.

```
OFF2#debug ip eigrp  
IP-EIGRP Route Events debugging is on  
  
OFF2#clear ip eigrp 12 neighbors
```

Когда дело доходит до устранения неполадок с сетью, вашими друзьями являются не Google или Яндекс, а команды `Debug` и `show`.

```
OFF2#  
IP-EIGRP (Default-IP-Routing-Table:12): 192.168.12.0/24 - do advertise out  
FastEthernet0/0
```

Странно, это единственная сеть, которую OFF2 объявляет.

```
OFF2#show ip route  
  
C    192.168.12.0/24 is directly connected, FastEthernet0/0
```

Одно из золотых правил маршрутизации: вы не можете объявлять то, чего у вас нет. Очевидно, OFF2 знает только о сети 192.168.12.0 / 24.

```
OFF2#show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	192.168.12.2	YES	manual	up	up
Loopback0	172.16.2.2	YES	manual	administratively down	down
Loopback1	172.16.22.22	YES	manual	administratively down	down

Вот это ошибка! Кто-то выполнил команду отключения на интерфейсах обратной связи.

```
OFF2(config)#interface loopback 0  
OFF2(config-if)#no shutdown  
OFF2(config)#interface loopback 1  
OFF2(config-if)#no shutdown
```

Включим интерфейсы.

```
OFF2#  
IP-EIGRP (Default-IP-Routing-Table:12): 172.16.2.0/24 - don't advertise out  
FastEthernet0/0  
IP-EIGRP (Default-IP-Routing-Table:12): 172.16.22.0/24 - don't advertise out  
FastEthernet0/0  
IP-EIGRP (Default-IP-Routing-Table:12): 172.16.0.0/16 - do advertise out  
FastEthernet0/0
```

Теперь мы видим, что суммирование объявляется.

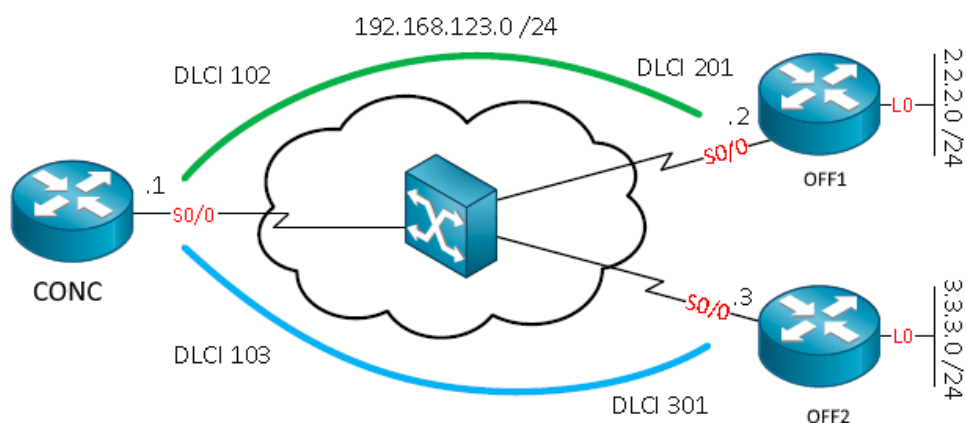
```
OFF1#show ip route | include 172.16.0.0
D    172.16.0.0/16 [90/156160] via 192.168.12.2, 00:09:39, FastEthernet0/0
```

Теперь мы видим суммирование в таблице маршрутизации OFF1- проблема решена!

**Извлеченный урок:** вы не можете объявлять то, чего у вас нет в таблице маршрутизации.

**ВАЖНО.** Последняя проблема может быть показаться простой, но есть важный момент, который вы не должны забывать: для объявления итогового маршрута в таблице маршрутизации объявляемого маршрутизатора должен быть указан хотя бы один префикс, попадающий в итоговый диапазон!

## CASE #5



Давайте посмотрим на другую топологию. На рисунке выше у нас есть концентратор **Frame Relay** и соответствующая топология. Каждый из OFF1 и OFF2 имеет интерфейс обратной связи, который мы будем объявлять в EIGRP. Вот соответствующая конфигурация всех маршрутизаторов:

```
CONC(config)#router eigrp 123
CONC(config-router)#no auto-summary
CONC(config-router)#network 192.168.123.0
OFF1(config-if)#router eigrp 123
OFF1(config-router)#no auto-summary
OFF1(config-router)#network 192.168.123.0
OFF1(config-router)#network 2.2.2.0 0.0.0.255
OFF2(config)#router eigrp 123
OFF2(config-router)#no auto-summary
OFF2(config-router)#network 192.168.123.0
OFF2(config-router)#network 3.3.3.0 0.0.0.255
```

Видно, что все сети объявлены.

```
CONC#show ip route
C    192.168.123.0/24 is directly connected, Serial0/0
C    2.0.0.0/24 is subnetted, 1 subnets
D    2.2.2.0 [90/2297856] via 192.168.123.2, 00:00:11, Serial0/0
C    3.0.0.0/24 is subnetted, 1 subnets
D    3.3.3.0 [90/2297856] via 192.168.123.3, 00:00:03, Serial0/0
```

Наш концентратор-маршрутизатор видит сети из двух OFF-маршрутизаторов.

OFF1#show ip route

```
C    192.168.123.0/24 is directly connected, Serial0/0
C    2.0.0.0/24 is subnetted, 1 subnets
C    2.2.2.0 is directly connected, Loopback0
```

OFF2#show ip route

```
C    192.168.123.0/24 is directly connected, Serial0/0
C    3.0.0.0/24 is subnetted, 1 subnets
C    3.3.3.0 is directly connected, Loopback0
```

К сожалению, наши маршрутизаторы не видят ничего ...

```
CONC#debug ip eigrp
IP-EIGRP Route Events debugging is on
```

Похоже, что маршрутизатор-концентратор не объявляет сети, которые он изучает с помощью OFF-маршрутизаторов. Давайте включим отладку, чтобы увидеть, что происходит.

```
CONC#clear ip eigrp 123 neighbors
```

Сбросим **соседство EIGRP**, чтобы ускорить процесс.

```
CONC# processing incoming UPDATE packet
IP-EIGRP(Default-IP-Routing-Table:123): Processing incoming UPDATE packet
IP-EIGRP(Default-IP-Routing-Table:123): Int 3.3.3.0/24 metric 2297856 -
1657856 640000
IP-EIGRP(Default-IP-Routing-Table:123): Processing incoming UPDATE packet
IP-EIGRP(Default-IP-Routing-Table:123): Int 2.2.2.0/24 M 2297856 - 1657856
640000 SM 128256 - 256 128000
IP-EIGRP(Default-IP-Routing-Table:123): route installed for 2.2.2.0 ()
IP-EIGRP(Default-IP-Routing-Table:123): 192.168.123.0/24 - do advertise out
Serial0/0
```

В отладке мы видим, что наш маршрутизатор-концентратор узнает о сети 2.2.2.0 / 24 и 3.3.3.0 / 24, но объявляет только сеть 192.168.123.0 / 24 для OFF-маршрутизаторов. Разделение горизонта не позволяет размещать объявление от одного маршрутизатора на другой.

```
CONC(config)#interface serial 0/0
CONC(config-if)#no ip split-horizon eigrp 123
```

Давайте отключим разделение горизонта на последовательном интерфейсе маршрутизатора-концентратора.

```
CONC#
IP-EIGRP(Default-IP-Routing-Table:123): 192.168.123.0/24 - do advertise out
Serial0/0
IP-EIGRP(Default-IP-Routing-Table:123): 3.3.3.0/24 - do advertise out
Serial0/0
IP-EIGRP(Default-IP-Routing-Table:123): 2.2.2.0/24 - do advertise out
Serial0/0
```

Теперь мы видим, что маршрутизатор-концентратор объявляет все сети.

```
OFF1#show ip route

C    192.168.123.0/24 is directly connected, Serial0/0
    2.0.0.0/24 is subnetted, 1 subnets
C    2.2.2.0 is directly connected, Loopback0
    3.0.0.0/24 is subnetted, 1 subnets
D    3.3.3.0 [90/2809856] via 192.168.123.1, 00:16:02, Serial0/0
```

```
OFF2#show ip route

C    192.168.123.0/24 is directly connected, Serial0/0
    2.0.0.0/24 is subnetted, 1 subnets
D    2.2.2.0 [90/2809856] via 192.168.123.1, 00:16:33, Serial0/0
    3.0.0.0/24 is subnetted, 1 subnets
C    3.3.3.0 is directly connected, Loopback0
```

OFF-маршрутизаторы теперь могут узнавать о сетях друг друга, поскольку split horizon отключено. Это хорошо, но это еще не все.

**Извлеченный урок:** RIP и EIGRP являются протоколами маршрутизации на расстоянии и используют **split horizon**. Split horizon предотвращает объявление префикса вне интерфейса, на котором мы его узнали.

```
OFF1#ping 3.3.3.3
```

```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:  
.....  
Success rate is 0 percent (0/5)
```

```
OFF2#ping 2.2.2.2
```

```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:  
.....  
Success rate is 0 percent (0/5)
```

Хотя сети отображаются в таблицах маршрутизации мы не можем пропинговать от одного OFF-маршрутизатора к другому. Это не проблема EIGRP, но она связана с Frame Relay. Мы должны это исправить.

Когда OFF1 отправляет IP-пакет на OFF2, IP-пакет выглядит следующим образом:

Источник :	Пункт назначения:
192.168.123.2	3.3.3.3

Давайте пока подумаем, как роутер, и посмотрим, что здесь происходит. Сначала нам нужно проверить, знает ли OFF1, куда отправить 3.3.3.3:

```
OFF1#show ip route 3.3.3.3  
Routing entry for 3.3.3.0/24  
  Known via "eigrp 123", distance 90, metric 2809856, type internal  
  Redistributing via eigrp 123  
  Last update from 192.168.123.1 on Serial0/0, 00:38:10 ago  
  Routing Descriptor Blocks:  
    * 192.168.123.1, from 192.168.123.1, 00:38:10 ago, via Serial0/0  
      Route metric is 2809856, traffic share count is 1  
      Total delay is 45000 microseconds, minimum bandwidth is 1544 Kbit  
      Reliability 255/255, minimum MTU 1500 bytes  
      Loading 1/255, Hops 2
```

Существует запись для 3.3.3.3, а IP-адрес следующего перехода - 192.168.123.1 (маршрутизатор-концентратор). Можем ли мы **достичь** 192.168.123.1?

```
OFF1#ping 192.168.123.1
```

```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 192.168.123.1, timeout is 2 seconds:  
!!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/8 ms
```

Нет проблем, кажется, OFF1 может пересылать пакеты, предназначенные для сети 3.3.3.0/24. Давайте перейдем к маршрутизатору **CONC**.

```
CONC#ping 3.3.3.3
```

```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:  
!!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
```

У маршрутизатора-концентратора нет проблем с отправкой трафика в сеть 3.3.3.0 / 24, поэтому на данный момент мы можем сделать вывод, что проблема должна быть в маршрутизаторе OFF2.

Источник :	Пункт назначения:
192.168.123.2	3.3.3.3

Это IP-пакет, который получает маршрутизатор OFF2, и когда он отвечает, он создает новый IP-пакет, который выглядит следующим образом:

Источник :	Пункт назначения:
3.3.3.3	192.168.123.2

Способен ли OFF2 достигать IP-адрес 192.168.123.2 Давайте узнаем!

```
OFF2#ping 192.168.123.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.123.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Теперь мы знаем проблему ... OFF2 не может достичь IP-адреса 192.168.123.2

```
OFF2#show ip route | include 192.168.123.0
C    192.168.123.0/24 is directly connected, Serial0/0
```

Если мы посмотрим на таблицу маршрутизации OFF2, то увидим, что сеть 192.168.123.0 / 24 подключена напрямую. С точки зрения третьего уровня у нас нет никаких проблем. Пришло время перейти вниз по модели OSI и проверить уровень 2 ... или, может быть, между уровнем 2 и 3.

```
OFF2#show frame-relay map
Serial0/0 (up): ip 192.168.123.1 dlci 301(0x12D,0x48D0), dynamic,
                broadcast,, status defined, active
```

Frame Relay использует Inverse ARP для привязки уровня 2 (DLCI) к уровню 3 (IP-адрес). Вы можете видеть, что нет сопоставления для IP-адреса 192.168.123.2.

```
OFF2(config)#int s0/0
OFF2(config-if)#frame-relay map ip 192.168.123.2 301
```

Давайте frame-relay map сами.

```
OFF2#show frame-relay map
Serial0/0 (up): ip 192.168.123.1 dlci 301(0x12D,0x48D0), dynamic,
                broadcast,, status defined, active
Serial0/0 (up): ip 192.168.123.2 dlci 301(0x12D,0x48D0), static,
                CISCO, status defined, active
```

Теперь роутер OFF2 знает, как связаться с роутером OFF1

```
OFF1#ping 3.3.3.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/10/16 ms
```

Наконец, маршрутизатор OFF1 может пропинговать интерфейс обратной связи маршрутизатора OFF2. Когда мы пытаемся пропинговать от маршрутизатора OFF2 к интерфейсу обратной связи маршрутизатора OFF1, у нас возникает та же проблема, поэтому мы также добавим туда оператор **frame-relay map**:

```
OFF1(config)#int s0/0
OFF1(config-if)#frame-relay map ip 192.168.123.3 201
```



```

OFF1#show frame-relay map
Serial0/0 (up): ip 192.168.123.1 dlci 201(0xC9,0x3090), dynamic,
                broadcast,, status defined, active
Serial0/0 (up): ip 192.168.123.3 dlci 201(0xC9,0x3090), static,
                CISCO, status defined, active

```

Теперь у нас есть extra frame-relay map на маршрутизаторе OFF1.

```

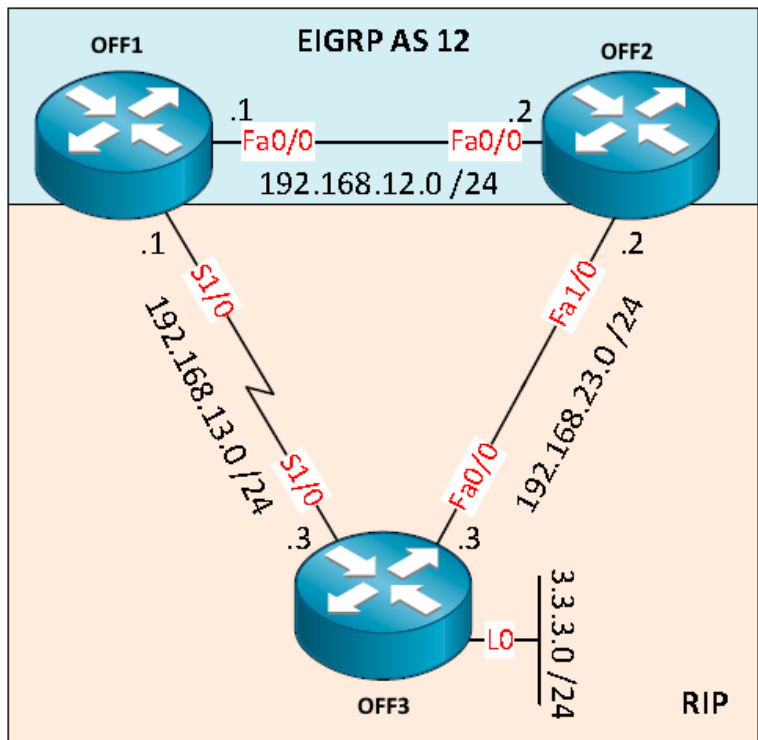
OFF2#ping 2.2.2.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/8 ms

```

И наш пинг проходит! **Извлеченный урок:** убедитесь, что IP-адрес следующего прыжка доступен, и при необходимости добавьте **frame-relay map**.

## CASE #6



Вот вам совершенно другой сценарий. На рисунке выше OFF1 и OFF2 работают с EIGRP AS 12. Все маршрутизаторы работают на RIP. OFF3 имеет **loopback** интерфейс с сетью 3.3.3.0 / 24, который будет объявлен в RIP. OFF2 собирается перераспределить эту сеть в EIGRP AS 12. Вот конфигурации EIGRP и RIP всех маршрутизаторов:

<pre> OFF1# router eigrp 12   network 192.168.12.0   no auto-summary ! router rip   version 2   network 192.168.13.0   no auto-summary </pre>	<pre> OFF2# router eigrp 12   redistribute rip metric 1 1 1 1 1500   network 192.168.12.0   no auto-summary ! router rip   version 2   redistribute eigrp 12 metric 5   network 192.168.23.0   no auto-summary </pre>	<pre> OFF3# router rip   version 2   network 3.0.0.0   network 192.168.13.0   network 192.168.23.0   no auto-summary </pre>
---	---	---

Обратите внимание, что OFF2 был настроен для перераспределения между EIGRP AS 12 и RIP. Как это влияет на наш выбор маршрутизации?

```

OFF1#show ip route 3.3.3.0
Routing entry for 3.3.3.0/24
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 192.168.13.3 on Serial1/0, 00:00:06 ago
  Routing Descriptor Blocks:
  * 192.168.13.3, from 192.168.13.3, 00:00:06 ago, via Serial1/0
    Route metric is 1, traffic share count is 1

```

OFF1 отправляет трафик в сеть 3.3.3.0 / 24, используя последовательную связь между OFF1 и OFF3. Было бы лучше, если бы мы использовали канал между OFF1 и OFF2, потому что использование каналов FastEthernet быстрее, чем последовательный канал.

```

OFF1#show ip eigrp topology 3.3.3.0 255.255.255.0
IP-EIGRP (AS 12): Topology entry for 3.3.3.0/24
  State is Passive, Query origin flag is 1, 0 Successor(s), FD is
  4294967295
  Routing Descriptor Blocks:
  192.168.12.2 (FastEthernet0/0), from 192.168.12.2, Send flag is 0x0
    Composite metric is (2560002816/2560000256), Route is External

```

Мы можем посмотреть на таблицу топологии EIGRP и увидеть, что OFF1 узнал о сети 3.3.3.0 / 24 через EIGRP от OFF2. Так почему же мы не используем эту информацию вместо RIP? Проблема в том, что внешние маршруты EIGRP имеют AD (административное расстояние) 170, а RIP имеет AD 120.

```

OFF1(config)#router rip
OFF1(config-router)#distance 175 192.168.13.3 0.0.0.0

```

Мы можем решить эту проблему, изменив AD. В приведенном выше примере мы установим AD в 175 для всех маршрутов RIP, которые мы узнаем по IP-адресу 192.168.13.3 (OFF3).

```

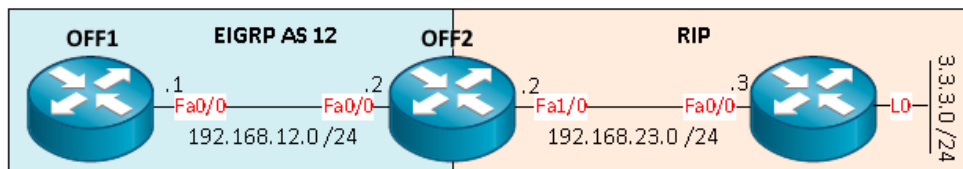
OFF1#show ip route 3.3.3.0
Routing entry for 3.3.3.0/24
  Known via "eigrp 12", distance 170, metric 2560002816, type external
  Redistributing via eigrp 12
  Last update from 192.168.12.2 on FastEthernet0/0, 00:17:53 ago

OFF1#show ip route | include 3.3.3.0
D EX 3.3.3.0 [170/2560002816] via 192.168.12.2, 00:18:46, FastEthernet0/0

```

Так как AD EIGRP external (170) ниже, чем AD маршрутов RIP (175), OFF1 установит информацию EIGRP external . Теперь мы используем каналы FastEthernet для доступа к сети 3.3.3.0 / 24 ... проблема решена!

**Извлеченный урок:** измените административное расстояние, чтобы поменять схему движения, но имейте в виду, в более сложных топологиях это также может вызвать циклы маршрутизации.



Следующая проблема. Существует 3 маршрутизатора, и OFF2 - это маршрутизатор, выполняющий перераспределение между EIGRP AS 12 и RIP. Вот конфигурации:

```

OFF1#
router eigrp 12
  network 192.168.12.0
  no auto-summary

OFF2#
router eigrp 12
  redistribute rip
  network 192.168.12.0
  no auto-summary
!
router rip
  version 2
  redistribute eigrp 12 metric 1
  network 192.168.23.0
  no auto-summary
router rip
  version 2
  network 3.0.0.0
  network 192.168.23.0
  no auto-summary

```

Как вы можете видеть, OFF1 работает с EIGRP, OFF2 работает с RIP и EIGRP и выполняет перераспределение, OFF3 работает только с RIP.

```

OFF1#show ip route
C    192.168.12.0/24 is directly connected, FastEthernet0/0

```

Однако, когда мы изучаем таблицу маршрутизации OFF1, мы ожидаем увидеть некоторые перераспределенные маршруты. К сожалению, здесь ничего нет. В чем дело?

```

OFF2#show ip route rip
  3.0.0.0/24 is subnetted, 1 subnets
R    3.3.3.0 [120/1] via 192.168.23.3, 00:00:23, FastEthernet1/0

```

OFF2 - маршрутизатор, выполняющий перераспределение, и мы видим, что он изучил сеть 3.3.3.0 / 24 через RIP. Следует перераспределить эту сеть в EIGRP.

```

OFF2#show ip eigrp topology 3.3.3.0 255.255.255.0
% IP-EIGRP (AS 12): Route not in topology table

```

Странно, сети 3.3.3.0 / 24 нет в таблице топологии EIGRP OFF2, хотя мы настроили перераспределение, давайте проверим конфигурацию перераспределения:

```

OFF2#
router eigrp 12
  redistribute rip
  network 192.168.12.0
  no auto-summary

```

У нас есть команда **redistribute rip**, но нет никакой начальной метрики по умолчанию. Если вы не зададите метрику, то начальная метрика по умолчанию будет бесконечной. Другими словами, она недостижима.

```

OFF2(config)#router eigrp 12
OFF2(config-router)#default-metric 1500 100 255 1 1500

```

Давайте настроим начальную метрику по умолчанию. Вы должны сами указать пропускную способность, задержку, нагрузку и надежность. Просто используем случайные значения.

```

OFF2#show ip eigrp topology 3.3.3.0 255.255.255.0
IP-EIGRP (AS 12): Topology entry for 3.3.3.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 1732096
  Routing Descriptor Blocks:
    192.168.23.3, from Redistributed, Send flag is 0x0
      Composite metric is (1732096/0), Route is External

```

Теперь мы видим сеть 3.3.3.0 / 24 в таблице топологии EIGRP OFF2.

```
OFF1#show ip route | include 3.3.3.0
```

```
D EX 3.3.3.0 [170/1734656] via 192.168.12.2, 00:04:08, FastEthernet0/0
```

И в результате он может быть объявлен в OFF1 ... проблема решена!

**Извлеченный урок:** когда вы перераспределяете что-то в EIGRP, вам необходимо настроить начальные метрики.