



# Протокол EIGRP (усовершенствованный внутренний протокол маршрутизации шлюзов)

---

**Взаимодействие:** В данном документе содержится анализ конкретного устройства Cisco.

---

## Содержание

### Введение

#### Принцип работы EIGRP

- Основные версии протокола
- Основные принципы
- Обнаружение и обслуживание соседей
- Построение таблицы топологии
- Метрики EIGRP
- Возможное расстояние, объявленное расстояние и возможный преемник
- Определение беспетлевого пути

#### Расщепленный горизонт и обратный запрет

- Режим инициализации
- Изменения в таблице топологии
- Запросы

#### Задержка в активных маршрутах

- Устранение SIA маршрутов

#### Перераспределение

- Перераспределение между двумя автономными системами EIGRP
- Перераспределение между EIGRP и IGRP в двух разных автономных системах
- Перераспределение между EIGRP и IGRP в одной автономной системе
- Перераспределение в другие протоколы и из них
- Перераспределение статических маршрутов на интерфейсы

#### Суммирование

- Автоматическое суммирование
- Ручное суммирование
- Автоматическое суммирование внешних маршрутов

#### Обработка и диапазон запросов

- Каким образом точки суммирования влияют на диапазон запроса
- Каким образом границы автономной системы влияют на диапазон запроса
- Каким образом списки распространения влияют на диапазон запроса

#### Пошаговое продвижение пакетов

#### Маршрутизация по умолчанию

#### Балансировка нагрузки

#### Использование метрик

#### Использование административных тегов в перераспределении

#### Понимание информации, выводимой в листинге команд EIGRP

- show ip eigrp topology
- show ip eigrp topology <network>
- show ip eigrp topology [active | pending | zero-successors]
- show ip eigrp topology all-links

#### Дополнительная информация

---

## Введение

Протокол EIGRP (усовершенствованный внутренний протокол маршрутизации шлюзов) является внутренним протоколом шлюзов и пригоден для использования в различных топологиях и средах. В хорошо спроектированной сети EIGRP хорошо масштабируется и позволяет обеспечить малое время конвергенции при минимальном сетевом трафике.

# Принцип работы EIGRP

Основными преимуществами EIGRP являются:

- низкое потребление сетевых ресурсов в режиме нормальной эксплуатации (в условиях стабильной сети передаются только пакеты "hello")
- при возникновении изменений по сети передаются только изменения, произошедшие в маршрутной таблице, а не вся таблица целиком; это позволяет уменьшить нагрузку на сеть, создаваемую протоколом маршрутизации
- малое время конвергенции в случае изменения в топологии сети (в отдельных случаях сходимость обеспечивается почти мгновенно)

протокол EIGRP является усовершенствованным протоколом дистанционной-векторной маршрутизации, в котором для расчета кратчайшего пути к конечному адресу используется алгоритм диффузного обновления (Diffused Update Algorithm – DUAL).

## Основные версии протокола

Существуют две основные версии протокола EIGRP – версия 0 и 1. В ранних версиях программного обеспечения Cisco IOS (вплоть до версий 10.3(11), 11.0(8) и 11.1(3)) используется более ранняя версия протокола EIGRP (по этой причине некоторые объяснения, содержащиеся в настоящем документе, могут быть не действительны для этих версий). Мы настоятельно рекомендуем использовать последнюю версию EIGRP, поскольку эта версия содержит множество улучшений, связанных со стабильностью и производительностью.

## Основные принципы

При расчете наилучшего пути к конечному адресу типичный дистанционно-векторный протокол сохраняет следующую информацию: расстояние (distance) (суммарная метрика или расстояние, например, счетчик переходов) и вектор (следующий переход). Например, на всех маршрутизаторах сети на рис. 1 выполняется протокол маршрутной информации (Routing Information Protocol – RIP). Маршрутизатор 2 выбирает путь к сети А, проверяя число переходов для каждого имеющегося пути.

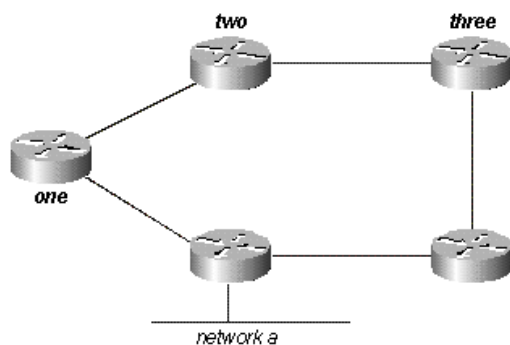


Figure 1

Поскольку длина пути, проходящего через Маршрутизатор 3, равна трем переходам, а длина пути через маршрутизатор 1 равна двум переходам, маршрутизатор 2 выбирает путь, идущий через Маршрутизатор 1, и отбрасывает информацию, полученную от Маршрутизатора 3. Если маршрут, пролегающий между маршрутизатором 1 и сетью А, нарушается, то маршрутизатор 2 теряет связь с этим пунктом назначения до тех пор, пока этот маршрут не будет блокирован в таблице маршрутизации по времени простоя, которое равно трем периодам обновления (90 секунд); а маршрутизатор 3 объявляет маршрут повторно (что происходит в RIP каждые 30 секунд). Если не учитывать период удержания, то маршрутизатору 2 потребуется 90-120 секунд, чтобы перевести путь с маршрутизатора 1 на маршрутизатор 3.

Вместо того, чтобы рассчитывать на полные регулярные обновления для выполнения повторной сходимости, EIGRP (вместо отбрасывания данных) строит таблицу топологии используя для этого все объявления своих соседей и выполняет сходимость либо посредством поиска подходящего беспетлевого маршрута в таблице топологии, либо (если о таком маршруте ничего не известно) посредством опроса своих соседей. Второй маршрутизатор сохраняет информацию, полученную от первого и третьего маршрутизаторов. Он выбирает путь через маршрутизатор 1 как лучший путь ("преемник"), а путь через маршрутизатор 3 – как путь без петель (возможный преемник). Когда маршрут через маршрутизатор 1 становится недоступным, маршрутизатор 2 в поисках

возможного преемника обращается к таблице топологии и сразу же начинает использовать маршрут через маршрутизатор 3.

Из этих кратких пояснений очевидно, что EIGRP должен обеспечить следующее:

- систему, при которой EIGRP пересылает обновления, необходимые в данный момент (это достигается посредством обнаружения и обслуживания соседей)
- способ, позволяющий определить, какой из маршрутов, полученный маршрутизатором, является беспетлевым
- процедуру, позволяющую удалять нерабочие маршруты из таблиц топологии на всех маршрутизаторах, находящихся в сети
- процедуру опроса соседей, которая позволит найти новые маршруты к конечному адресу взамен утраченных старых маршрутов

Все эти требования будут рассмотрены далее.

## Обнаружение и обслуживание соседей

Для распространения маршрутной информации по сети в EIGRP используется неперiodическое инкрементное обновление маршрутов. Это означает, что EIGRP пересылает обновления только для изменяющихся маршрутов и только в тот момент, когда такие маршруты изменяются.

Основной недостаток таких обновлений заключается в том, что вы можете не узнать, в какое время маршрут, проходящий через соседний маршрутизатор, стал недоступным. Нельзя блокировать по времени маршруты, ожидающие получение новой таблицы маршрутизации от соседа. Для надежной пересылки изменений в маршрутной таблице в EIGRP используется механизм взаимодействия между соседними маршрутизаторами (два маршрутизатора становятся "соседями" в том случае, если каждый из них получает пакеты "hello" от своего соседа).

EIGRP посылает пакеты "hello" каждые 5 секунд (для каналов с высокой пропускной способностью) и каждые 60 секунд (для многоточечных каналов с низкой пропускной способностью).

- Пятисекундный пакет "hello" используется:
  - в сетевых средах (например, Ethernet, Token Ring и FDDI)
  - в последовательных каналах "точка-точка" (например, выделенные линии, использующие протоколы PPP или HDLC; подчиненные интерфейсы Frame Relay типа "точка-точка" и подчиненный интерфейс ATM)
  - многоточечные линии с высокой пропускной способностью (выше, чем у линии T1) (например, ISDN PRI и Frame Relay)
- Шестидесяти секундный пакет "hello":
  - многоточечные линии с пропускной способностью как у T1 или ниже – как у многоточечных интерфейсов Frame Relay, многоточечных интерфейсов ATM, коммутируемых виртуальных каналов ATM и базовых интерфейсов обмена ISDN

Частота, с которой EIGRP отправляет пакеты "hello", называется hello-интервалом; этот параметр можно настраивать для каждого интерфейса в отдельности при помощи команды **ip hello-interval eigrp**. Время удержания – время, в течение которого маршрутизатор будет считать соседнее устройство действующим, не получая при этом от него пакет "hello". Время удержания обычно равняется трем hello-интервалам (15 и 180 секунд по умолчанию). Время удержания настраивается при помощи команды **ip hold-time eigrp**.

Обратите внимание на то, что при изменении hello-интервала время удержания не корректируется автоматически (время удержания необходимо изменить вручную в соответствии с новым значением hello-интервала).

Два EIGRP-маршрутизатора могут стать соседями даже в том случае, если таймер hello и таймер удержания не совпадают. Время удержания указывается в пакетах hello, поэтому каждый сосед должен оставаться в рабочем состоянии, даже если интервал hello и таймер удержания не совпадают.

Величину hello-интервала, установленную на маршрутизаторе, невозможно определить напрямую. Однако эту величину можно выяснить из листинга команды **show ip eigrp neighbor**, выполненной на соседнем маршрутизаторе.

Если листинг команды **show ip eigrp neighbor** получен с устройства Cisco, то в этом случае для интерпретации результатов можно воспользоваться утилитой Output Interpreter (только для зарегистрированных клиентов). Данная утилита отображает потенциальные проблемы и предлагает способы их решения. Для использования Output Interpreter необходимо включить JavaScript.

```
router# show ip eigrp neighbor
IP-EIGRP neighbors for process 1
H   Address          Interface   Hold Uptime   SRTT   RTO  Q  Seq  Type
      (sec)          (ms)          Cnt Num
1   10.1.1.2         Et1         13 12:00:53   12    300  0  620
0   10.1.2.2         S0         174 12:00:56   17    200  0  645
```

```
rp-2514aa# show ip eigrp neighbor
IP-EIGRP neighbors for process 1
H   Address          Interface   Hold Uptime   SRTT   RTO  Q  Seq  Type
      (sec)          (ms)          Cnt Num
1   10.1.1.2         Et1         12 12:00:55   12    300  0  620
0   10.1.2.2         S0         173 12:00:57   17    200  0  645
```

```
rp-2514aa# show ip eigrp neighbor
IP-EIGRP neighbors for process 1
H   Address          Interface   Hold Uptime   SRTT   RTO  Q  Seq  Type
      (sec)          (ms)          Cnt Num
1   10.1.1.2         Et1         11 12:00:56   12    300  0  620
0   10.1.2.2         S0         172 12:00:58   17    200  0  645
```

Значение, указанное в листинге команды в столбце "Hold", никогда не должно превышать значения времени удержания, а также не должно быть меньше разницы значений времени удержания и интервала между сообщениями приветствия (если, конечно, пакеты приветствия не были утеряны). Если значение в столбце "Hold" обычно варьируется от 10 до 15 секунд, то интервал приветствия составляет 5 секунд, а время удержания – 15 секунд. Если значение в столбце "Hold" обычно варьируется от 120 до 180 секунд, то интервал приветствия составляет 60 секунд, а время удержания – 180 секунд. Если значения не совпадают со значениями таймера, заданными по умолчанию, тогда необходимо проверить соответствующий интерфейс на соседнем маршрутизаторе (таймер "hello" и таймер интервала приветствия могли быть изменены вручную).

#### Примечание:

- EIGRP не создает одноранговые связи с использованием вторичных адресов. Источник всего трафика EIGRP – основной адрес интерфейса.
- При настройке EIGRP по сети Frame Relay с множественным доступом (многоточечному соединению и др.) в операторах **frame-relay map** следует настроить ключевое слово **broadcast**. Смежность двух маршрутизаторов EIGRP не будет создана без ключевого слова **broadcast**. Более подробно см. раздел "Настройка и устранение неполадок Frame Relay".
- Количество соседних узлов, поддерживаемых EIGRP, не ограничено. Фактическое количество поддерживаемых соседних узлов зависит от характеристик устройства:
  - объем памяти
  - вычислительная мощность
  - объем передаваемой информации (например, количество передаваемых маршрутов)
  - уровень сложности топологии
  - стабильность сети

#### Построение таблицы топологии

Теперь мы знаем, каким образом маршрутизаторы общаются между собой. А вот о чем они "говорят"? Конечно – о таблицах топологии. В отличие от протоколов RIP и IGRP, EIGRP не использует таблицу маршрутизации (или перенаправления) для хранения

всех данных, необходимых для его работы. Вместо этого EIGRP формирует вторую таблицу (таблицу топологии), на основе которой осуществляется установка маршрутов в таблице маршрутизации.

**Примечание:** В Cisco IOS начиная с версий 12.0T и 12.1 протокол RIP ведет свою собственную базу данных, откуда маршруты поступают в таблицу маршрутизации.

Чтобы узнать основной формат таблицы топологии на EIGRP-маршрутизаторе, выполните команду **show ip eigrp topology**. В таблице топологии содержатся сведения, необходимые для построения набора расстояний и векторов для каждой достижимой сети, включая:

- наименьшая пропускная способность на маршруте, идущем к данной сети (данные предоставляются восходящим соседом)
- суммарная задержка
- надежность маршрута
- загрузка маршрута
- минимальный путь максимального размера передаваемого блока данных (MTU)
- возможное расстояние
- объявленное расстояние
- источник маршрута (внешние узлы маркируются)

Вероятное и фактическое расстояния будут обсуждаться далее в этом разделе.

Если листинг команды **show ip eigrp topology** получен с устройства Cisco, то в этом случае для интерпретации результатов можно воспользоваться утилитой Output Interpreter (только для зарегистрированных клиентов). Данная утилита отображает потенциальные проблемы и предлагает способы их решения. Для использования Output Interpreter необходимо включить JavaScript.

## Метрики EIGRP

Для расчета метрик маршрутизации протокол EIGRP использует минимальную пропускную способность маршрута до конечного адреса, а также суммарную задержку. Можно также настроить и другие метрики. Однако мы не рекомендуем делать этого, поскольку в этом случае в вашей сети могут появиться петли по маршрутизации. Метрики пропускной способности и задержки определяются на основе значений, установленных на интерфейсах маршрутизаторов, которые являются частью маршрута к сети назначения.

На рисунке 2 показано, как маршрутизатор 1 вычисляет наилучший путь в сеть A.

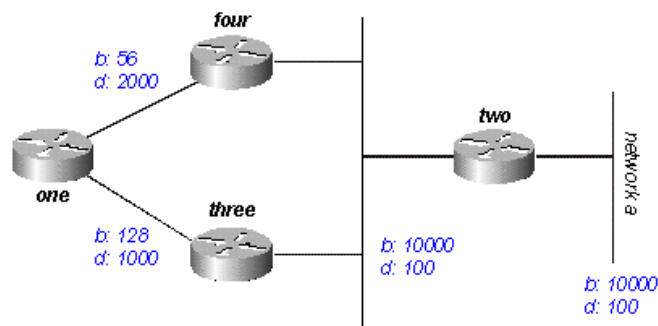


Figure 2

Расчет начинается с двух объявлений, касающихся искомой сети: первый маршрут проходит через маршрутизатор 4 (минимальная пропускная способность – 56 и суммарная задержка – 2200), а второй – через маршрутизатор 3 (минимальная пропускная способность – 128 и суммарная задержка – 1200). Маршрутизатор 1 выбирает маршрут с наименьшим значением метрики.

Теперь вычислим метрику. EIGRP вычисляет суммарную метрику путем взвешивания метрики пропускной способности и метрики задержки. Для измерения пропускной способности EIGRP использует следующую формулу:

- $\text{пропуск. способн.} = (10000000/\text{пропуск. способн.}(i)) * 256$

где пропускная способность(i) – наименьшая пропускная способность всех выходных интерфейсов, входящих в состав маршрута, ведущего к сети назначения (измеряется в килобитах).

Для вычисления задержки EIGRP использует следующую формулу:

- $\text{задержка} = \text{задержка}(i) * 256$

где задержка(i) – сумма задержек, которые настроены на всех интерфейсах, входящих в состав маршрута к сети назначения (измеряется в десятках микросекунд). В листинге команд **show ip eigrp topology** и **show interface** задержка указана в микросекундах, поэтому для использования этих значений в формуле их необходимо разделить на 10. В этом документе используется задержка, настроенная и показанная на интерфейсе.

Указанные выше значения используются для определения суммарной метрики для искомой сети:

- $\text{метрика} = [\text{K1} * \text{пропускная способность} + (\text{K2} * \text{пропускная способность}) / (256 - \text{нагрузка}) + \text{K3} * \text{задержка}] * [\text{K5} / (\text{надежность} + \text{K4})]$

**Примечание:** значения **K** следует использовать только после тщательного планирования. Неверные значения **K** приводят к невозможности построения связей между соседними узлами, в результате чего в вашей сети будет невозможно выполнить сходимости.

**Примечание:** при  $\text{K5} = 0$  формула принимает следующий вид:  $\text{метрика} = [\text{k1} * \text{пропускная способность} + (\text{k2} * \text{пропускная способность}) / (256 - \text{нагрузка}) + \text{k3} * \text{задержка}]$ .

По умолчанию **K** имеет следующие значения:

- $\text{K1} = 1$
- $\text{K2} = 0$
- $\text{K3} = 1$
- $\text{K4} = 0$
- $\text{K5} = 0$

Для функционирования по умолчанию формулу формулу можно упростить следующим образом:

```
metric = bandwidth + delay
```

Маршрутизаторы Cisco не выполняют расчеты с плавающей запятой, поэтому для правильного подсчета метрики на каждом этапе калькуляции полученный результат необходимо округлять в меньшую сторону до целого числа. В этом примере общая стоимость маршрута через маршрутизатор 4 равна:

```
minimum bandwidth = 56k
```

```
total delay = 100 + 100 + 2000 = 2200
```

```
[(10000000/56) + 2200] x 256 = (178571 + 2200) x 256 = 180771 x 256 = 46277376
```

А общая стоимость маршрута через маршрутизатор 3 равна:

```
minimum bandwidth = 128k
total delay = 100 + 100 + 1000 = 1200
[(10000000/128) + 1200] x 256 = (78125 + 1200) x 256 = 79325 x 256 = 20307200
```

Таким образом, чтобы достичь сети А, маршрутизатор 1 выбирает маршрут через маршрутизатор 3.

Обратите внимание на то, что значения пропускной способности и задержки – это значения, настроенные на интерфейсе, через который маршрутизатор достигает следующего сегмента по пути к сети назначения. Например, маршрутизатор 2 объявил о сети А и указал задержку, настроенную на его Ethernet-интерфейсе. Затем маршрутизатор 4 добавил к этому значению задержку, настроенную на его Ethernet-интерфейсе, а маршрутизатор 1 добавил задержку, настроенную на его последовательном порте.

### Возможное расстояние, объявленное расстояние и возможный преемник

Возможное расстояние – это наилучшая метрика по маршруту в сеть назначения, включая метрику до соседа, объявившего маршрут. Объявленное расстояние является суммарной метрикой пути к сети назначения, который объявлен восходящим соседом. Возможный заместитель – это маршрут, величина объявленного расстояния которого меньше величины возможного расстояния (текущий наилучший маршрут). На рис. 3 изображен этот процесс:

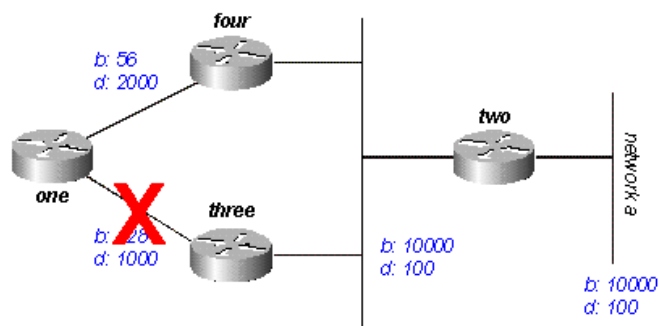


Figure 3

Маршрутизатор 1 определил, что в сеть А имеются два маршрута: один маршрут проходит через маршрутизатор 3, а другой – через маршрутизатор номер 4.

- Стоимость маршрута, проходящего через маршрутизатор 4, равняется 46277376, а объявленное расстояние – 307200.
- Стоимость маршрута, проходящего через маршрутизатор 3, равна 20307200, объявленное расстояние равно 307200.

Обратите внимание, что в каждом случае EIGRP отсчитывает объявленное расстояние начиная с маршрутизатора, объявившего маршрут к искомой сети. Другими словами, объявленное расстояние от маршрутизатора 4 – это метрика маршрута до сети А, начинающегося с маршрутизатора 4, а объявленное расстояние от маршрутизатора 3 – это метрика маршрута до сети А начиная с маршрутизатора 3. EIGRP выбирает маршрут через маршрутизатор 3 в качестве наилучшего и использует метрику через маршрутизатор 3 как возможное расстояние. Поскольку объявленное расстояние до этой сети через маршрутизатор 4 меньше возможного расстояния, маршрутизатор 1 считает путь через маршрутизатор 4 возможным преемником.

Когда связь между маршрутизаторами 1 и 3 нарушается, маршрутизатор 1 перебирает все известные ему пути к сети А, и определяет, что у него есть возможный преемник, доступный через маршрутизатор 4. Маршрутизатор 1 обращается к этому маршруту, используя метрику через маршрутизатор 4 в качестве нового возможного расстояния. Сходимость выполняется мгновенно, а обновления, поступающие к нисходящим соседям, являются единственным трафиком, который генерируется протоколом маршрутизации.

Теперь давайте рассмотрим более сложный случай, показанный на рисунке 4.

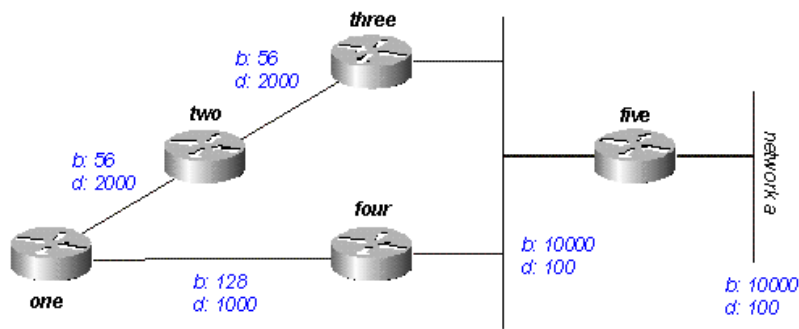


Figure 4

От маршрутизатора 1 в сеть А ведут два маршрута: один маршрут проходит через маршрутизатор 2 и имеет метрику 46789376, а другой через маршрутизатор 4 и имеет метрику 20307200. В качестве маршрута в сеть А маршрутизатор 1 выберет наименьшую из двух метрик, которая станет возможным расстоянием. Теперь рассмотрим маршрут, идущий через маршрутизатор 2. Нам необходимо выяснить, подходит ли этот маршрут в качестве возможного преемника. Объявленное расстояние от второго маршрутизатора равно 46277376. Это значение выше значения возможного расстояния, поэтому этот путь не является возможным преемником. Если бы мы посмотрели таблицу топологии маршрутизатора 1 (при помощи команды **show ip eigrp topology**), то мы бы увидели только одну запись для сети А – через маршрутизатор 4. (В действительности в таблице топологии маршрутизатора 1 имеются две записи, но возможным преемником может быть только одна из них, другая же запись в листинге команды **show ip eigrp topology** отображена не будет; все маршруты, которые не являются возможными преемниками, можно посмотреть воспользовавшись командой **show ip eigrp topology all-links**).

Допустим, что соединение между маршрутизаторами 3 и 4 было нарушено. Обнаружив, что единственный путь к сети А утрачен, маршрутизатор 1 начинает опрашивать всех своих соседей на предмет маршрута в сеть А. Поскольку маршрутизатор 2 может предложить маршрут в сеть А, он откликнется на запрос, инициированный маршрутизатором 1. Поскольку наилучший маршрут, идущий через маршрутизатор 4, был утрачен, маршрутизатор 1 принимает маршрут, который идет в сеть А через маршрутизатор 2.

## Определение беспетлевого пути

Каким образом протокол EIGRP использует понятия возможного расстояния, объявленного расстояния и возможного преемника для того, чтобы определить, является ли данный путь действительным и беспетлевым? На рисунке 4а показано, что маршрутизатор 3 выполняет оценку маршрутов в сеть А. Поскольку функция "разделенный горизонт" отключена (например, это необходимо при работе с многоточечными интерфейсами Frame Relay), маршрутизатор 3 укажет в сеть А три маршрута: через маршрутизатор 4, через маршрутизатор 2 (путь проходит через маршрутизаторы два, один, три и четыре) и через маршрутизатор 1 (путь проходит через маршрутизаторы один, два, три, четыре).

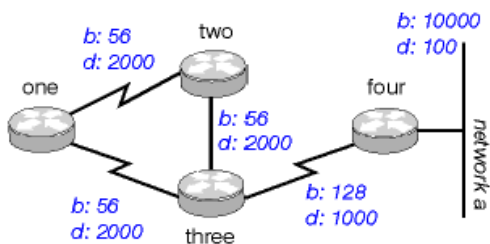


Figure 4a

Если маршрутизатор примет все эти маршруты, то образуется маршрутная петля. Маршрутизатор 3 полагает, что он сможет попасть в сеть А через маршрутизатор 2, однако прежде чем попасть в сеть А, путь, идущий через маршрутизатор 2, проходит через маршрутизатор 3. Если соединение между маршрутизатором 4 и маршрутизатором 3 будет нарушено, тогда маршрутизатор 3 будет считать, что он сможет достигнуть сети А используя один из оставшихся маршрутов, но из-за действия правил для определения возможных преемников он никогда не будет использовать эти маршруты в качестве альтернативных. Чтобы это понять, обратимся к метрике:

- суммарная метрика в сеть А через маршрутизатор 4: 20281600
- суммарная метрика в сеть А через маршрутизатор 3: 47019776
- суммарная метрика в сеть А через маршрутизатор 1: 47019776



Поскольку путь через маршрутизатор 4 обладает наилучшей метрикой, маршрутизатор 3 устанавливает этот маршрут в таблице перенаправлений и использует метрику 20281600 в качестве возможного расстояния в сеть А. Затем маршрутизатор 3 вычисляет объявленное расстояние в сеть А для путей, идущих через маршрутизаторы 2 и 1: 47019776 соответствует пути через маршрутизатор 2, а 47019776 соответствует пути через маршрутизатор 1. Поскольку значение этих метрик превышает значение возможного расстояния, маршрутизатор 3 не назначит ни один из этих маршрутов в качестве возможных приемников, ведущих в сеть А.

Предположим, что связь между маршрутизаторами 3 и 4 была нарушена. Маршрутизатор 3 запрашивает альтернативный маршрут к сети А у всех соседних узлов. Маршрутизатор 2 получает запрос и, поскольку этот запрос поступает от приемника, выполняет поиск всех записей о возможных приемниках в таблице топологии. Единственная такая запись в таблице топологии принадлежит маршрутизатору 1 (объявленное расстояние этой записи равняется значению последней известной метрике через маршрутизатор 3). Поскольку объявленное расстояние через маршрутизатор 1 больше чем последнее известное возможное расстояние, маршрутизатор 2 помечает этот маршрут как недостижимый и начинает опрашивать своих соседей (в этом случае опрашивается только маршрутизатор 1) на предмет пути в сеть А.

Маршрутизатор 3 отправляет маршрутизатору 1 запрос относительно сети А. Маршрутизатор 1 выполняет поиск в своей таблице топологии и обнаруживает, что только другой единственный маршрут к сети А проходит через маршрутизатор 2 с объявленным расстоянием, равным последнему известному возможному расстоянию через маршрутизатор 3. Поскольку объявленное расстояние через маршрутизатор 2 опять-таки превышает последнее известное возможное расстояние, этот маршрут не является возможным приемником. Маршрутизатор 1 помечает этот маршрут как недостижимый и запрашивает у единственного оставшегося соседа (маршрутизатор 2) путь к сети А.

Пытаясь найти маршрут в сеть А маршрутизатор 3 опросил всех своих соседей. В свою очередь маршрутизаторы 1 и 2 помечают маршрут как недостижимый и опросили своих соседей, чтобы найти путь в сеть А. При получении запроса от маршрутизатора 1 маршрутизатор 2 выполняет поиск по своей таблице топологии, в ходе которого обнаруживает, что пункт назначения помечен как недостижимый. Маршрутизатор 2 отвечает маршрутизатору 1, что сеть А недоступна. Когда маршрутизатор 1 получает запрос от маршрутизатора 2, он также посылает обратно ответ о том, что сеть А недоступна. Поскольку маршрутизаторы 1 и 2 установили, что сеть А не доступна, они отвечают на первоначальный запрос маршрутизатора 3. Сходимость закончена, и все маршруты переходят в пассивное состояние.

## Расщепленный горизонт и обратный запрет

В предыдущем примере мы оговорились, что функция "расщепленный горизонт" была выключена (мы хотели показать, каким образом EIGRP использует возможное расстояние и объявленное расстояние, чтобы определить, является ли маршрут петлей). Однако в некоторых случаях EIGRP использует расщепленный горизонт, чтобы предотвратить возникновения петель маршрутизации. Прежде чем начать рассматривать особенности использования расщепленного горизонта, необходимо объяснить, что это такое и как это работает. Правило расщепленного горизонта выглядит следующим образом:

- никогда не объявлять маршрут через интерфейс, посредством которого маршрутизатор узнал о маршруте.

Приведем пример. На рисунке 4а показано, что если маршрутизатор 1 подключен к маршрутизаторам 2 и 3 через единый многоточечный интерфейс (например, Frame Relay) и если при этом маршрутизатор 1 узнает о сети А через маршрутизатор 2, то он не будет использовать тот же самый интерфейс, чтобы объявить маршрутизатору 3 маршрут в сеть А. Маршрутизатор 1 предполагает, что маршрутизатор 3 получит информацию о сети А непосредственно от маршрутизатора 2.

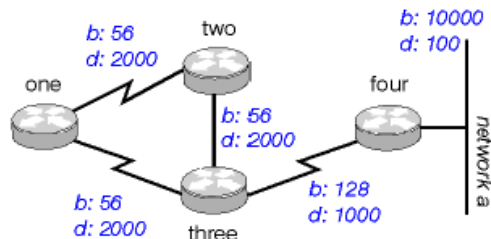


Figure 4a

Обратный запрет – это еще один способ предотвратить появление петель. Правило обратного запрета выглядит следующим образом:

- если информация о маршруте поступила через какой-либо интерфейс, то этот маршрут необходимо объявить недостижимым через тот же самый интерфейс.

Предположим, что на маршрутизаторах, показанных на рисунке 4а, включена функция обратного запрета.. Когда маршрутизатор 1 узнает от маршрутизатора 2 о сети А, он объявляет маршрутизаторам 2 и 3, что сеть А недостижима через его канал. В случае если маршрутизатор 3 указывает какой-либо путь в сеть А через маршрутизатор 1, он должен удалить его, поскольку было объявлено о том, что сеть А недостижима через этот путь. EIGRP использует два этих правила, чтобы предотвратить возникновение петель маршрутизации.

EIGRP использует расщепленный горизонт и объявляет маршрут недостижимым в следующих случаях:

- если два маршрутизатора работают в режиме инициализации (т.е. выполняют первичный обмен таблицами топологии)
- если объявляется об изменении в таблице топологии
- при отправке запроса

Теперь рассмотрим каждый из этих случаев.

### Режим инициализации

Когда два маршрутизатора впервые становятся "соседями", они, находясь в режиме инициализации, обмениваются таблицами топологии. Каждая запись в таблице, которую маршрутизатор принял в режиме инициализации, повторно объявляется (принимающим маршрутизатором) новому соседу, при этом к этой записи применяется максимальная метрика (это так называемый "запрещенный маршрут" – poison route, букв. отравленный маршрут).

### Изменения в таблице топологии

На рисунке 5 в маршрутизаторе 1 используется дисперсия для балансировки трафика, предназначенного для сети А на отрезке между двумя последовательными каналами (канал в 56 Кбит между маршрутизаторами 2 и 4 и канал в 128 Кбит между маршрутизаторами 3 и 4). (Подробно о дисперсии см. Балансировка нагрузки, раздел, в котором обсуждается дисперсия).

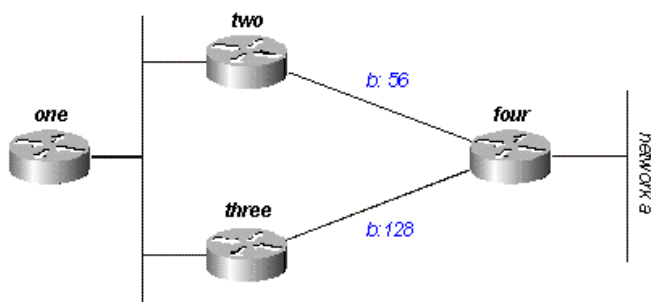


Figure 5

Маршрутизатор 2 рассматривает путь через маршрутизатор 3 в качестве возможного приемника. Если канал между маршрутизаторами 2 и 4 будет нарушен, то маршрутизатор 2 выполнит сходимость по маршруту, проходящему через маршрутизатор 3. Поскольку правило расщепленного горизонта гласит, что нельзя объявлять маршрут через интерфейс, посредством которого были получены сведения об этом маршруте, маршрутизатор 2 обычно не передает обновление. Однако это оставляет маршрутизатор 1 с недопустимой записью в таблице топологии. Когда маршрутизатор изменяет свою таблицу топологии таким образом, что при этом происходит смена интерфейса, через который маршрутизатор соединяется с сетью, то он отключает расщепленный горизонт, а исправления меняют все маршруты, исходящие из интерфейсов, в обратном направлении. В этом случае маршрутизатор 2 отключает расщепленный горизонт для данного маршрута и объявляет сеть А недоступной. При получении этого объявления Маршрутизатор 1 удаляет из своей таблицы маршрутизации свой маршрут к сети А, идущий через маршрутизатор 2.

### Запросы

Запросы приводят к возникновению расщепленного горизонта только тогда, когда маршрутизатор получает запрос или обновление от приемника, которого он использует для получателя в запросе. Рассмотрим сеть, показанную на рисунке 6.

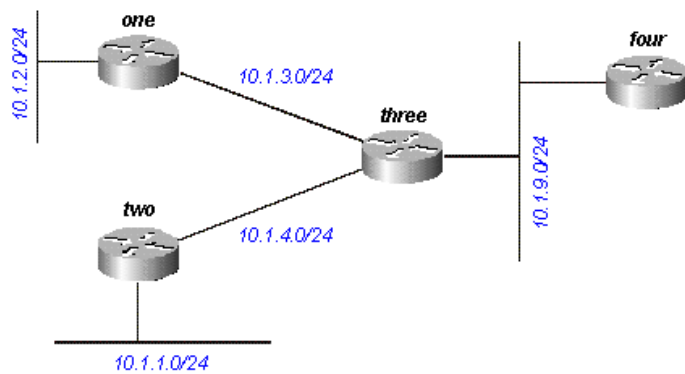


Figure 6

Маршрутизатор 3 получает от маршрутизатора 4 запрос о сети 10.1.2.0/24 (которой он достигает через маршрутизатор 1). Если у маршрутизатора 3 отсутствует преемник для данного пункта назначения (по причине переключения линии или из-за другой временной ситуации в сети), этот маршрутизатор отправляет запрос каждому из своих соседей (в данном случае это маршрутизаторы 1, 2 и 4). Однако если маршрутизатор 3 получит от маршрутизатора 1 запрос или обновление (например, изменение метрики) для сети 10.1.2.0/24, тот он не будет отправлять запрос обратно маршрутизатору 1, поскольку последний является его преемником на маршруте к этой сети. Вместо этого он отправит запросы только маршрутизаторам 2 и 4.

## Задержка в активных маршрутах

При определенных обстоятельствах может пройти большое количество времени, прежде чем на запрос будет получен ответ. "Молчание" может быть настолько длительным, что инициировавший запрос маршрутизатор может прекратить ожидание и сбросить соединение с "молчащим" маршрутизатором, при этом происходит фактический перезапуск сеанса связи с соседним узлом. Такое событие называется "затор на активном маршруте" (stuck in active – SIA). Наиболее простые случаи SIA возникают, если требуется слишком много времени для достижения запросом другого конца сети, а также для обратного прохождения отклика. Например, на рисунке 7 маршрутизатор 1 записывает большое число маршрутов SIA от маршрутизатора 2.

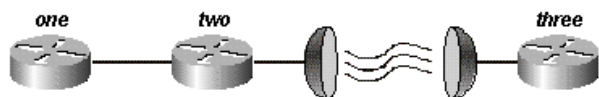


Figure 7

После проведения проверок обнаруживается, что проблема сводится к наличию задержки в спутниковом канале между маршрутизаторами 2 и 3. Существуют два возможных решения проблемы этого типа. Первое – увеличить интервал, в течение которого маршрутизатор будет ожидать ответа на запрос, прежде чем объявить маршрут маршрутом SIA. Эту настройку можно изменить с помощью команды **timers active-time**.

Однако оптимальное решение проблемы заключается в том, чтобы перекомпоновать сеть таким образом, чтобы уменьшить диапазон запросов (чтобы по спутниковому каналу передавалось минимальное количество запросов). Этот вопрос рассматривается в разделе Диапазон запросов. Проблема диапазона запросов не является распространенной причиной возникновения SIA маршрутов. Значительно чаще некоторые маршрутизаторы в сети не могут ответить на запрос по одной из следующих причин:

- маршрутизатор сильно загружен работой и не может ответить на запрос (обычно это связано с сильной загрузкой ЦП)
- у маршрутизатора имеются трудности при работе с памятью, и он не может выделить память для обработки запроса или для формирования пакета отклика
- канал между двумя маршрутизаторами не обеспечивает достаточного качества (это значит, что по такому каналу по-прежнему можно передавать пакеты в количестве, достаточном для поддержания связи "сосед-сосед", однако при этом некоторые запросы или отклики все-таки теряются при пересылке от одного маршрутизатора к другому)
- однонаправленные каналы (т.е. каналы, в которых из-за сбоя трафик может передаваться только в одном направлении)

## Устранение SIA маршрутов

Процесс устранения маршрутов SIA обычно состоит из трех этапов:

1. Сначала необходимо установить маршруты, о которых система из раза в раз сообщает как о маршрутах SIA.
2. Затем необходимо установить маршрутизатор, который из раза в раз не отвечает на запросы на данных маршрутах.
3. Потом необходимо определить причину, по которой маршрутизатор не получает запросов или не отвечает на них.

Первый шаг выполняется относительно просто. Если в вашей системе ведется учет консольных сообщений, то при быстром просмотре журнала можно определить, какие маршруты чаще всего помечаются как маршруты SIA. Второй шаг более сложный. Для сбора этой информации необходимо использовать команду **show ip eigrp topology active**:

```
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status

A 10.2.4.0/24, 0 successors, FD is 512640000, Q
  1 replies, active 00:00:01, query-origin: Local origin
    via 10.1.2.2 (Infinity/Infinity), Serial1
  1 replies, active 00:00:01, query-origin: Local origin
    via 10.1.3.2 (Infinity/Infinity), r, Serial3
  Remaining replies:
    via 10.1.1.2, r, Serial0
```

Все соседние узлы, для которых отображена буква **R**, еще только должны ответить (активный таймер показывает время, на протяжении которого маршрут является активным). Обратите внимание на то, что соседние узлы могут отсутствовать в разделе "Remaining replies" (ожидаемые отклики). Эти узлы могут появиться среди других RDB. Особое внимание необходимо уделять маршрутам, которые были активны в течение определенного времени (обычно 2-3 минуты) и ответ от которых только ожидается. Выполните эту команду несколько раз, чтобы выяснить, какие соседи не отвечают на запросы или какие интерфейсы имеют много неотвеченных запросов. Проверьте этот соседний узел, чтобы узнать, находится ли он постоянно в режиме ожидания откликов от своих соседей. Повторяйте эту процедуру до тех пор, пока не найдете маршрутизатор, который из раза в раз не отвечает на запросы. Можно также поискать неполадки, относящиеся к соединению с этим соседом, к памяти, а также неполадки, связанные с загрузкой ЦП и другими проблемами, имеющимися у данного соседа.

Если вы сталкиваетесь с ситуацией, когда по вашему мнению причиной неполадок является диапазон запросов, лучшим решением проблемы будет уменьшить диапазон запроса (а не увеличивать значения таймера SIA).

## Перераспределение

Данный раздел содержит описание различных случаев перераспределения. Пожалуйста, помните о том, что в примерах, рассматриваемых ниже, указан минимум, который необходим для настройки перераспределения. Перераспределение связано с потенциальными проблемами, такими как: неоптимальная маршрутизация, образование петель или медленная сходимости. Чтобы избежать подобных проблем, обратитесь к разделу "Как избежать проблем при перераспределении" в документе "Перераспределение протоколов маршрутизации".

### Перераспределение между двумя автономными системами EIGRP

Маршрутизаторы, показанные на рисунке 8, настроены следующим образом:

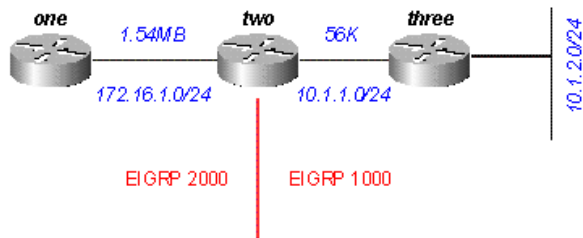


Figure 8

## Маршрутизатор 1

```
router eigrp 2000
!--- 2000 - это автономная система
network 172.16.1.0 0.0.0.255
```

## Маршрутизатор 2

```
router eigrp 2000
redistribute eigrp 1000 route-map to-eigrp2000
network 172.16.1.0 0.0.0.255
!
router eigrp 1000
redistribute eigrp 2000 route-map to-eigrp1000
network 10.1.0.0 0.0.255.255

route-map to-eigrp1000 deny 10
match tag 1000
!
route-map to-eigrp1000 permit 20
set tag 2000
!
route-map to-eigrp2000 deny 10
match tag 2000
!
route-map to-eigrp2000 permit 20
set tag 1000
```

## Маршрутизатор 3

```
router eigrp 1000
network 10.1.0.0 0.0.255.255
```

Маршрутизатор 3 объявляет маршрутизатору 2 о сети 10.1.2.0/24 через автономную систему 1000. Маршрутизатор 2 перераспределяет этот маршрут в автономную систему 2000 и объявляет о нем маршрутизатору 1.

**Примечание:** перед перераспределением в EIGRP 2000 маршруты, идущие из EIGRP 1000, помечаются как 1000. Когда маршруты, идущие из EIGRP 2000, перераспределяются обратно в EIGRP 1000, то эти метки отбрасываются с тем, чтобы обеспечить беспетлевую топологию. Более подробно о перераспределении между протоколами маршрутизации см. в разделе Перераспределение протоколов маршрутизации.

На маршрутизаторе 1 можно видеть следующее:

```
one# show ip eigrp topology 10.1.2.0 255.255.255.0
IP-EIGRP topology entry for 10.1.2.0/24
State is Passive, Query origin flag is 1, 1 Successor(s), FD is 46763776
Routing Descriptor Blocks:
20.1.1.1 (Serial0), from 20.1.1.1, Send flag is 0x0
Composite metric is (46763776/46251776), Route is External
Vector metric:
  Minimum bandwidth is 56 Kbit
  Total delay is 41000 microseconds
  Reliability is 255/255
  Load is 1/255
  Minimum MTU is 1500
  Hop count is 2
External data:
  Originating router is 10.1.2.1
  AS number of route is 1000
  External protocol is EIGRP, external metric is 46251776
  Administrator tag is 1000 (0x000003E8)
```

Обратите внимание, что несмотря на то, что пропускная способность канала между маршрутизаторами 1 и 2 составляет 1,544 Мбит, минимальная пропускная способность, указанная в таблице топологии, равна 56 Кбит. Это значит, что при операциях перераспределения между двумя автономными системами EIGRP протокол EIGRP сохраняет все метрики.

## Перераспределение между EIGRP и IGRP в двух разных автономных системах

На рисунке 9 настройки были изменены следующим образом:

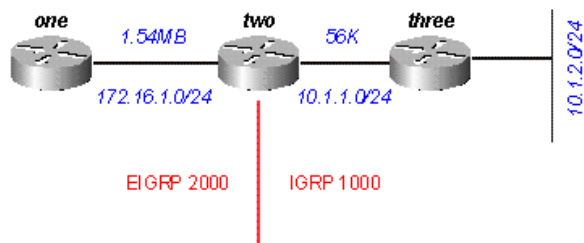


Figure 9

### Маршрутизатор 1

```
router eigrp 2000
network 172.16.1.0
```

### Маршрутизатор 2

```
router eigrp 2000
redistribute igrp 1000 route-map to-eigrp2000
network 172.16.1.0
!
router igrp 1000
redistribute eigrp 2000 route-map to-igrp1000
network 10.0.0.0
!

route-map to-igrp1000 deny 10
match tag 1000
!
route-map to-igrp1000 permit 20
set tag 2000
!
route-map to-eigrp2000 deny 10
match tag 2000
!
route-map to-eigrp2000 permit 20
set tag 1000
```

### Маршрутизатор 3

```
router igrp 1000
network 10.0.0.0
```

Конфигурация маршрутизатора 1 приведена ниже:

```
one# show ip eigrp topology 10.1.2.0 255.255.255.0
IP-EIGRP topology entry for 10.1.2.0/24
```

```

State is Passive, Query origin flag is 1, 1 Successor(s), FD is 46763776
Routing Descriptor Blocks:
20.1.1.1 (Serial0), from 20.1.1.1, Send flag is 0x0
  Composite metric is (46763776/46251776), Route is External
  Vector metric:
    Minimum bandwidth is 56 Kbit
    Total delay is 41000 microseconds
    Reliability is 255/255
    Load is 1/255
    Minimum MTU is 1500
    Hop count is 1
  External data:
    Originating router is 10.1.1.1
    AS number of route is 1000
    External protocol is IGRP, external metric is 180671
    Administrator tag is 1000 (0x000003E8)

```

Метрики IGRP сохраняются, когда маршруты перераспределяются в EIGRP с другой автономной системой, но они масштабируются путем умножения метрики IGRP на константу 256. Следует обратить внимание на один момент, касающийся перераспределения между IGRP и EIGRP. Если сеть не подключена напрямую к маршрутизатору, который выполняет перераспределение, то она объявляет маршрут с метрикой, равной 1.

Например, сеть 10.1.1.0/24 напрямую подсоединена к маршрутизатору 2, а IGRP выполняет маршрутизацию для этой сети (есть инструкция сети для маршрутизатора IGRP, который обслуживает этот интерфейс). Для этой сети протокол EIGRP не выполняет маршрутизацию, но он узнает об этом напрямую подключенном интерфейсе через перераспределение от IGRP. На маршрутизаторе 1 запись в таблице топологии для 10.1.1.0/24 показывает:

```

one# show ip eigrp topology 10.1.1.0 255.255.255.0
IP-EIGRP topology entry for 10.1.1.0/24
State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2169856
Routing Descriptor Blocks:
20.1.1.1 (Serial0), from 20.1.1.1, Send flag is 0x0
  Composite metric is (2169856/1), Route is External

  Vector metric:
    Minimum bandwidth is 1544 Kbit
    Total delay is 20000 microseconds
    Reliability is 0/255
    Load is 1/255
    Minimum MTU is 1500
    Hop count is 1
  External data:
    Originating router is 10.1.1.1
    AS number of route is 1000
    External protocol is IGRP, external metric is 0
    Administrator tag is 1000 (0x000003E8)

```

Обратите внимание, что объявленное расстояние от маршрутизатора 2, который выделен полужирным шрифтом, равно "1".

## Перераспределение между EIGRP и IGRP в одной автономной системе

На рисунке 10 в конфигурацию маршрутизатора внесены следующие изменения:

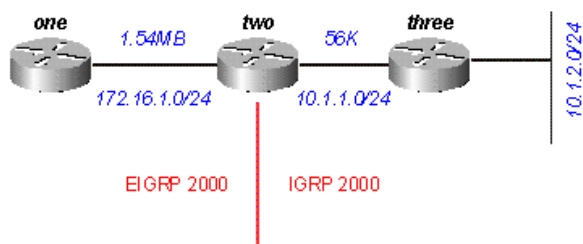


Figure 10

## Маршрутизатор 1

```
router eigrp 2000
network 172.16.1.0
```

## Маршрутизатор 2

```
router eigrp 2000
network 172.16.1.0
!
router igrp 2000
network 10.0.0.0
```

## Маршрутизатор 3

```
router igrp 2000
network 10.0.0.0
```

Маршрутизатор 1 настроен следующим образом:

```
one# show ip eigrp topology 10.1.2.0 255.255.255.0
IP-EIGRP topology entry for 10.1.2.0/24
State is Passive, Query origin flag is 1, 1 Successor(s), FD is 46763776
Routing Descriptor Blocks:
20.1.1.1 (Serial0), from 20.1.1.1, Send flag is 0x0
Composite metric is (46763776/46251776), Route is External
Vector metric:
Minimum bandwidth is 56 Kbit
Total delay is 41000 microseconds
Reliability is 255/255
Load is 1/255
Minimum MTU is 1500
Hop count is 1
External data:
Originating router is 10.1.1.1
AS number of route is 2000
External protocol is IGRP, external metric is 180671
Administrator tag is 0 (0x00000000)
```

Эта конфигурация выглядит удивительно похожей на предыдущий результат, когда мы выполняли перераспределение между двумя разными автономными системами, использующими соответственно IGRP и EIGRP. Непосредственно примыкающая 10.1.1.0/24 сеть регулируется одинаково в обоих случаях:

```
one# show ip eigrp topology 10.1.1.0 255.255.255.0
IP-EIGRP topology entry for 10.1.1.0/24
State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2169856
Routing Descriptor Blocks:
20.1.1.1 (Serial0), from 20.1.1.1, Send flag is 0x0
Composite metric is (2169856/1), Route is External
Vector metric:
Minimum bandwidth is 1544 Kbit
Total delay is 20000 microseconds
Reliability is 255/255
Load is 1/255
Minimum MTU is 1500
Hop count is 1
External data:
Originating router is 10.1.1.1
AS number of route is 2000
External protocol is IGRP, external metric is 0
Administrator tag is 0 (0x00000000)
```



Эта сеть, напрямую соединенная с маршрутизатором 1, перераспределяется от IGRP в EIGRP со значением метрики равным 1 (такая же метрика используется при перераспределении между двумя различными автономными системами).

Следует обратить внимание на пару моментов, касающихся перераспределения EIGRP/IGRP в пределах одной автономной системы:

- Внутренним маршрутам EIGRP всегда отдается предпочтение перед внешними маршрутами EIGRP или IGRP.
- Метрики внешних маршрутов EIGRP сравниваются с масштабированными метриками IGRP (административное расстояние игнорируется).

Теперь давайте посмотрим, как выглядят эти моменты, о которых следует помнить, на рисунке 11:

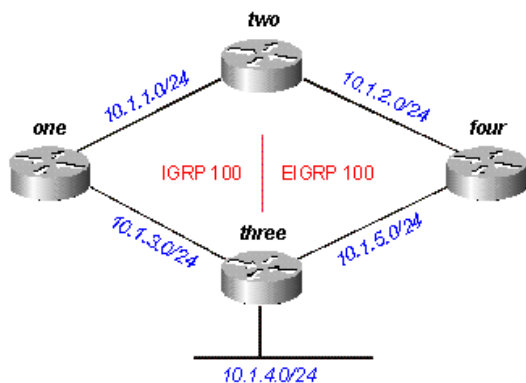


Figure 11

Маршрутизатор 1 объявляет маршрут 10.1.4.0/24 в автономной IGRP-системе 100. Маршрутизатор 4 объявляет маршрут 10.1.4.0/24 в автономной EIGRP-системе 100 в качестве внешнего. На маршрутизаторе 2 в автономной системе 100 выполняется как протокол EIGRP, так и IGRP.

Если мы игнорируем маршрут EIGRP, объявленный маршрутизатором 4 (например, посредством отключения канала между маршрутизаторами 2 и 4), то на маршрутизаторе 2 отображается следующее сообщение:

```
two# show ip route 10.1.4.0
Routing entry for 10.1.4.0/24
  Known via "igrp 100", distance 100, metric 12001
  Redistributing via igrp 100, eigrp 100
  Advertised by igrp 100 (self originated)
    eigrp 100
  Last update from 10.1.1.2 on Serial11, 00:00:42 ago
  Routing Descriptor Blocks:
  * 10.1.1.2, from 10.1.1.2, 00:00:42 ago, via Serial11
    Route metric is 12001, traffic share count is 1
    Total delay is 20010 microseconds, minimum bandwidth is 1000 Kbit
    Reliability 1/255, minimum MTU 1 bytes
    Loading 1/255, Hops 0
```

Обратите внимание, административное расстояние равно 100. При добавлении EIGRP-маршрута, на маршрутизаторе 2 будет отображено следующее:

```
two# show ip route 10.1.4.0
Routing entry for 10.1.4.0/24
  Known via "eigrp 100", distance 170, metric 3072256, type external
  Redistributing via igrp 100, eigrp 100
  Last update from 10.1.2.2 on Serial0, 00:53:59 ago
  Routing Descriptor Blocks:
  * 10.1.2.2, from 10.1.2.2, 00:53:59 ago, via Serial0
    Route metric is 3072256, traffic share count is 1
    Total delay is 20010 microseconds, minimum bandwidth is 1000 Kbit
    Reliability 1/255, minimum MTU 1 bytes
```

Обратите внимание, что после масштабирования из IGRP в EIGRP метрики для этих маршрутов одинаковы (см. раздел Метрики):

- $12001 \times 256 = 3072256$

где 12001 (метрика IGRP) – это метрика маршрута через маршрутизатор 1, а 3072256 (метрика EIGRP) – это метрика через маршрутизатор 4.

После выполнения масштабирования второй маршрутизатор предпочитает внешний маршрут EIGRP, имеющий такую же метрику и большее административное расстояние. Это справедливо для всех случаев автоматического перераспределения между EIGRP и IGRP в одной автономной системе. Маршрутизатор всегда выбирает путь с наименьшей стоимостью метрики и игнорирует административное расстояние.

### Перераспределение в другие протоколы и из них

Перераспределение между EIGRP и другими протоколами (например, RIP и OSPF) работает так же, как и любое другое перераспределение. При выполнении перераспределения между протоколами рекомендуется всегда использовать метрику по умолчанию. При перераспределении между EIGRP и другими протоколами необходимо помнить о следующих двух проблемах:

- Маршруты, перераспределенные в EIGRP не всегда суммируются (более подробно об этом см. раздел Суммирование).
- Внешние маршруты EIGRP имеют административное расстояние 170.

### Перераспределение статических маршрутов на интерфейсы

При установке статического маршрута на интерфейс и настройке инструкции сети с использованием **EIGRP-маршрутизатора**, который включает в себя статический маршрут, EIGRP перераспределяет этот маршрут так, как если бы это был интерфейс, подключенный напрямую. Рассмотрим сеть на рисунке 12:

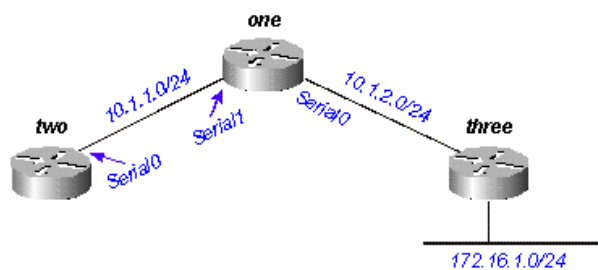


Figure 12

Маршрутизатор 1 имеет в сеть 172.16.1.0/24 статический маршрут, настроенный через интерфейс Serial 0:

```
ip route 172.16.1.0 255.255.255.0 Serial0
```

Маршрутизатор 1 также имеет инструкцию сети для данного пункта этого статического маршрута:

```
router eigrp 2000
network 10.0.0.0
network 172.16.0.0
no auto-summary
```

Маршрутизатор 1 перераспределяет этот маршрут, даже если это не перераспределяемые статические маршруты, поскольку EIGRP воспринимает это как непосредственно подключенную сеть. Для первого маршрутизатора это выглядит так:

```
two# show ip route
.....
 10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.1.1.0/24 is directly connected, Serial0
D       10.1.2.0/24 [90/2169856] via 10.1.1.1, 00:00:47, Serial0
       172.16.0.0/24 is subnetted, 1 subnets
D       172.16.1.0 [90/2169856] via 10.1.1.1, 00:00:47, Serial0
```

Обратите внимание, что маршрут в сеть 172.16.1.0/24 выглядит на маршрутизаторе 2 как внутренний EIGRP-маршрут.

## Суммирование

В EIGRP существуют два вида суммирования: автоматическое суммирование и суммирование вручную

### Автоматическое суммирование

EIGRP выполняет автосуммирование всякий раз, когда он пересекает границу между двумя различными крупными сетями. Например, на рисунке 13 маршрутизатор 2 объявляет маршрутизатору 1 только сеть 10.0.0.0/8, потому что интерфейс, который использует маршрутизатор 2, чтобы добраться до маршрутизатора 1, находится в другой крупной сети.

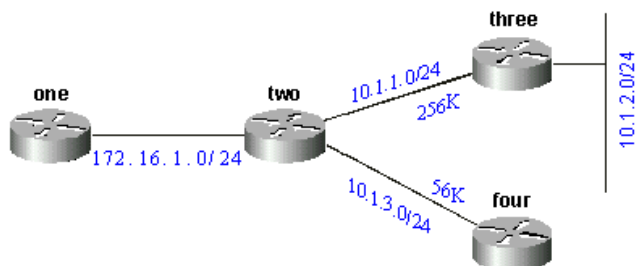


Figure 13

Для первого маршрутизатора это выглядит так:

```
one# show ip eigrp topology 10.0.0.0
IP-EIGRP topology entry for 10.0.0.0/8
State is Passive, Query origin flag is 1, 1 Successor(s), FD is 11023872
Routing Descriptor Blocks:
 172.16.1.1 (Serial0), from 172.16.1.2, Send flag is 0x0
   Composite metric is (11023872/10511872), Route is Internal
   Vector metric:
     Minimum bandwidth is 256 Kbit
     Total delay is 40000 microseconds
     Reliability is 255/255
     Load is 1/255
     Minimum MTU is 1500
     Hop count is 1
```

Этот маршрут не помечается как суммарный маршрут: он выглядит как внутренний маршрут. Данная метрика – это лучшая метрика из всех суммарных маршрутов. Обратите внимание, что минимальная пропускная способность на этом маршруте равна 256 Кбит, несмотря на то, что в сети 10.0.0.0 существуют каналы с пропускной способностью равной 56 Кбит.

На маршрутизаторе, выполняющем суммирование, для суммарного адреса создается маршрут до null0:

```
two# show ip route 10.0.0.0
Routing entry for 10.0.0.0/8, 4 known subnets
Attached (2 connections)
Variably subnetted with 2 masks
Redistributing via eigrp 2000

C      10.1.3.0/24 is directly connected, Serial2
D      10.1.2.0/24 [90/10537472] via 10.1.1.2, 00:23:24, Serial1
D      10.0.0.0/8 is a summary, 00:23:20, Null0
C      10.1.1.0/24 is directly connected, Serial1
```

Маршрут к сети 10.0.0.0/8 помечается как суммарный через Null0. Запись, размещенная в таблице топологии для данного суммарного маршрута, выглядит следующим образом:

```
two# show ip eigrp topology 10.0.0.0
IP-EIGRP topology entry for 10.0.0.0/8
State is Passive, Query origin flag is 1, 1 Successor(s), FD is 10511872
Routing Descriptor Blocks:
 0.0.0.0 (Null0), from 0.0.0.0, Send flag is 0x0
   (note: the 0.0.0.0 here means this route is originated by this router)
Composite metric is (10511872/0), Route is Internal
Vector metric:
  Minimum bandwidth is 256 Kbit
  Total delay is 20000 microseconds
  Reliability is 255/255
  Load is 1/255
  Minimum MTU is 1500
  Hop count is 0
```

Чтобы маршрутизатор 2 начал объявлять компоненты сети 10.0.0.0 вместо предоставления сводки, в процессе EIGRP маршрутизатора 2 необходимо настроить **no auto-summary** (запрет на автосводку) :

## На маршрутизаторе 2

```
router eigrp 2000
network 172.16.0.0
network 10.0.0.0
no auto-summary
```

Теперь при выключенной автосводке маршрутизатор 1 видит все компоненты сети 10.0.0.0:

```
one# show ip eigrp topology
IP-EIGRP Topology Table for process 2000

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status

P 10.1.3.0/24, 1 successors, FD is 46354176
   via 20.1.1.1 (46354176/45842176), Serial0
P 10.1.2.0/24, 1 successors, FD is 11049472
   via 20.1.1.1 (11049472/10537472), Serial0
P 10.1.1.0/24, 1 successors, FD is 11023872
   via 20.1.1.1 (11023872/10511872), Serial0
P 172.16.1.0/24, 1 successors, FD is 2169856
   via Connected, Serial0
```

Некоторые моменты, которые необходимо учитывать при суммировании внешних маршрутов, рассмотрены ниже в разделе "Автоматическое суммирование внешних маршрутов".

## Ручное суммирование

EIGRP позволяет суммировать внутренние и внешние маршруты фактически на любой битовой границе с помощью функции

суммирования маршрутов вручную. Например, на рис. 14 показано, как маршрутизатор 2 суммирует 192.1.1.0/24, 192.1.2.0/24 и 192.1.3.0/24 в блок CIDR 192.1.0.0/22.

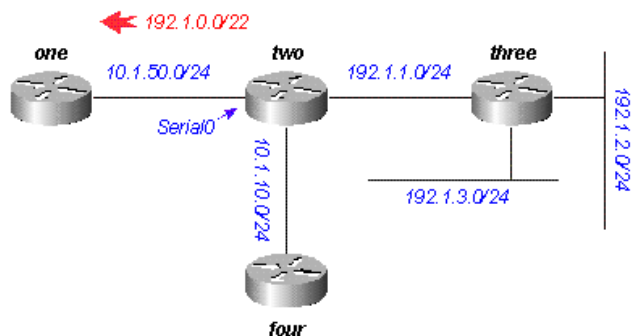


Figure 14

Ниже приведена конфигурация маршрутизатора 2:

```
two# show run
....
!
interface Serial0
ip address 10.1.50.1 255.255.255.0
ip summary-address eigrp 2000 192.1.0.0 255.255.252.0
no ip mroute-cache
!
....

two# show ip eigrp topology
IP-EIGRP Topology Table for process 2000

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status

P 10.1.10.0/24, 1 successors, FD is 45842176
   via Connected, Loopback0
P 10.1.50.0/24, 1 successors, FD is 2169856
   via Connected, Serial0
P 192.1.1.0/24, 1 successors, FD is 10511872
   via Connected, Serial1
P 192.1.0.0/22, 1 successors, FD is 10511872
   via Summary (10511872/0), Null0
P 192.1.3.0/24, 1 successors, FD is 10639872
   via 192.1.1.1 (10639872/128256), Serial1
P 192.1.2.0/24, 1 successors, FD is 10537472
   via 192.1.1.1 (10537472/281600), Serial1
```

Обратите внимание на команду **ip summary-address** в интерфейсе Serial0, а также на суммарный маршрут через Null0. На маршрутизаторе 1 это отображается как внутренний маршрут:

```
one# show ip eigrp topology
IP-EIGRP Topology Table for process 2000

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status

P 10.1.10.0/24, 1 successors, FD is 46354176
   via 10.1.50.1 (46354176/45842176), Serial0
P 10.1.50.0/24, 1 successors, FD is 2169856
   via Connected, Serial0
P 192.1.0.0/22, 1 successors, FD is 11023872
   via 10.1.50.1 (11023872/10511872), Serial0
```

## Автоматическое суммирование внешних маршрутов

EIGRP будет выполнять автосуммирование для внешних маршрутов в том случае, когда имеется компонент той же самой крупной

сети, который является внутренним маршрутом. В качестве примера рассмотрим рисунок 15.

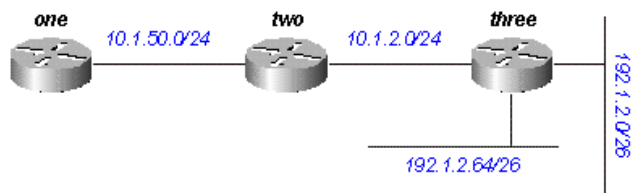


Figure 15

Маршрутизатор 3 внедряет (инжектирует) внешние маршруты в 192.1.2.0/26 и 192.1.2.64/26 в EIGRP используя для этого команду **redistribute connected** (см. конфигурационные листинги ниже).

### Маршрутизатор 3

```
interface Ethernet0
ip address 192.1.2.1 255.255.255.192
!
interface Ethernet1
ip address 192.1.2.65 255.255.255.192
!
interface Ethernet2
ip address 10.1.2.1 255.255.255.0
!router eigrp 2000
redistribute connected
network 10.0.0.0
default-metric 10000 1 255 1 1500
```

При данной конфигурации маршрутизатора 3 в маршрутной таблице на маршрутизаторе 1 будет отображено следующее:

```
one# show ip route
....
 10.0.0.0/8 is subnetted, 2 subnets
D       10.1.2.0 [90/11023872] via 10.1.50.2, 00:02:03, Serial0
C       10.1.50.0 is directly connected, Serial0
   192.1.2.0/26 is subnetted, 1 subnets
D EX   192.1.2.0 [170/11049472] via 10.1.50.2, 00:00:53, Serial0
D EX   192.1.2.64 [170/11049472] via 10.1.50.2, 00:00:53, Serial0
```

Несмотря на то, что обычно при автосуммировании маршрутизатор 3 суммирует маршруты 192.1.2.0/26 и 192.1.2.64/26 в один основной пункт назначения (192.1.2.0/24), в данном случае этого не произойдет, поскольку оба маршрута являются внешними. Однако, если перенастроить канал между маршрутизаторами 2 и 3 на адрес 192.1.2.128/26, а также добавить сетевые операторы для данной сети на маршрутизаторах 2 и 3, в этом случае автоматическое суммирование 192.1.2.0/24 сформируется на маршрутизаторе 2.

### Маршрутизатор 3

```
interface Ethernet0
ip address 192.1.2.1 255.255.255.192
!
interface Ethernet1
ip address 192.1.2.65 255.255.255.192
!
interface Serial0
ip address 192.1.2.130 255.255.255.192
!
router eigrp 2000
network 192.1.2.0
```

Теперь маршрутизатор 2 создает сводные данные для 192.1.2.0/24:

```
two# show ip route
....
D      192.1.2.0/24 is a summary, 00:06:48, Null0
....
```

Маршрутизатор 1 показывает только суммарный маршрут:

```
one# show ip route
....
10.0.0.0/8 is subnetted, 1 subnets
C      10.1.1.0 is directly connected, Serial0
D      192.1.2.0/24 [90/11023872] via 10.1.50.2, 00:00:36, Serial0
```

## Обработка и диапазон запросов

При обработке маршрутизатором запроса от соседа действуют следующие правила:

Источник запроса	Состояние маршрута	Действие
сосед (но не текущий преемник)	пассивный	ответить информацией о текущем преемнике
преемник	пассивный	попытаться найти нового преемника (в случае успеха выслать новую информацию; в случае неудачи пометить пункт назначения как недостижимый и опросить всех соседей, кроме предыдущего преемника)
любой сосед	до опроса путь через данного соседа отсутствовал	отправить ответ с указанием наилучшего пути, известного на данный момент
любой сосед	до запроса не был известен	отправить ответ с указанием того, что пункт назначения недостижим
сосед (но не текущий преемник)	активный	при отсутствии текущего преемника для данного пункта назначения (обычно этот так и бывает) ответить, что данный адрес недостижим
		при наличии подходящего преемника отправить информацию о текущем пути
преемник	активный	попытаться найти нового преемника (в случае успеха выслать новую информацию; в случае неудачи пометить пункт назначения как недостижимый и опросить всех соседей, кроме предыдущего

Действия, описанные в приведенной выше таблице, влияют на диапазон запросов в сети, т.е. диапазон определяется количеством маршрутизаторов, которые примут запрос и ответят на него, прежде чем сеть выполнит сходимость для адаптации к новой топологии. Чтобы понять, каким образом эти правила влияют на способ обработки запросов, посмотрим на схему сети на рисунке 16 (данная сеть работает в штатном режиме).

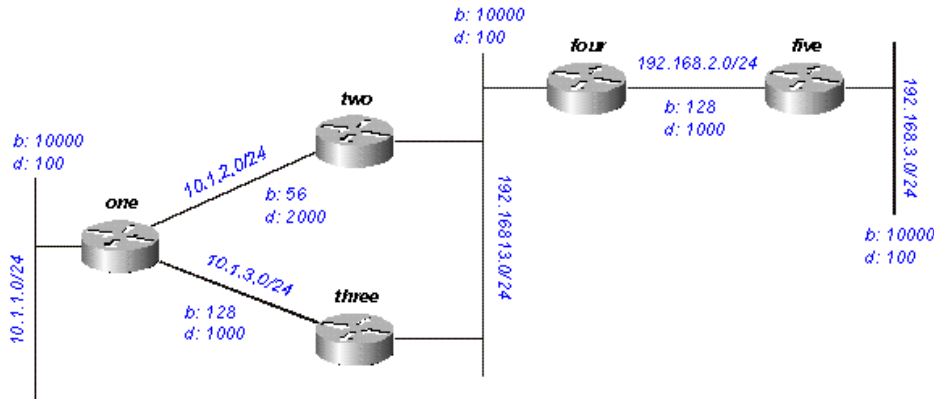


Figure 16

Можно ожидать, с сетью 192.168.3.0/24 (крайняя правая сторона) произойдет следующее:

- Маршрутизатор 1 имеет два маршрута к адресу 192.168.3.0/24:
  - через маршрутизатор 2 (расстояние – 46533485 и объявленное расстояние – 20307200)
  - через маршрутизатор 3 (расстояние – 20563200 и объявленное расстояние – 20307200)
- Маршрутизатор 1 выбирает путь через маршрутизатор 3 и рассматривает путь через маршрутизатор 2 как возможного преемника
- Маршрутизаторы 2 и 3 указывают один путь к сети 192.168.3.0/24 – через маршрутизатор 4

Представим себе, что в сети 192.168.3.0/24 происходит сбой. Что же в этом случае будет происходить данной сети? Рисунки с 16а по 16h иллюстрируют данный процесс.

Маршрутизатор 5 помечает сеть 192.168.3.0/24 как недостижимую и отправляет запрос маршрутизатору 4:

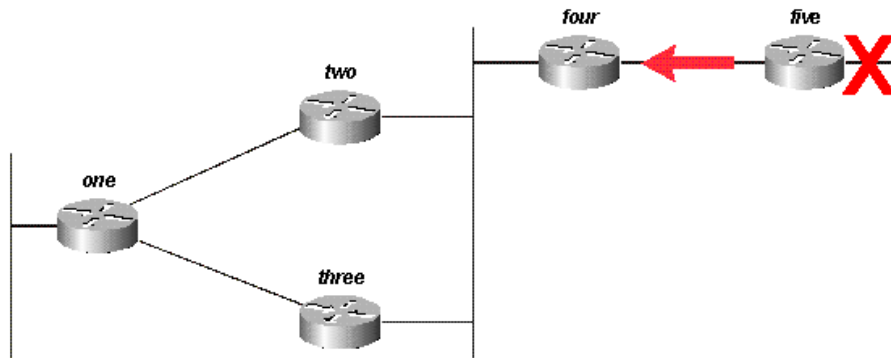


Figure 16a

Получив запрос от своего преемника, маршрутизатор 4 попытается найти нового подходящего преемника для этой сети. Если таковой не будет обнаружен, тогда он пометит сеть 192.168.3.0/24 как недостижимую и отправит запросы маршрутизаторам 2 и 3:



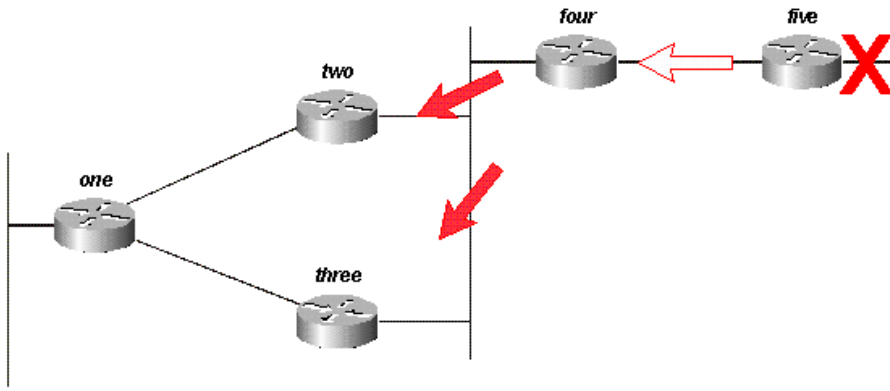


Figure 16b

В свою очередь маршрутизаторы 2 и 3 обнаружат, что они утратили единственно возможный маршрут в сеть 192.168.3.0/24. Соответственно с этим они пометят данный адрес как недостижимый и направят запросы маршрутизатору 1:

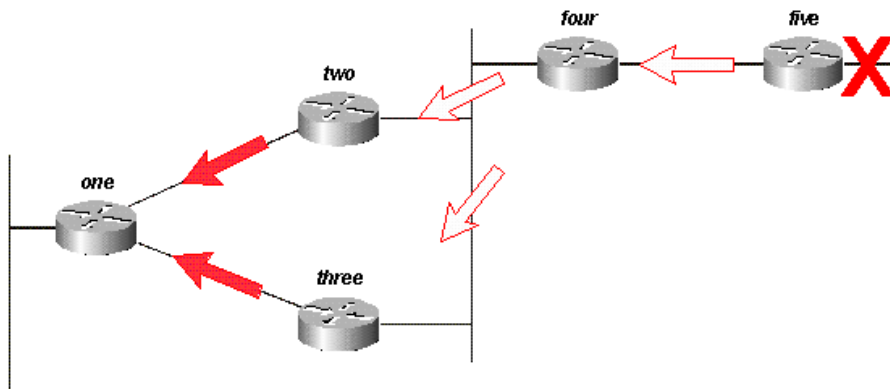


Figure 16c

Для простоты предположим, что маршрутизатор 1 первым получит запрос от маршрутизатора 3 и пометит маршрут как недостижимый. Затем маршрутизатор 1 получает запрос от маршрутизатора 2. Возможна и другая последовательность, но конечный результат все равно будет такой же.

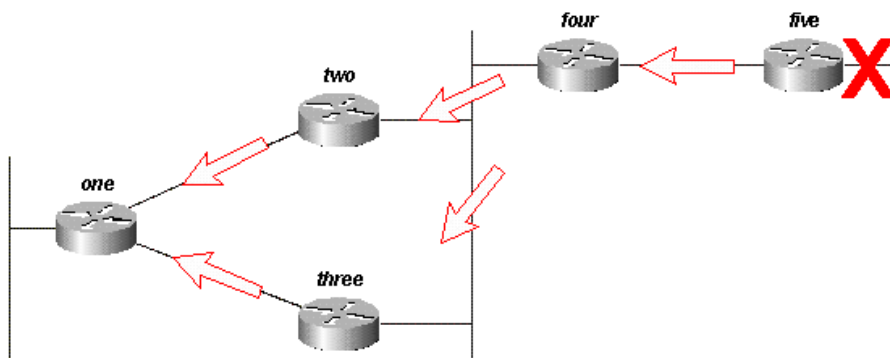


Figure 16d

Маршрутизатор один отвечает на оба запроса откликами "недостижим" и в настоящий момент находится в пассивном режиме для 192.168.3.0/24:

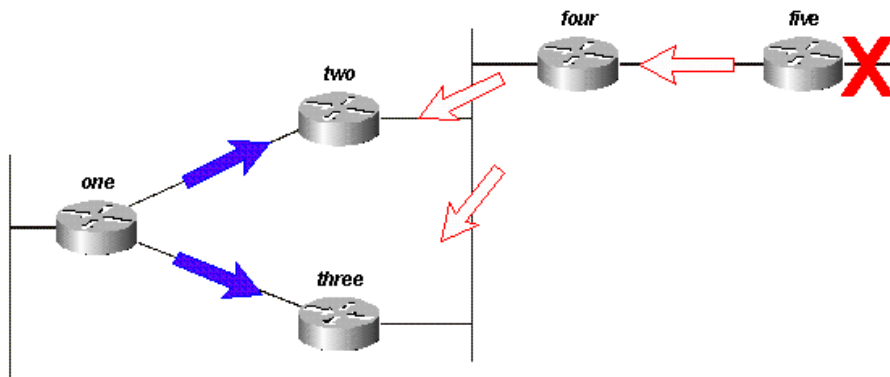


Figure 16e

Маршрутизаторы 2 и 3 отвечают на запрос маршрутизатора 4 и в настоящий момент находятся в пассивном режиме для 192.168.3.0/24:

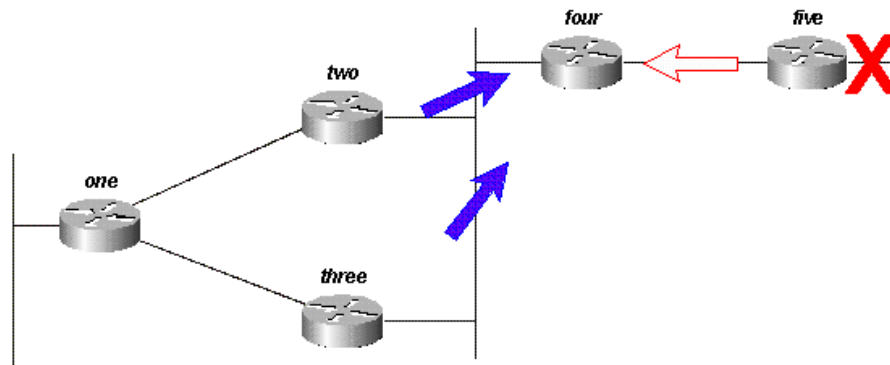


Figure 16f

Получив отклик от маршрутизатора 4, маршрутизатор 5 удалит из своей маршрутной таблицы запись о сети 192.168.3.0/24. В настоящий момент маршрутизатор 5 находится в пассивном режиме для сети 192.168.3.0/24. Он отправляет маршрутизатору 4 пакеты обновления, и таким образом маршрут удаляется из таблиц топологии и маршрутизации на всех оставшихся маршрутизаторах.

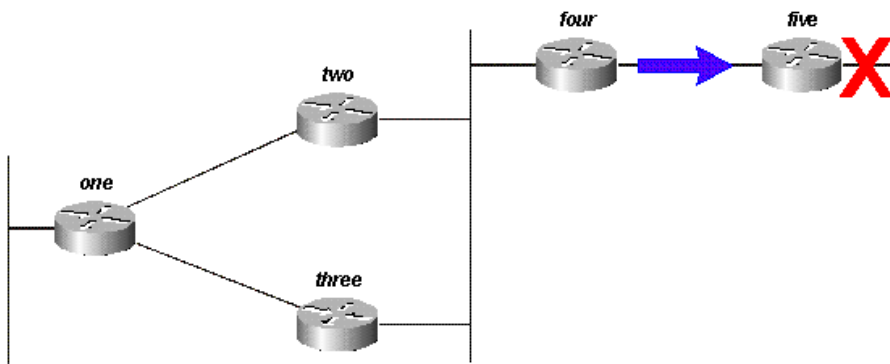


Figure 16g

Необходимо понимать, что несмотря на то, что в сети могут существовать другие пути для запросов и что запросы могут обрабатываться в другой последовательности, все имеющиеся в сети маршрутизаторы все равно будут обрабатывать запрос для сети 192.168.3.0/24 в случае, если этот канал будет нарушен. Может оказаться так, что некоторые маршрутизаторы будут обрабатывать не один, а большее количество запросов (в нашем примере это маршрутизатор 1). Если бы запросы действительно поступили на маршрутизаторы в другом порядке, тогда оказалось бы, что некоторые маршрутизаторы обрабатывали бы три или четыре запроса. Этот пример хорошо иллюстрирует неограниченный запрос в сети EIGRP.

### Каким образом точки суммирования влияют на диапазон запроса

Теперь рассмотрим маршруты в сеть 10.1.1.0/24 для той же самой топологии сети:

- В таблице топологии маршрутизатора 2 содержится запись для сети 10.1.1.0/24 со стоимостью равной 46251885 (через маршрутизатор 1).
- В таблице топологии маршрутизатора 3 содержится запись для сети 10.1.1.0/24 с ценой 20281600 (через маршрутизатор 1).
- В таблице топологии маршрутизатора 4 содержится запись для сети 10.0.0.0/8 (маршрутизаторы 2 и 3 выполняют автосуммирование к границе главной сети). Маршрут проходит через маршрутизатор 3 и имеет метрику 20307200 (поскольку объявленное расстояние через маршрутизатор 2 больше, чем суммарная метрика через маршрутизатор 3, маршрутизатор 2 не является возможным преемником).

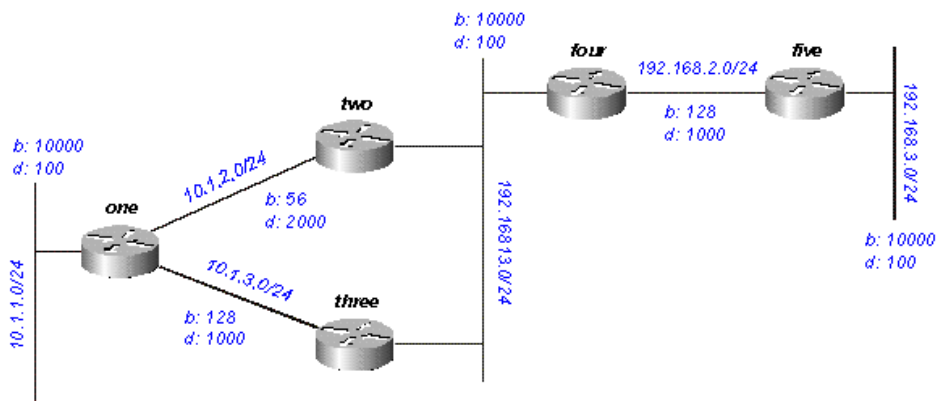


Figure 17

Если в сети 10.1.1.0/24 произойдет сбой, то маршрутизатор 1 пометит ее как недостижимую и запросит у всех соседних узлов (маршрутизаторы 2 и 3) новый путь к этой сети:

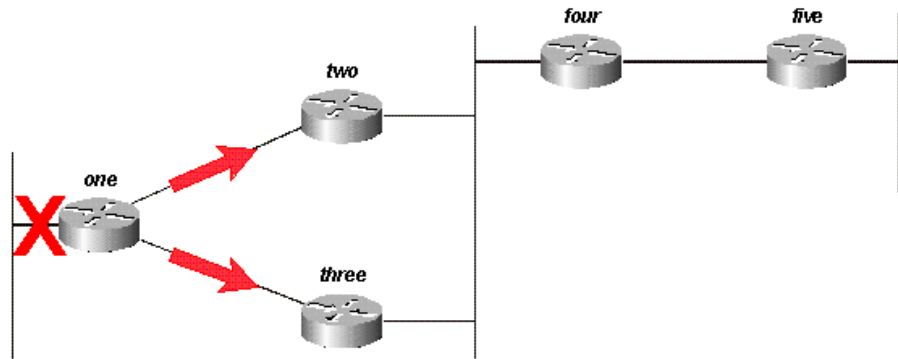


Figure 17a

Получив запрос от маршрутизатора 1, маршрутизатор 2 помечает маршрут как недостижимый (поскольку этот запрос был получен преемника), а затем запрашивает маршрутизаторы 4 и 3:

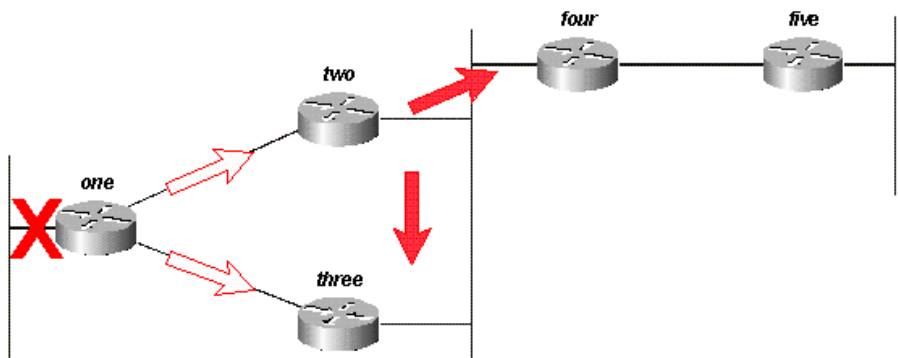


Figure 17b

Когда маршрутизатор 3 получает запрос от маршрутизатора 1, он помечает пункт назначения как недостижимый и отправляет запросы на маршрутизаторы 2 и 4:

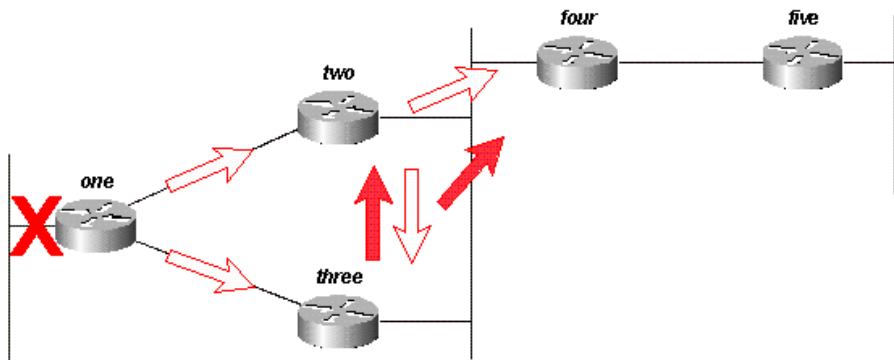


Figure 17c

Получив запросы от маршрутизаторов 2 и 3, маршрутизатор 4 отвечает, что маршрут 10.1.1.0/24 недостижим (заметьте, что маршрутизатор 4 не располагает данными об искомой подсети, так как в него загружены данные только по маршруту 10.0.0.0/8):

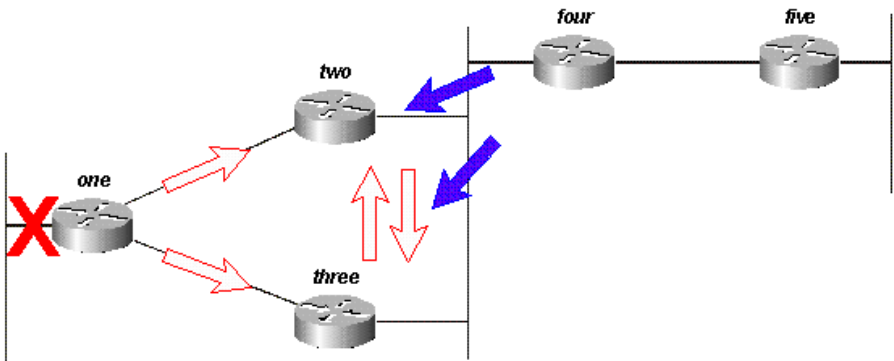


Figure 17d

Маршрутизаторы 2 и 3 сообщают друг другу о том, что 10.1.1.0/24 недоступен:

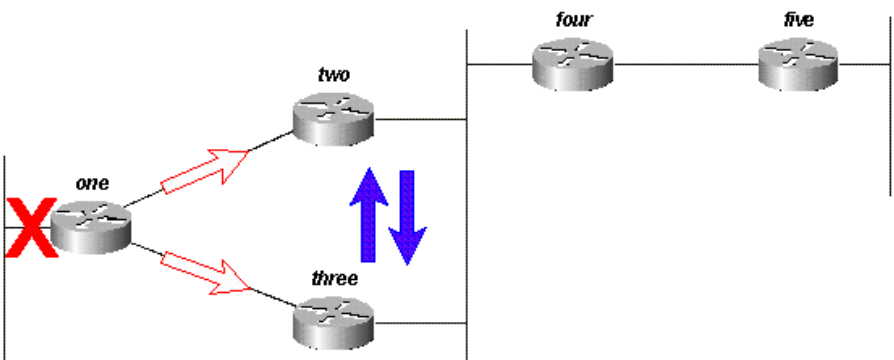


Figure 17e

Поскольку у маршрутизаторов 2 и 3 нет невыполненных запросов, они отвечают маршрутизатору 1, что 10.1.1.0/24 недоступен:

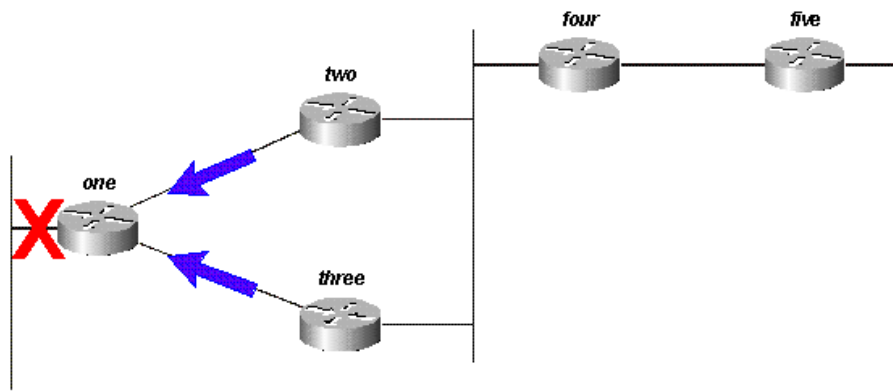


Figure 17f

В этом случае запрос ограничен автосуммированием на маршрутизаторах 2 и 3. Маршрутизатор 5 не участвует в процедуре отправки запросов и, соответственно, не участвует в процессе повторной сходимости. Запросы могут быть ограничены ручным суммированием, границами автономной системы и списками распространения.

### Каким образом границы автономной системы влияют на диапазон запроса

Если маршрутизатор повторно распределяет маршруты между двумя автономными системами EIGRP, то он отвечает на запрос в рамках обычных правил обработки и отправляет новый запрос в другую автономную систему. Например, если соединение с сетью, подключенной через маршрутизатор 3, прерывается, то маршрутизатор 3 помечает маршрут как недостижимый и запрашивает у маршрутизатора 2 новый путь:

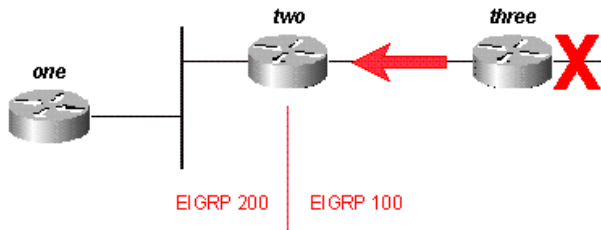


Figure 18a

Маршрутизатор 2 отвечает, что данная сеть недостижима, и инициирует запрос в автономную систему 200 в направлении маршрутизатора 1. Как только маршрутизатор 3 получает ответ на свой исходный запрос, он удаляет маршрут из таблицы. Маршрутизатор 3 теперь пассивен для этой сети:

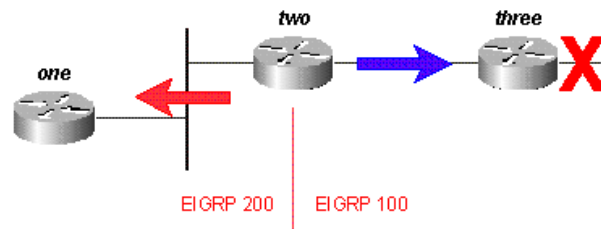


Figure 18b

Маршрутизатор 1 отвечает маршрутизатору 2, и маршрут переходит в пассивное состояние:

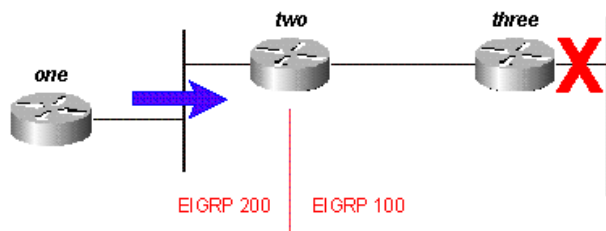


Figure 18c

Поскольку исходный запрос не распространялся по всей сети (он был ограничен автономными границами системы), то он попадает во вторую автономную систему в виде нового запроса. Такой метод позволит предотвратить возникновение проблем, связанных с зависанием в активном состоянии (stuck in active – SIA) в сети. Это достигается посредством ограничения количества маршрутизаторов, через которые запрос должен пройти, прежде чем на него будет дан ответ. Однако этот метод не позволяет решить проблему в целом, заключающуюся в том, что запрос должен обрабатываться каждым маршрутизатором. Фактически, применение данного метода ограничения запроса может только усугубить проблему из-за запрета на автосуммирование маршрутов, которые при других обстоятельствах были бы просуммированы (суммирование внешних маршрутов не выполняется за исключением тех случаев, когда в данной крупной сети присутствуют внешние компоненты).

### Каким образом списки распространения влияют на диапазон запроса

Вместо того чтобы блокировать распространение запроса, в EIGRP списки распространения помечают любой отклик на запрос как недостижимый. В качестве примера рассмотрим схему на рисунке 19.

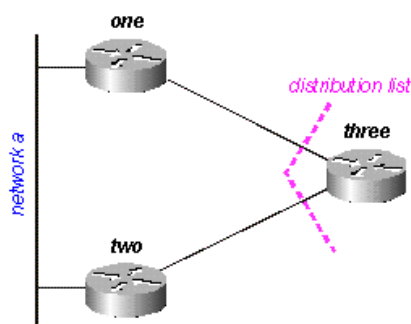


Figure 19

В этой схеме:

- На маршрутизаторе 3 к последовательным интерфейсам применен список распространения, который разрешает объявлять только маршрут в сеть В.
- Маршрутизаторы 1 и 2 не знают, что в сеть А можно попасть через маршрутизатор 3 (который не используется в качестве транзитной точки между маршрутизаторами 1 и 2).
- Маршрутизатор 3 использует маршрутизатор 1 в качестве предпочтительного пути в сеть А. и не использует маршрутизатор 2 в качестве возможного приемника.

При утрате связи с сетью А маршрутизатор 1 помечает этот маршрут как недостижимый и отправляет запрос маршрутизатору 3. Маршрутизатор 3 не объявляет путь в сеть А, поскольку к его последовательным портам был применен список распространения.

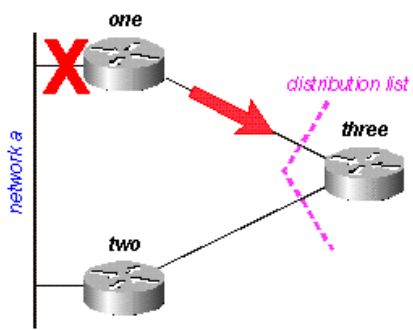


Figure 19a

Маршрутизатор 3 помечает этот маршрут как недоступный и затем отправляет запрос маршрутизатору 2:

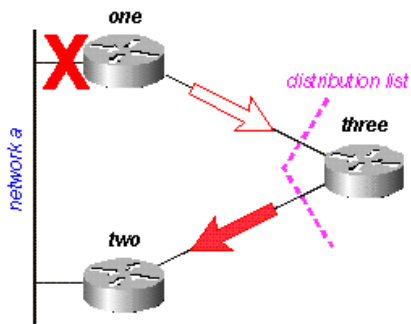


Figure 19b

Маршрутизатор 2 производит поиск в своей таблице топологии, в результате которого определяет, что у него есть разрешенное соединение с сетью А. Обратите внимание, что на запрос не влияет список рассылки в маршрутизаторе 3:

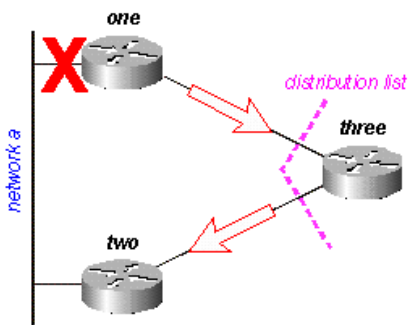


Figure 19c

Маршрутизатор 2 отвечает, что сеть А достижима. Теперь у маршрутизатора 3 есть разрешенный путь:

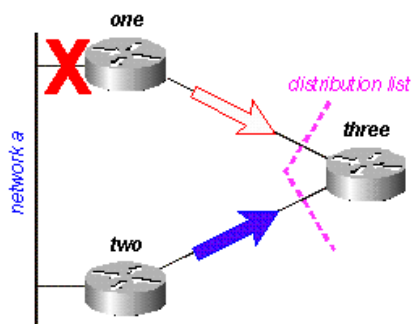
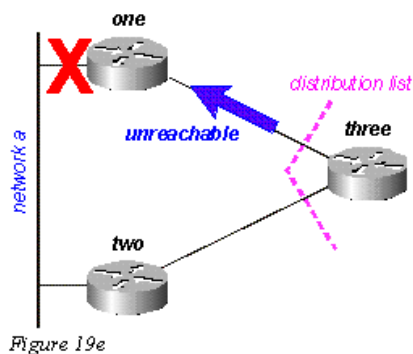


Figure 19d

Маршрутизатор 3 формирует ответ на запрос маршрутизатора 1, однако по причине действия списка распределения он отправляет отклик, в котором сообщает, что сеть A недоступна, даже при том, что у маршрутизатора 3 имеется допустимый маршрут в сеть A:



## Пошаговое продвижение пакетов

Некоторые протоколы маршрутизации в процессе выполнения конвергенции (адаптации к изменениям в сети) используют всю пропускную способность, которая имеется у каналов с низкой пропускной способностью. EIGRP позволяет избежать таких перегрузок за счет регулирования скорости передачи пакетов по сети. Таким образом, будет использоваться только часть имеющейся пропускной способности. Конфигурация EIGRP по умолчанию предусматривает использование до 50 процентов имеющейся пропускной способности. Однако это можно изменить, воспользовавшись следующей командой:

```
router(config-if)# ip bandwidth-percent eigrp 2 ?  
<1-999999> Maximum bandwidth percentage that EIGRP may use
```

В сущности, всякий раз, когда EIGRP ставит в очередь пакет, предназначенный для отправки через интерфейс, то для того чтобы определить задержку перед отправкой, он использует следующую формулу:

- $(8 * 100 * \text{размер пакета в байтах}) / (\text{пропускная способность в Кбит/с} * \text{процент от общей пропускной способности})$

Например, если EIGRP ставит в очередь пакет, размер которого равен 512 байтам и который подлежит отправке через последовательный интерфейс с пропускной способностью 56 Кбит/с, то задержка перед отправкой составляет:

- $(8 * 100 * 512 \text{ байт}) / (56000 \text{ бит/с} * 50\% \text{ от имеющейся пропускной способности}) = (8 * 100 * 512) / (56000 * 50) = 409600 / 2800000 = 0.1463 \text{ секунд}$

Таким образом пакеты (или группы пакетов), величина которых равна или превышает 512 байт, могут быть переданы по этому каналу до того, как EIGRP выполнит отправку своего пакета. Пошаговый таймер определяет время отправки пакета и обычно задается в миллисекундах. Для пакета, указанного выше, время задержки перед отправкой составляет 0,1463 секунды. В листинге команды **show ip eigrp interface** имеется поле, в котором указывается значение таймера регулирования скорости отправки:

```
router# show ip eigrp interface  
IP-EIGRP interfaces for process 2
```

Interface	Peers	Xmit Queue Un/Reliable	Mean SRTT	Pacing Time Un/Reliable	Multicast Flow Timer	Pending Routes
Se0	1	0/0	28	0/15	127	0
Se1	1	0/0	44	0/15	211	0

```
router#
```

Указанное значение представляет собой интервал, установленный для максимально допустимого блока данных (MTU), т.е. для самого большого пакета который может быть отправлен через данный интерфейс.



## Маршрутизация по умолчанию

В EIGRP маршрут по умолчанию можно создать двумя способами: 1) перераспределить статический маршрут; 2) выполнить суммирование в сеть 0.0.0.0/0. Используйте первый метод, если вам необходимо направить весь трафик, имеющий неизвестный пункт назначения, по маршруту по умолчанию, который находится в центральной части сети. Этот метод хорошо работает для объявления соединений с Интернетом. Например:

```
ip route 0.0.0.0 0.0.0.0 x.x.x.x (next hop to the internet)
!
router eigrp 100
 redistribute static
 default-metric 10000 1 255 1 1500
```

Статический маршрут, перенаправляемый в EIGRP, не обязательно должен перенаправляться в сеть 0.0.0.0. При использовании другой сети необходимо воспользоваться командой **ip default-network**, чтобы установить эту сеть как сеть по умолчанию. Более подробно см. "Настройка шлюза последней очереди".

Суммирование в маршрут по умолчанию эффективно только тогда, когда требуется обеспечить удаленные узлы маршрутом по умолчанию. Поскольку суммирование настраивается для каждого интерфейса в отдельности, вам не потребуется прибегать к использованию листов распространения или других механизмов для того, чтобы предотвратить распространение маршрута по умолчанию в центральную часть (ядро) сети. Обратите внимание, что суммирование на 0.0.0.0/0 замещает маршрут по умолчанию, полученный от любого другого протокола маршрутизации. Единственный способ настроить маршрут по умолчанию на маршрутизаторе с помощью этого метода – это установить статический маршрут 0.0.0.0/0 (начиная с версии 12.0(4)T программное обеспечение Cisco IOS позволяет настраивать административное расстояние в конце команды **summary-address**, таким образом локальное суммирование не будет отменять маршрут 0.0.0.0/0).

```
router eigrp 100
 network 10.0.0.0
 !
 interface serial 0
 encapsulation frame-relay
 no ip address
 !
 interface serial 0.1 point-to-point
 ip address 10.1.1.1
 frame-relay interface-dlci 10
 ip summary-address eigrp 100 0.0.0.0 0.0.0.0
```

## Балансировка нагрузки

Для четырех равноценных маршрутов в таблице маршрутизации протокол EIGRP указывает маршрутизатор, выравнивающий нагрузку. Тип балансировки нагрузки (по пакетам или по назначению) определяется типом коммутации, выполняемой на маршрутизаторе. EIGRP, однако, может распределять нагрузку по каналам с неодинаковой стоимостью.

**Примечание:** при помощи **max-paths** можно настроить EIGRP таким образом, что он будет использовать до шести равноценных путей.

Допустим, у нас имеется четыре пути к требуемому пункту назначения и эти пути имеют следующие метрики:

- путь 1: 1100
- путь 2: 1100
- путь 3: 2000
- путь 4: 4000

По умолчанию маршрутизатор направляет трафик как по пути 1, так и по пути 2. В протоколе EIGRP вы можете воспользоваться командой **variance**, чтобы указать маршрутизатору направлять трафик и по путям 3 и 4. Команда variance выполняет перемножение: трафик будет передан на любой канал, метрика которого меньше, чем наилучший путь, умноженный на дисперсию. Чтобы равномерно распределить нагрузку по путям 1, 2 используйте дисперсию равную 2, поскольку  $1100 \times 2 = 2200$ . Получающееся значение выше значения метрики через путь 3. Подобным же образом, для того чтобы добавить путь 4, используйте дисперсию равную 4 при команде **router eigrp**. Более подробно см. Как работает распределение нагрузки с неравной стоимостью путей (дисперсия) в IGRP и EIGRP?

Каким же образом маршрутизатор будет распределять трафик по этим путям? Значение метрики каждого пути будет поделено на наибольшую метрику. Полученный результат будет округлен до ближайшего целого числа, а затем это число будет использовано в качестве счетчика совместного трафика.

```
router# show ip route 10.1.4.0
Routing entry for 10.1.4.0/24
Known via "igrp 100", distance 100, metric 12001
Redistributing via igrp 100, eigrp 100
Advertised by igrp 100 (self originated)
    eigrp 100
Last update from 10.1.2.2 on Serial1, 00:00:42 ago
Routing Descriptor Blocks:
* 10.1.2.2, from 10.1.2.2, 00:00:42 ago, via Serial1
  Route metric is 12001, traffic share count is 1
  Total delay is 20010 microseconds, minimum bandwidth is 1000 Kbit
  Reliability 1/255, minimum MTU 1 bytes
  Loading 1/255, Hops 0
```

В нашем примере счетчики совместного трафика имеют следующие значения:

- для путей 1 и 2:  $4000/1100 = 3$
- для пути 3:  $4000/2000 = 2$
- для пути 4:  $4000/4000 = 1$

Таким образом, маршрутизатор отправит первые три пакета по пути 1, следующие три пакета – по пути 2, следующие два пакета – по пути 3, а затем следующие 4 пакета будут отправлены по пути 4. После этого маршрутизатор вернется в исходное положение и снова отправит первые три пакета по пути 1 и т.д.

**Примечание:** Даже при настроенной дисперсии EIGRP может не отправлять трафик по пути с неравной стоимостью. Это происходит в случае, если для данного конкретного маршрута объявленное расстояние превышает возможное расстояние. Более подробно см. в разделе "Возможное расстояние, объявленное расстояние и возможные приемники".

## Использование метрик

Если при первой настройке протокола EIGRP вы хотите воздействовать на его метрики, то вам необходимо придерживаться следующих базовых правил:

- Пропускная способность должна быть всегда равной фактической пропускной способности интерфейса (исключение составляют многоточечные последовательные каналы и другие ситуации, связанные с изменениями скорости из-за несовпадающих типов носителей).
- Задержку всегда следует использовать для воздействия на решения о маршрутизации, принимаемые протоколом EIGRP.

Поскольку для того чтобы определить скорость пересылки пакетов, EIGRP использует значение пропускной способности интерфейса, очень важно, чтобы это значение было задано правильно. Всегда используйте задержку, если вам требуется воздействовать на процесс выбора пути протоколом EIGRP.

При меньшей пропускной способности величина пропускной способности оказывает значительное воздействие на суммарную метрику; при более высокой пропускной способности задержка также оказывает значительное воздействие на суммарную метрику.

## Использование административных тегов в перераспределении

Внешние административные теги позволяют "разрывать" петли при обмене маршрутами между EIGRP и другими протоколами. При маркировке маршрута во время его перераспределения в EIGRP можно блокировать перераспределение из EIGRP во внешний протокол. Ниже мы приведем общий пример настройки этих тегов. Однако этот пример не демонстрирует всю процедуру настройки, которая используется для разрыва петель перераспределения.

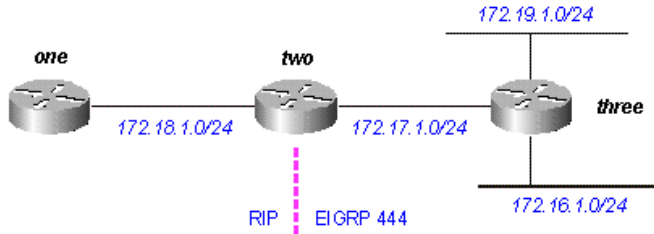


Figure 20

Маршрутизатор 3, распределяющий маршруты, связанные с EIGRP, выводит следующую информацию:

```
three# show run
....
interface Loopback0
ip address 172.19.1.1 255.255.255.0
!
interface Ethernet0
ip address 172.16.1.1 255.255.255.0
loopback
no keepalive
!
interface Serial0
ip address 172.17.1.1 255.255.255.0
....

router eigrp 444
 redistribute connected route-map foo
 network 172.17.0.0
 default-metric 10000 1 255 1 1500
....

access-list 10 permit 172.19.0.0 0.0.255.255
route-map foo permit 10
 match ip address 10
 set tag 1
....

three# show ip eigrp topo
IP-EIGRP Topology Table for process 444

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status

P 172.17.1.0/24, 1 successors, FD is 2169856
   via Connected, Serial0
   via Redistributed (2169856/0)
P 172.16.1.0/24, 1 successors, FD is 281600
   via Redistributed (281600/0)
P 172.19.1.0/24, 1 successors, FD is 128256, tag is 1
   via Redistributed (128256/0)
```

Маршрутизатор 2, перераспределяющий маршруты из EIGRP в RIP, выводит следующую информацию:

```
two# show run
```

```

....

interface Serial0
ip address 172.17.1.2 255.255.255.0
!
interface Serial1
ip address 172.18.1.3 255.255.255.0

....

router eigrp 444
network 172.17.0.0
!
router rip
redistribute eigrp 444 route-map foo
network 10.0.0.0
network 172.18.0.0
default-metric 1
!
no ip classless
ip route 1.1.1.1 255.255.255.255 Serial0
route-map foo deny 10
match tag 1
!
route-map foo permit 20

....

two# show ip eigrp topo
IP-EIGRP Topology Table for process 444

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status

P 172.17.1.0/24, 1 successors, FD is 2169856
   via Connected, Serial0
P 172.16.1.0/24, 1 successors, FD is 2195456
   via 172.17.1.1 (2195456/281600), Serial0
P 172.19.1.0/24, 1 successors, FD is 2297856, tag is 1
   via 172.17.1.1 (2297856/128256), Serial0

```

Обратите внимание на тег 1 на 172.19.1.0/24.

Маршрутизатор 1, который получает маршруты RIP, перенаправленные маршрутизатором 2, выводит следующую информацию:

```

one# show run

....

interface Serial0
ip address 172.18.1.2 255.255.255.0
no fair-queue
clockrate 1000000

router rip
network 172.18.0.0

....

one# show ip route

Gateway of last resort is not set

R    172.16.0.0/16 [120/1] via 172.18.1.3, 00:00:15, Serial0
R    172.17.0.0/16 [120/1] via 172.18.1.3, 00:00:15, Serial0
     172.18.0.0/24 is subnetted, 1 subnets
C        172.18.1.0 is directly connected, Serial0

```

Обратите внимание, что 172.19.1.0/24 отсутствует.

## Понимание информации, выводимой в листинге команд EIGRP

**show ip eigrp topology**

## show ip eigrp topology

Данная команда отображает только возможных преемников. Для отображения всех записей в таблице топологии используйте команду **show ip eigrp topology all-links**. Пояснения к каждой строке листинга приводятся после таблицы.

### show ip eigrp topology

```
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status

A 10.2.4.0/24, 0 successors, FD is 512640000, tag is 0x0, Q
    1 replies, active 00:00:01, query-origin: Local origin
        via 10.1.2.2 (Infinity/Infinity), r, Q, Serial1
    Remaining replies:
        via 10.1.1.2, r, Serial0

P 10.3.9.0/24, 1 successors, FD is 512640000, U
    * via 10.1.2.2 (512640000/128256), Serial1
```

## Объяснение конфигурационных параметров

**A** означает "активный". Также может отображаться "P", что означает "пассивный".

**10.2.4.0/24** – адрес назначения или маска.

**0 successors** указывает на количество преемников (или путей), имеющихся для данного адреса назначения; если слово "successors" написано заглавными буквами, это означает, что маршрут находится в переходном режиме.

**FD is 512640000** указывает возможное расстояние (feasible distance – FD). Данное значение может представлять собой либо наилучшую метрику для достижения данного конечного адреса либо наилучшую метрику, известную на момент, когда маршрут стал активным.

**tag is 0x0** может устанавливаться и/или фильтроваться при помощи карт маршрутов (команды **set tag** и **match tag**).

**Q** означает, что запрос ожидает обработки. В данном поле также может быть указано: "U" (ожидание обновления); "R" (ожидание ответа).

**1 replies** указывает количество откликов/запросов, ожидающих обработки.

**active 00:00:01** показывает время, в течение которого данный путь является активным.

**query origin: Local origin** указывает на то, что источником запроса (local origin) является данный маршрут. В данном поле также может быть указано: Multiple origins (множественные источники) – это означает, что запросы на данный адрес были отправлены несколькими соседними узлами (но не преемником); Successor origin (источник запроса: преемник) – означает, что преемник является источником запроса.

**via 10.1.2.2** указывает, что данный маршрут был получен от соседнего узла с IP-адресом – 10.1.2.2. В данном поле также может быть указано: Connected (подключен) – если сеть напрямую подключена к этому маршрутизатору; Redistributed (перераспределен) – если этот маршрут перераспределяется в EIGRP на данном маршрутизаторе; Summary (суммарный) – если данный маршрут является суммарным маршрутом, созданным на данном маршрутизаторе.

**(Infinity/Infinity)** указывает метрику, необходимую для достижения этого пути через соседа, указанного в первом поле, а также объявленное расстояние через соседа, указанного во втором поле.

**r** указывает на то, что данному соседу был отправлен запрос и что ответ ожидается.

**Q** – это флаг посылки для этого маршрутизатора, означающий, что имеется запрос, ожидающий обработки. В данном поле также может быть указано: "U" (ожидание обновления); "R" (ожидание ответа).

**Serial1** – это интерфейс, через который можно получить доступ к этому соседнему узлу.

**Via 10.1.1.2** указывает соседа, от которого ожидается ответ.

**r** указывает, что соседу был отправлен запрос о маршруте и что ответ пока не был получен.

**Serial0** – это интерфейс, через который можно получить доступ к этому соседнему узлу.

**Via 10.1.2.2 (512640000/128256), Serial1** указывает, что используется этот маршрут (указывает, какой путь выберет следующий путь/назначение при наличии нескольких равноценных маршрутов).

### **show ip eigrp topology <network>**

Эта команда отображает все записи в таблице топологии для этого адреса назначения, а не только возможных преемников. Пояснения к каждой строке листинга приводятся после таблицы.

#### **show ip eigrp topology network**

```
IP-EIGRP topology entry for 20.0.0.0/8
State is Passive, Query origin flag is 1, 2 Successor(s), FD is 307200
Routing Descriptor Blocks:
 10.1.1.2 (Ethernet1), from 10.1.1.2, Send flag is 0x0
   Composite metric is (307200/281600), Route is Internal
   Vector metric:
     Minimum bandwidth is 10000 Kbit
     Total delay is 2000 microseconds
     Reliability is 0/255
     Load is 1/255
     Minimum MTU is 1500
     Hop count is 2
 10.1.2.2 (Ethernet0), from 10.1.2.2, Send flag is 0x0
   Composite metric is (307200/281600), Route is Internal
   Vector metric:
     Minimum bandwidth is 10000 Kbit
     Total delay is 2000 microseconds
     Reliability is 0/255
     Load is 1/255
     Minimum MTU is 1500
     Hop count is 2
```

### **Объяснение конфигурационных параметров**

**State is Passive** означает, что сеть находится в пассивном состоянии (другими словами, поиск маршрута в данную сеть в настоящий момент не осуществляется). В стабильной сети маршруты почти всегда находятся в пассивном состоянии.

**Query origin flag is 1.** Если этот маршрут активен, то в данном поле будет указан инициатор (источник) запроса.

- 0: данный маршрут является активным, но для него не был сформирован запрос (поиск возможного преемника осуществляется локально).
- 1: данный маршрутизатор сформировал запрос для данного маршрута (или маршрут является пассивным).

- 2: для данного запроса выполняются множественные диффузионные расчеты. Этот маршрутизатор получил более одного запроса на данный маршрут от нескольких источников.
- 3: маршрутизатор, от которого был получен маршрут в эту сеть, запрашивает другой маршрут.
- 4: множество источников запроса для данного маршрута, включая маршрутизатор, с которого был получен этот маршрут. Имеет сходство с 2, но также означает наличие строки источника запроса, которая описывает запросы, ожидающие выполнения для данного пути.

**2 Successor(s)** означает, что существуют два возможных пути в эту сеть.

**FD is 307200** указывает лучшую текущую метрику для этой сети. Если маршрут активен, то данная метрика является метрикой пути, которая использовалась ранее для маршрутизации пакетов в эту сеть.

**Routing Descriptor Blocks** Каждая из последующих записей характеризует один путь в эту сеть.

- **10.1.1.2 (Ethernet1)** – это следующий сегмент сети, а также интерфейс, через который можно получить доступ в следующий сегмент сети.
- **from 10.1.2.2** - источник данной информации о пути.
- **Send flag is:**
  - *0x0*: при наличии пакетов, которые необходимо отправить в связи с данной записью, данный параметр указывает тип пакета.
  - *0x1*: этот маршрутизатор получил запрос на эту сеть и нуждается в отправке одноадресного ответа.
  - *0x2*: данный маршрут активен, и следует разослать пакеты очереди многоадресной рассылки.
  - *0x3*: данный маршрут изменился, и следует разослать пакеты многоадресной рассылки.

**Composite metric is (307200/281600)** отображает суммарные расходы до этой сети. В скобках: первое число – это суммарные расходы до этой сети при использовании этого пути, включая расходы до следующего узла. Второе число – это объявленное расстояние, или, другими словами, расходы маршрутизатора следующего узла.

**Route is Internal** означает, что этот маршрут был изначально создан внутри этой автономной системы (AS) EIGRP. Если маршрут был перераспределен в EIGRP AS, то в данном поле будет указано, что маршрут является внешним (External).

**Vector metric** показывает индивидуальные метрики, используемые EIGRP для расчета расходов в какую-либо сеть. EIGRP не распространяет по всей сети информацию об общих расходах. Вместо этого распространяются векторные метрики, и каждый маршрутизатор самостоятельно рассчитывает расходы и объявленное расстояние.

- **Minimum bandwidth is 10000 Kbit** показывает самую низкую пропускную способность на пути к этой сети.
- **Total delay is 2000 microseconds** указывает сумму всех задержек на пути в данную сеть.
- **Reliability is 0/255** указывает коэффициент надежности. Данное число рассчитывается динамически, но при этом оно не является числом, используемым по умолчанию при расчете метрик.
- **Load is 1/255** указывает величину нагрузки на линию. Это значение вычисляется динамически, но не используется по умолчанию при расчете протоколом EIGRP стоимости использования пути.
- **Minimum MTU is 1500** Данное поле не используется для расчета метрик.
- **Hop count is 2** Данное поле не используется при расчете метрик. Однако указанное значение ограничивает максимальный размер EIGRP AS. Максимальное количество переходов, принимаемое EIGRP по умолчанию, составляет 100, хотя максимальное число переходов в соответствии с метрикой при настройке может достигать 220.

Для внешних маршрутов (External Route) добавляется приведенная ниже информация. Пояснения к каждой строке листинга приводятся после таблицы.

### External Route

```
External data:  
  Originating router is 10.1.2.2  
  AS number of route is 0  
  External protocol is Static, external metric is 0  
  Administrator tag is 0 (0x00000000)
```

#### Объяснение конфигурационных параметров

**Originating Router** указывает маршрутизатор, который включил данный маршрут в EIGRP AS.

**External AS** указывает автономную систему (AS), являющуюся источником этого маршрута (если таковая имеется).

**External Protocol** указывает, от какого протокола (если таковой имеется) поступил данный маршрут.

**external metric** указывает внутреннюю метрику во внешнем протоколе.

**Administrator Tag** может быть установлен и/или отфильтрован с помощью карт маршрутов (команды **set tag** и **match tag**).

#### **show ip eigrp topology [active | pending | zero-successors]**

Формат листинга такой же, как и у команды **show ip eigrp topology**, но при этом в данном листинге отображается часть таблицы топологии.

#### **show ip eigrp topology all-links**

Формат листинга такой же, как и у команды **show ip eigrp topology**, но при этом в данном листинге указываются все каналы, имеющиеся в таблице топологии (а не только возможные преемники).

## Дополнительная информация

- [Страничка поддержки EIGRP \(материалы на английском\)](#)
- [Справочник по командам EIGRP](#)
- [Страница поддержки IP-маршрутизации](#)
- [Техническая поддержка – Cisco Systems](#)