Введение в IGRP

Содержание

Введение

Этот документ содержит вводные сведения о протоколе внутреннего шлюза. У него две цели. Дается введение в технологию IGRP для тех, кто интересуется ее использованием, оценивает возможность ее внедрения или внедряет ее. Кроме того, более широко освещаются некоторые существенные соображения и концепции, реализованные в IGRP. Способ настройки IGRP описан в документах Настройка IGRP, Внедрение Cisco IGRP и Команды IGRP.

Задачи протокола IGRP

Протокол IGRP позволяет многим шлюзам координировать свою маршрутизацию. Он ставит следующие цели:

- Стабильная маршрутизация даже в больших или составных сетях. Никакие циклы маршрутизации не должны происходить, как раз когда переходные процессы.
- Быстрая реакция на изменения сетевой топологии.

- Низкие накладные расходы. Таким образом, IGRP не должен использовать большую пропускную способность, чем необходимо для его задачи.
- Разбиение трафика на несколько параллельных маршрутов имеющих примерно одинаковую желательность.
- Учет уровня ошибок и уровня трафика на различных путях.

Текущая реализация IGRP обеспечивает маршрутизацию для TCP/IP. Однако основы проектирования предназначены, чтобы быть в состоянии обработать множество протоколов.

Никакое программное средство не переходит, решают все проблемы маршрутизации. Обычно проблему маршрутизации можно разделить на несколько элементов. Протоколы, такие как IGRP называют "протоколами внутреннего шлюза" (IGPs). Их предполагается использовать в едином наборе сетей, с единым или тесно связанным управлением. Такие группы сетей соединяются с помощью протоколов внешней маршрутизации (EGP). IGP создан для отслеживания большого количества сведений о топологии сети. Приоритет в разработке IGP размещен в создание оптимальных маршрутов, и ответ быстро изменяется. Протокол EGP предназначен для защиты системы сетей от ошибок или преднамеренного искажения другими системами, BGP - пример такого протокола внешнего шлюза (EGP).. Приоритет в проектировании EGP должен отдаваться стабильности и административному контролю. Часто EGP достаточно создать разумный маршрут вместо оптимального.

Протокол IGRP имеет определенное сходство с предыдущими протоколами, например, протоколом маршрутной информации компании Xerox, протоколом Berkeley RIP и протоколом Dave Mills' Hello). Он отличается от данных протоколов как минимум тем, что изначально разрабатывался для более масштабных и сложных сетей. Для получения подробных сведений о сравнении с RIP, который широко использовался в старых поколениях протоколов, см. раздел "Сравнение с RIP".

Как и эти устаревшие протоколы, IGRP является протоколом вектора расстояния. В таком протоколе шлюзы обмениваются сведениями о маршрутизации только с соседними шлюзами. Эти сведения о маршрутизации содержат сводную информацию об остатке сети. Можно показать математически, что все шлюзы, взятые вместе, решают проблему оптимизации какой суммы к распределенному алгоритму. Каждый шлюз необходим для частичного решения проблем и для получения части общего объема данных.

Основной альтернативой протоколу IGRP являются расширенный протокол IGRP (EIGRP) и класс алгоритмов под общим названием SPF ("предпочтение кратчайшего пути"). OSPF использует это понятие. Для узнавания больше о OSPF обращаются к Руководству по проектированию OSPF. OSPF, который Это, основывается на способе затопления, где каждый шлюз усовершенствован о статусе каждого интерфейса на любом шлюзе. Каждый шлюз независимо друг от друга решает проблему оптимизации с его точки зрения при помощи данных из всей сети. Каждый подход имеет свои преимущества. В некоторых случаях SPF может более быстро реагировать на произошедшие изменения. Для предотвращения появления циклов маршрутизации IGRP должен игнорировать новые данные в течение нескольких минут после изменений определенного вида. Использование в протоколе SPF информации непосредственно от каждого шлюза позволяет избежать возникновения зацикливания маршрутизации. Таким образом, он может сразу действовать в соответствии с новыми сведениями. Однако SPF приходится работать со значительно большим количеством данных, чем IGRP, как во внутренних структурах данных, так и в сообщениях между шлюзами.

Проблема маршрутизации

IGRP предназначен для использования на шлюзах, подключенных к нескольким сетям. Мы предполагаем, что сети используют пакетную технологию. Фактически, шлюзы играют роль коммутаторов пакетов. Когда соединенная с одной сетью система собирается отправить пакет системе в другой сети, она адресует пакет на шлюз. Если назначение – одна из сетей, подключенных к шлюзу, шлюз будет перенаправлять пакеты до назначения. Если назначение будет более удалено, то шлюз передаст пакет к другому шлюзу, который ближе к назначению. Шлюзы используют таблицы маршрутизации, чтобы помочь им решать, что сделать с пакетами. Вот таблица маршрутизации простого примера. (Адресами, используемыми в примерах, являются IP-адреса, взятые от Университета Ратджерса. Обратите внимание, что основная проблема маршрутизации одинакова для всех протоколов, однако в описании подразумевается использование протокола IGRP для IP-маршрутизации.)

Рисунок 1

```
network      gateway     interface
  -------      -------     ---------
  128.6.4      none        ethernet 0
  128.6.5      none        ethernet 1
  128.6.21     128.6.4.1   ethernet 0
  128.121      128.6.5.4   ethernet 1
  10           128.6.5.4   ethernet 1
```

(Фактические таблицы Маршрутизации IGRP имеют дополнительные сведения для каждого шлюза, как мы будем видеть.) Этот шлюз связан с двумя Ethernets, вызванными 0 и 1. Им дали Номера сети IP (фактически номера подсетей) 128.6.4 и 128.6.5. Таким образом пакеты, обращенные для этих определенных сетей, могут быть переданы непосредственно назначению, просто при помощи соответствующего Интерфейса Ethernet. Существует два соседних шлюза, 128.6.4.1 и 128.6.5.4. Пакеты для сетей кроме 128.6.4 и 128.6.5 будут переданы одному или другим из тех шлюзов. Таблица маршрутизации указывает, какой шлюз должен использоваться для который сеть. Например, пакеты, предназначенные хосту в сети 10, необходимо переадресовать на шлюз 128.6.5.4. Каждый надеется, что этот шлюз ближе к сети 10, т.е. что оптимальный путь к сети 10 проходит этот шлюз. Основная цель IGRP – разрешить шлюзам создавать и поддерживать таблицы маршрутизации наподобие этой.

Основные сведения о IGRP

Как упомянуто выше, IGRP является протоколом, который позволяет шлюзам создавать свою таблицу маршрутизации путем обмена информацией с другими шлюзами. Построение таблицы шлюзом начинается с записей, соответствующих сетям, непосредственно подключенным к нему. Шлюз получает информацию о других сетях путем обмена обновлениями маршрутизации с соседними шлюзами. В самом простом случае шлюз найдет один путь, который представляет лучший способ добраться до каждой сети. Путь характеризуется следующими шлюзами, к которым должны быть отправлены пакеты, сетевой интерфейс, который должен быть использован, и метрические данные. Данные метрик являются рядом номеров, которые характеризуют, насколько хороший путь. Это позволяет шлюзу сравнивать пути, "услышанные" от различных шлюзов, и решать, какой из них использовать. Часто существуют случаи, где это целесообразно разделять трафик между двумя или больше путями. Протокол IGRP делает это каждый раз, когда два или более соединения одинаково успешны. Когда пути почти одинаково хороши, пользователь

может также настроить его для разделения трафика. В этом случае большее количество трафика будет посылаться по пути с лучшей метрикой. Замысел состоит в том, что трафик можно распределить между линиями 9600 бит/сек и 19200 бит/сек, и на линию 19200 ориентировочно придется в два раза больше трафика, чем на линию 9600 бит/сек.

Метрики, используемые IGRP, включают придерживающееся:

- Топологическое время задержки
- Пропускная способность самого узкого сегмента полосы пропускания данного пути
- Заполнение канала пути
- Надежность маршрута

Топологическое время задержки является количеством времени, которое оно заняло бы для получения до назначения вдоль того пути, приняв разгруженную сеть. Безусловно, при загрузке сети возникает дополнительная задержка. Однако загрузка составляется при помощи рисунка заполнения канала, не путем попытки измерить фактические задержки. Полоса пропускания пути - это пропускная способность самого медленного канала пути в битах в секунду. Заполнение канала указывает, сколько из той пропускной способности используется в настоящее время. Это измерено и изменится с загрузкой. Надежность означает текущий коэффициент ошибок. Это - часть пакетов, которые прибывают к неповрежденному месту назначения. Это измерено.

Несмотря на то, что они не используются в качестве части метрики, две части информации добавления передают с ним: счетчик переходов и значение MTU. Количество переходов – это просто число шлюзов, которые пакет должен пройти, чтобы достигнуть получателя. MTU – это максимальный размер пакета, который может быть передан по всему маршруту без фрагментации. (То есть это минимальное значение MTU для всех сетей, встречающихся на маршруте).)

На основе данных метрики для этого пути рассчитывается одна составная метрика. Составная метрика комбинирует эффект различных компонентов метрики в одиночный номер, представляющий "совершенство" того пути. Это - составная метрика, которая фактически используется для выбора оптимального пути.

Периодически каждый шлюз рассылает всю свою таблицу маршрутизации целиком (с некоторыми ограничениями, налагаемыми правилом разделения горизонта) всем соседним шлюзам. Когда шлюз получает это широковещательное сообщение от другого шлюза, это сравнивает таблицу со своей существующей таблицей. Любые новые назначения и пути добавлены к таблице маршрутизации шлюза. Пути в широковещании по сравнению с существующими путями. Если новый путь лучше, он может заменить существующий. Транслируемая информация также используется для обновления занятости канала и другой информации о существующих каналах. Эта общая процедура подобна используемой всеми протоколами маршрутизации по методу вектора расстояния. Это упомянуто в математической литературе как Алгоритм Беллмана-Форда. См. RFC 1058 для подробной разработки базовой процедуры, которая описывает RIP, более старый протокол маршрутизации по методу вектора расстояния.

В IGRP общий алгоритм Беллмана-Форда изменяется в трех критических аспектах. Во-первых, вместо простой метрики, вектор метрик используется для охарактеризования путей. Во-вторых, вместо выбора единственного пути с наименьшей метрикой, трафик распределяется по нескольким путям, метрики которых попадают в заданный диапазон. В-третьих, несколько функций представлены для обеспечения устойчивости в ситуациях, где изменяется топология.

Выбирается оптимальный маршрут с учетом составной метрики:

```
[(K1 / Be) + (K2 * Dc)] r
```

Если K1, K2 = константы, Be = полоса пропускания пути загрузки x (1 – канал занятости), Dc = топологическое ожидание, и r = надежность.

Путь с наименьшей составной метрикой является оптимальным. Где существуют разнообразные пути к тому же назначению, шлюз может направить пакеты по нескольким путям. Сделано в соответствии со сложной метрикой каждого пути данных. Например, если один маршрут имеет составную метрику 1, а другой маршрут - составную метрику 3, в три раза больше пакетов будут отосланы по маршруту данных, имеющих составную метрику 1.

Существует два преимущества для использования информации о векторе метрики. Во-первых, он обеспечивает возможность поддержки нескольких типов служб из одного и того же набора данных. Второе преимущество – это повышенная точность. Единая метрика обычно обрабатывается так же, как задержка. Каждая ссылка в пути добавлена к суммарной метрике. Если существует ссылка с низкой пропускной способностью, она обычно представляется большой задержкой. Однако ограничения пропускной способности действительно не накапливают способ, которым делают задержки. Путем обработки пропускной способности как отдельный компонент это может быть обработано правильно. Аналогичным образом нагрузка может рассматриваться как отдельный показатель загруженности канала.

IGRP предоставляет систему для взаимосвязанных компьютерных сетей, которые могут устойчиво обработать общую топологию графа включая петли. Система поддерживает данные метрик полного пути, т.е. это знает параметры пути ко всем другим сетям, с которыми связан любой шлюз. Трафик можно распределять по параллельным путям, и несколько параметров пути можно рассчитывать одновременно для всей сети.

Сравнение с RIP

Этот раздел сравнивает IGRP с RIP. Это сравнение полезно, поскольку RIP широко используется для целей, подобных IGRP. Однако такое сравнение не совсем корректно. RIP не был предназначен для совещания всех тех же целей как IGRP. RIP был предназначен для использования в небольших сетях с обоснованно унифицированной технологией. В таких приложениях этого в основном достаточно.

Большая часть основного отличия между IGRP и RIP является структурой их метрик. К сожалению это изменение нельзя просто встроить в RIP. Это требует нового подарка алгоритмов и структур данных в IGRP.

RIP использует простую метрику "hop count" (число переходов) для описания сети. В отличие от IGRP, где каждый путь описан задержкой, пропускной способностью, и т.д., в RIP, это описано номером от 1 до 15. Обычно этот номер используется для представления, сколько шлюзов путь проходит прежде, чем добраться до назначения. Это означает, что между медленной последовательной линией и Ethernet нет различий. В некоторых реализациях RIP для системного администратора возможно указать, что данный переход должен быть посчитан несколько раз. Низкоскоростные сети характеризуются большим числом переходов. Но так как максимум равняется 15, это не может быть сделано очень. Например, если Ethernet представлена 1 и линия 56 КБ 3, в пути может быть самое большее 5 линий 56 КБ, или максимум 15 превышен. Чтобы представлять полный диапазон доступных скоростей сети и обеспечить большую сеть, исследования, сделанные Cisco,

предполагают, что необходима 24-разрядная метрика. Если максимальная метрика слишком мала, то у системного администратора есть неприятный выбор: отказаться от различения быстрых и медленных маршрутов или от приведения сети в соответствие с ограничением. Фактически много национальных сетей являются теперь достаточно большими, что RIP не может обработать их, даже если каждый переход посчитан только однажды. Протокол RIP невозможно использовать в таких сетях.

Очевидным ответом было бы модифицировать RIP, чтобы разрешить более обширную метрику. К сожалению, это не сработает. Как все протоколы маршрутизации по методу вектора расстояния, RIP имеет проблему "подсчета к бесконечности". Это описано более подробно в RFC 1058. Когда изменения топологии, будут представлены фиктивные маршруты. Метрики, привязанные к этим фиктивным маршрутам медленно, увеличиваются, пока они не достигают 15, в этот момент маршруты удалены. 15 достаточно маленький максимум, что этот процесс будет сходиться справедливо быстро, предполагая, что используются синхронизируемые обновления. Если бы RIP модифицировался для разрешения 24-разрядной метрики, то петли сохранялись бы достаточно долго для метрики, которая будет посчитана до 2 ** 24. Это не терпимо. IGRP содержит функции, предназначенные для предотвращения использования фиктивных маршрутов. Они обсуждены ниже в разделе 5.2. Это не практично для обработки сложных сетей, не представляя такие функции или изменяясь на протокол, такие как SPF.

IGRP не только расширяет диапазон допустимых метрик. Метрика реструктурируется для описания задержки, пропускной способности, надежности и загрузки. Возможно отразить подобные моменты в одной метрике, такой как RIP. Однако подход, принятый IGRP, потенциально более точен. Например, с одиночной метрикой, несколько последовательных быстрых каналов, будет казаться, будут эквивалентны одиночному медленному. Это может иметь место для интерактивного трафика, где задержка является основным предприятием. Однако в случае групповой передачи данных основная проблема – полоса пропускания, и одновременное добавление метрик здесь нецелесообразно. IGRP обрабатывает задержки и полосу пропускания отдельно, накапливая задержки, но используя при этом минимум полосы пропускания. Нелегко понять, как соединить эффекты надежности и нагрузки в однокомпонентной метрике.

По моему мнению, одно из больших преимуществ IGRP является простотой конфигурации. Это может непосредственно представлять количества, которые имеют физический смысл. Это означает, что может быть установлено автоматически, на основе типа интерфейса, скорости линии, и т.д. С однокомпонентной метрикой метрику придется, более вероятно, "приготовить" для слияния эффектов нескольких разных вещей.

Другие нововведения больше касаются алгоритмов и структур данных, чем протокола маршрутизации. Например, IGRP задает алгоритмы и структуры данных, поддерживающие разделение трафика по нескольким маршрутам. Конечно, возможно разработать реализацию RIP, который делает это. Однако после повторного внедрения маршрутизации причины продолжать использовать RIP нет.

До сих пор я описал "IGRP общего назначения", технология, которая могла поддержать маршрутизацию для любого сетевого протокола. Однако в этом разделе стоит упомянуть о конкретной реализации TCP/IP. То есть реализации, которую предполагают сравнивать с RIP.

Сообщения обновления RIP просто содержат снимки таблицы маршрутизации. То есть у них есть ряд мест назначений с метриками и еще некоторые данные. Реализация IP IGRP имеет дополнительную структуру. Во-первых, обновленное сообщение определено "номером

автономной системы. Эта терминология берет свое начало из проекта Arpanet, где имела вполне конкретный смысл. Однако для большинства сетей это означает возможность выполнения нескольких различных систем маршрутизации в одной сети. Это полезно для мест, где сходятся сети от нескольких организаций. Каждая организация может поддерживать свою собственную маршрутизацию. Из-за того, что каждое обновление помечается, шлюзы можно настроить так, чтобы они обращали внимание только на нужные. Некоторые шлюзы настроены для получения обновлений от нескольких автономных систем. Они передают информацию между системами управляемым способом. Обратите внимание, что это не полное решение проблемы безопасности маршрутизации. Любой шлюз может быть настроен для слушания обновлений от любой автономной системы. Однако это весьма полезное средство для реализации политик маршрутизации с приемлемым уровнем доверия между сетевыми администраторами.

Вторая структурная функция, касающаяся сообщений об обновлении IGRP, влияет на способ обработки маршрутов по умолчанию с помощью IGRP. Большинство протоколов маршрутизации используют идею маршрута по умолчанию. Это часто не практично для обновлений маршрута для распечатки каждой сети в мире. Обычно набор шлюзов требует детальные сведения о маршрутизации для сетей внутри их организации. Весь трафик для назначений за пределами их организации может быть передан одному из нескольких граничных шлюзов. Эти граничные шлюзы могут содержать более полную информацию. Маршрут к лучшему граничному шлюзу является "маршрутом по умолчанию". Это - по умолчанию в том смысле, что это используется для получения до любого назначения, которое не перечислено в частности в обновлениях внутренней маршрутизации. RIP и некоторые другие протоколы маршрутизации, распространяют информацию о маршруте по умолчанию, как будто это была реальная сеть. IGRP использует другой подход. Вместо единой ложной записи для маршрута по умолчанию IGRP разрешает маркировку реальных сетей как кандидатов на статус сети по умолчанию. Это реализуется за счет размещения информации о данных сетях в специальном внешнем разделе сообщения об обновлении. Однако это могло бы также считаться включающий немного привязанный к тем сетям. IGRP периодически сканирует все потенциальные маршруты по умолчанию и выбирает один с наименьшей метрикой в качестве действующего маршрута по умолчанию.

Этот подход к выбору стандартных маршрутов может оказаться несколько более гибким, чем подход, применяемый в большинстве версий RIP. Чаще всего RIP-шлюзы можно настроить на создание маршрута по умолчанию с определенной меткой. Операция будет выполнена на граничных шлюзах.

Подробное описание

Этот раздел предоставляет подробное описание IGRP.

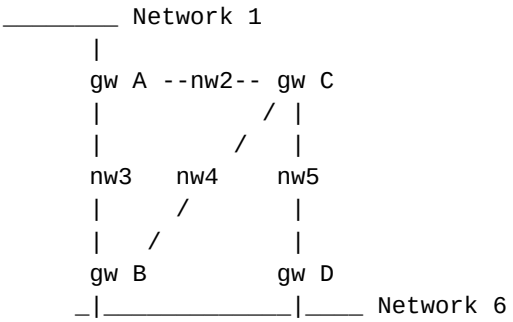
Общее описание

Когда шлюз включается впервые, происходит инициализация его таблицы маршрутизации. Это может сделать оператор при помощи консольного терминала либо считав данные из файлов конфигурации. Дается описание каждой сети, подключенной к шлюзу, включая топологическую задержку по каналу (например, сколько времени уходит у одного бита на прохождение канала) и полосу пропускания канала.

Рис. 2

К примеру, на приведенной выше диаграмме шлюзу S было бы указано, что он подключен к сетям 2 и 3 через соответствующие интерфейсы. Таким образом, первоначально, шлюз 2 только знает, что может достигнуть любого конечного компьютера в сетях 2 и 3. Все шлюзы запрограммированы для периодической передачи к их соседним шлюзам информации, что они инициализировались с, а также информация, собранная из других шлюзов. Таким образом шлюз S получил бы обновления от шлюзов R и T и узнал бы, что это может достигнуть компьютеров в сети 1 через шлюз R и компьютеры в сети 4 через шлюз T. Начиная со шлюза S передает свою всю таблицу маршрутизации, в следующем шлюзе цикла T узнает, что это может добраться до сети 1 через шлюз S. Очевидно, что информация о любой сети системы в конечном счете достигнет любого шлюза системы, при условии, что сеть полносвязная.

Рис. 3



Каждый шлюз рассчитывает композитную метрику для определения привлекательности маршрутов передачи данных к конечным компьютерам. Например, в схеме выше, для назначения в Сети 6, шлюз (gw A) вычислил бы метрические функции для двух путей через шлюзы B и C. Обратите внимание на то, что пути определены просто следующим переходом. Существует фактически три возможных маршрута от до Сети 6:

- Прямо к B
- К C и затем к B
- К C, затем к D

Однако шлюз потребность не выбирает между двумя маршрутами, включающими C. Таблица маршрутизации в A имеет одиночную запись, представляющую путь к C. Его метрика представляет лучший способ добраться от C до конечного назначения. Если A передает пакет C, решение об использовании B или D принимается C самостоятельно.

Уравнение 1

Ниже приведен расчет функции составной метрики для каждого пути данных:

```
[(K1 / Be) + (K2 * Dc)] r
```

Где r = частичная надежность (% передач, которые успешно получены в следующем переходе), Система цифрового управления = суммарная задержка, Быть = эффективная пропускная способность: пропускная способность незагруженного канала x (1 - заполнение канала), и K1 и K2 = константы.

Уравнение 2

В принципе, общая составное значение задержки (Dc) можно определить, как показано ниже:

```
Dc = Ds + Dcir + Dt
```

Где Ds = задержка переключения, Dcir = задержка цепи (задержка распространения 1 бита) и DT = задержка передачи (задержка для 1500-битного сообщения без учета времени загрузки).

Однако на практике цифра стандартной задержки используется для каждой технологии типа сети. Например, для Ethernet и линий последовательной передачи при любой скорости потока используется схема стандартной задержки.

Здесь приводится пример того, как может выглядеть таблица маршрутизации шлюза A' в случае топологии сети на схеме 6 вверху. (Обратите внимание, что для простоты отдельные компоненты вектора метрики не отображаются.).)

Пример таблицы маршрутизации:

Сеть	Interface	Следующий шлюз	Метрика
1	NW 1	Нет	Непосредственно связанный
2	NW 2	Нет	Непосредственно связанный
3	NW 3	Нет	Непосредственно связанный
4	NW 2	C	1270
	NW 3	B	1180
5	NW 2	C	1270
	NW 3	B	2130
6	NW 2	C	2040
	NW 3	B	1180

Основной процесс построения таблицы маршрутизации с помощью обмена сведениями с соседями описан алгоритмом Беллмана-Форда. Алгоритм использовался в более ранних протоколах, таких как RIP (RFC 1058). Для работы в более сложных сетях IGRP дополняет базовый алгоритм Беллмана-Форда тремя возможностями:

1. Вектор метрик используются для того, чтобы охарактеризовать путь, вместо простой метрики. Можно вычислить составную метрику при помощи данного вектора в соответствии с приведенным выше уравнением 1. Использование вектора позволяет шлюзу принимать различные типы сервиса, при помощи нескольких других коэффициентов в Уравнении 1. Это также дает возможность более точного представления характеристик сети, чем одиночная метрика.
2. Вместо того, чтобы выбрать один маршрут наименьшей длины, трафик будет поделен между несколькими маршрутами, значения длины которых принадлежат к указанному диапазону значений. Это позволяет нескольким маршрутам использоваться параллельно, предоставляя большую эффективную пропускную способность, чем какой-либо один маршрут. Дисперсию V задает администратор сети. Удерживаются все путирс минимальной композитной метрикой M. Кроме того, сохраняются все пути с метрикой ниже V x M. Трафик распределен среди разнообразных путей в обратной пропорции к составным метрикам.
3. Существует несколько проблем в связи с этим понятием дисперсии. Трудно придумать

стратегии, которые используют значения различия, больше, чем 1, и также не приводят к пакетному цикличному выполнению. Возможность дисперсии в Cisco выпуска 8.2 не реализована. (Я не уверен, в каком выпуске была удалена функция.) Эффект этого состоит в том, чтобы установить различие постоянно в 1.

4. Представлены некоторые возможности обеспечения стабильности в ситуациях изменения топологии. Эти функции предназначены для предотвращения циклов маршрутизации и "рассчитывающий к бесконечности", которые охарактеризовали предыдущие попытки использовать алгоритмы типа Форда для этого типа приложения. Основными функциями стабильности являются "удержания", "запускаемые обновления", "разделение горизонтов" и "устранение ошибок". Они будут обсуждены более подробно ниже.

Разбиение трафика (точка 2) вызывает незначительную угрозу. Применение отклонения V позволяет шлюзам использовать параллельные пути с разными скоростями передачи. Например, это может быть линия со скоростью передачи 9600 бит/с, работающая параллельно с линией со скоростью передачи на 19200 бит/с для обеспечения избыточности. Если различие V будет равняться 1, то только оптимальный путь будет использоваться. Если линия на 19200 битов в секунду будет иметь приемлемую надежность, таким образом, линия на 9600 битов в секунду не будет использоваться. (Однако, если несколько путей будут тем же, то загрузка будет разделена среди них.) Путем повышения различия, мы можем позволить трафику быть разделенным между лучшим маршрутом и другими маршрутами, которые являются почти как хорошие. При достаточно большой дисперсии трафик будет разделен между двумя линиями. Опасность заключается в том, что при большом разбросе вариантов становятся доступны пути не только более медленные, но и "в неправильном направлении". Таким образом, должно быть дополнительное правило, чтобы препятствовать отправке трафика "вверх по течению": Трафик не отправляется по каналам, удаленная комбинированная метрика которых (вычисляется в следующем узле) больше комбинированной метрики, вычисленной на шлюзе. Вообще говоря, системным администраторам не следует устанавливать изменения более чем на 1, кроме особых ситуаций, где требуется использование параллельных путей. В данном случае величина погрешности тщательно подобрана для получения "правильных" результатов.

IGRP предназначен для обработки нескольких типов служб и нескольких протоколов. Тип сервиса является спецификацией в пакете данных, который модифицирует дорожки, должны быть оценены. Например, протокол TCP/IP позволяет пакетам определять относительную важность высокой пропускной способности, низкого интервала задержки и высокой надежности. В общем случае интерактивные приложения устанавливают низкую задержку, а приложения группового переноса – высокую полосу пропускания. Эти требования определяют относительные значения K1 и K2, подходящие для использования в Eq. 1. Каждая комбинация спецификаций в пакете, которая должна поддерживаться, называется "типом обслуживания". Для каждого типа обслуживания должен быть выбран набор параметров K1 и K2. Таблица маршрутизации ведется для каждого типа обслуживания. Это сделано потому, что пути выбраны и упорядочены в соответствии со сложной метрикой, определённой Eq. 1. Это различается для каждого типа обслуживания. Данные всех таблиц маршрутизации объединяются с целью формирования сообщений обновления маршрутизации, которыми обмениваются шлюзы, как описано на рис. 7.

Средства обеспечения стабильности

Данный раздел содержит описание удержаний, обновлений, разделения горизонта и искажений. Данные функции разработаны для предотвращения перехвата шлюзами

ошибочных маршрутов. Когда маршрут становится неприменимым, из-за сбоя шлюза или сети, как описано в RFC 1058, это может произойти. В принципе соседние шлюзы обнаруживают сбои. Затем они посылают обновления маршрутизации, отображающие старый маршрут как неиспользуемый. Однако для обновлений возможно не достигнуть некоторых частей сети вообще или быть задержанным в достижении определенных шлюзов. Шлюз, который полагает, что старый маршрутизатор находится в рабочем состоянии, может продолжить распространение этих данных, повторно вводя в систему нерабочий маршрутизатор. В конечном счете эта информация распространится через сеть и возвратится к шлюзу что повторно внедренный это. Результатом является кольцевой маршрут.

Фактически среди контрмер существует некоторое резервирование. В принципе, достаточно использовать удержание и автоматические обновления, чтобы предотвратить использование недопустимых маршрутов. Однако на практике их может быть недостаточно для исключения сбоев связи. Расщепленный горизонт и искажение маршрута предназначены для предотвращения циклов маршрутизации в любом случае.

Обычно, новые таблицы маршрутизации передаются соседним шлюзам регулярно (каждые 90 секунд по умолчанию, невзирая на то, что это может быть отрегулировано системным администратором). Вызванное обновление – это новая таблица маршрутизации, которая посылается незамедлительно в ответ на некоторые изменения. Самое важное изменение является удалением маршрута. Это может произойти, потому что таймаут истек (вероятно, соседний шлюз или линия выключились), или потому что обновленное сообщение от следующего шлюза в пути показывает, что путь больше не применим. Обнаружив, что какой-либо маршрут больше не может использоваться, шлюз G немедленно инициирует обновление. Это обновление покажет, что маршрут непригоден для использования. Рассмотрим, что произойдет, когда это обновление достигнет соседних шлюзов. Если маршрут соседа указывает обратно на G, то сосед должен удалить этот маршрут. Это заставляет соседний узел инициировать обновление и т.д. Таким образом сбой инициирует волну обновленных сообщений. Эта волна распространится всюду по той части сети, в которой маршруты прошли отказавший шлюз или сеть.

Обновления, запускаемые по условию, могут быть достаточными в случае, если можно гарантировать, что волна обновлений достигнет всех соответствующих шлюзов немедленно. Однако существует две проблемы. Во-первых, пакеты, содержащие обновленное сообщение, могут быть отброшены или повреждены некоторой ссылкой в сети. Во-вторых, запускаемые обновления не устанавливаются мгновенно. Может случиться так, что шлюз, еще не получивший инициированное обновление, сам запустит регулярное обновление в самый неподходящий момент, в результате чего в соседний узел, уже получивший обновление, будет добавлен неверный маршрут. Чтобы обойти эту проблему были разработаны удержания. Правило удержания предписывает, что при удалении маршрута в течение некоторого времени для данного назначения не будут приниматься новые маршруты. Это позволяет запущенным обновлениям достичь всех остальных шлюзов, и пользователь может быть уверен, что все новые маршруты не являются старыми маршрутами, повторно вставляемыми каким-либо маршрутизатором. Период удержания должен быть достаточно длинным для учета волны синхронизируемых обновлений для движения всюду по сети. Кроме того, для обработки отброшенных пакетов в нем следует учитывать несколько циклов регулярного широковещания. Рассмотрите то, что происходит, если одно из синхронизируемых обновлений отброшено или повреждено. Шлюз, запустивший то обновление, запустит еще одно обновление при выполнении планового обновления. Таким образом перезапускается волна обновлений на соседних узлах, которые пропустили первую волну.

Комбинация синхронизируемых обновлений и holddowns должна быть достаточной, чтобы избавиться от просроченных маршрутов и препятствовать тому, чтобы они были повторно вставлены. Однако некоторые дополнительные меры предосторожности стоит сделать так или иначе. Они обеспечивают очень сети с потерями данных и сети, которые стали разделенными. Дополнительные меры предосторожности, вызванные IGRP, предполагают разделение диапазонов и искажение маршрута. Разделение горизонта возникает из наблюдения, что нет смысла отправлять маршрут назад по направлению его прибытия. Рассмотрим следующую ситуацию:

```
network 1        network 2
  -------------X-----------------X
           gateway A          gateway B
```

Шлюз A скажет B, что это имеет маршрут к сети 1. Когда B передает обновления A, никогда нет никакой причины для него для упоминания сети 1. Так как A ближе к 1, нет никакой причины для него для рассмотрения движения через B. Правило разделения горизонтов говорит, что отдельное обновленное сообщение должно генерироваться для каждого соседнего узла (фактически каждая соседняя сеть). Обновление определенного соседа должно пропускать маршруты, которые указывают на этого соседа. Это правило предотвращает петли между соседними шлюзами. Предположим, например, что неисправен интерфейс A для сети 1. Без правила разделения горизонтов B сказал бы, что это может добраться до 1. Так как это больше не имеет реальный маршрут, A мог бы забрать тот маршрут. В этом случае A и B оба имел бы маршруты к 1. Но A указал бы к B, и B укажет к A. Конечно, синхронизируемые обновления и holddowns должны предотвратить это. Но поскольку необходимости в возврате информации в исходное место нет, рекомендуется все равно выполнить разделение горизонта. Помимо участия в предотвращении зацикливаний, разделение горизонта сдерживает увеличение размера сообщений об обновлении.

Разделение горизонтов должно предотвратить появление циклов между соседними шлюзами. Искажение маршрута предназначено для ломки больших петель. Если в процессе обновления отображается значительно возросшая метрика существующего маршрута, это свидетельствует о возникновении зацикливания. Маршрут должен быть удален и переведен в режим удержания. В настоящее время правило состоит в том, что маршрут удален, если составная метрика увеличивает больше, чем фактор 1.1. Не безопасно для просто никакого увеличения составной метрики инициировать удаление маршрута, так как небольшие метрические изменения могут произойти из-за изменений в заполнении канала или надежности. Таким образом, коэффициент 1,1 получен опытным путем. Точное значение неважно. Мы ожидаем, что это правило только будет необходимо для ломки очень больших петель, так как маленькие будут предотвращены синхронизируемыми обновлениями и holddowns.

Отключение удержаний

Начиная с версии 8.2, код Cisco предоставляет возможность отключения удержаний. Неудобством удержаний является задержка принятия нового маршрута при отказе старого. Если заданы параметры по умолчанию, маршрутизатору может понадобиться несколько минут на утверждение нового маршрута. Однако, для причин, описанных выше, не является безопасным просто удалить крепления. Результатом было бы зацикливание, как описано в RFC 1058. Мы догадываемся, но не можем доказать, что с более сильной версией искажения маршрута, holddowns больше не необходимы для остановки зацикливания. Таким образом, отключение закреплений может привести к более сильному искажению маршрута. Следует отметить, что еще действуют разделение горизонтов и запускаемые обновления.

Более сильная форма искажения маршрута базируется на количестве переходов. Если счетчик переходов для пути увеличивается, маршрут удаляется. Это, очевидно, удалит маршруты, которые все еще допустимы. Если в сети что-то где-то меняется так, что теперь маршрут проходит через еще один шлюз, число прыжков увеличивается. В данном случае маршрут остается действительным. Однако нет совершенно надежного способа отличить этот случай от петель маршрутизации (число отсчётов идет к бесконечности). Так, наиболее безопасным подходом является отключение маршрута всякий раз, когда возрастает число переходов. Если маршрут еще действителен, он будет установлен заново при последующем обновлении, что приведет к выполнению вызванного обновления и повторной установке маршрутов во всей системе.

В целом маршрутизация по методу вектора расстояния algorithms1 принимает новые маршруты легко. Проблему представляет собой полная очистка системы от устаревших маршрутов. Таким образом, чрезмерно агрессивное правило удаления подозрительных маршрутов должно обеспечивать безопасность.

Подробные сведения о процессе обновления

Набор процессов, описанных на рис. 4 – 8, предназначен для обработки отдельного сетевого протокола, например протокола TCP/IP, DECnet или ISO/OSI. Данные приводятся только для протокола TCP/IP. Единственный шлюз может обрабатывать данные, поддерживающие несколько протоколов. Поскольку каждый протокол имеет другие структуры адресации и форматы пакета, код компьютера, используемый для реализации рисунков 4 - 8, обычно будет другим для каждого протокола. Процесс, описанный на рисунке 4, будет варьироваться больше всего, как описано в подробных примечаниях для рисунка 4. Процессы, описанные на рисунке 5 - 8, будут иметь ту же общую структуру. Основное различие между протоколами заключается в формате пакета обновления маршрутизации, который должен быть совместимым с определенным протоколом.

Обратите внимание, что определения адреса назначения могут отличаться для разных протоколов. Метод, описанный здесь, можно использовать для конфигурирования маршрутизации к индивидуальным хостам, сетям или для более сложных иерархических схем адресации. Используемый тип маршрутизации зависит от структуры адресации в протоколе. Текущая реализация TCP/IP поддерживает только маршрутизацию в сети IP. Таким образом "назначение" действительно имеет в виду Номер сети IP или номер подсети. Данные подсети сохраняются только для подключенных сетей.

Рисунки с 4 по 7 отображают псевдокод для различных секций процесса маршрутизации, который используется шлюзами. При запуске программы вводятся применимые протоколы и параметры, описывающие каждый интерфейс.

Шлюз только обработает определенные протоколы, которые перечислены. Любая связь с системой, использующей протокол, не входящий в список, будет игнорироваться. Входные данные являются придерживающимся:

- Сети, к которым подключен шлюз.
- Незагруженная пропускная способность каждой сети.
- Топологическая задержка каждой сети.
- Надежность каждой сети.
- Заполнение канала каждой сети.
- MTU каждой сети.

Метрическая функция для каждого пути данных тогда вычислена согласно Уравнению 1.

Обратите внимание на то, что первые три элемента являются довольно постоянными. Это функции основной сетевой технологии, которые не зависят от нагрузки. Их настройку можно выполнить с помощью файла конфигурации или использовав прямой ввод оператора. Обратите внимание, что в протоколе IGRP не используется измеряемая задержка. Как теория, так и опыт говорят о том, что протоколам, использующим измерения задержки, очень трудно поддерживать стабильную маршрутизацию. Существуют два измеряемых параметра: надежность и заполнение канала. Надежность измеряется количеством ошибок, регистрируемых на аппаратном или микропрограммном уровне сетевого интерфейса.

Помимо этих входных данных, алгоритм маршрутизации требует указать значение нескольких параметров. Это включает значения таймера, различие, и включены ли holddowns. Обычно это устанавливается посредством файла конфигурации или ввода оператора. (Как и в Cisco версии 8.2, дисперсия имеет постоянное значение "1".).)

Как только исходные данные введены, операции в шлюзе инициированы событиями — или прибытие пакета данных в одном из сетевых интерфейсов или истечение таймера. Процессы, описанные на рисунках 4-7, запускаются следующим образом:

- Когда пакет поступает, он обработан согласно рисунку 4. В результате пакет отправляется с другого интерфейса, отбрасывается или принимается на дальнейшую обработку.
- Когда пакет принят шлюзом для дальнейшей обработки, это проанализировано определяемой протоколом формой, не описанной в этой спецификации. Если это пакет обновления маршрутизации, он обрабатывается в соответствии с рис. 5.
- Рисунок 6 показывает события, инициированные таймером. Таймер настроен на создание прерывания один раз в секунду. Процесс, показанный на рисунке 6, выполняется, если произошло прерывание.
- Рисунок 7 показывает подпрограмму обновления маршрута. Вызовы к этой подпрограмме показаны на рис. 5 и 6.
- Кроме того, на рис. 8 показаны подробности расчета метрик, упомянутых на рис. 5 и 7.

Существует четыре раза константы, которые управляют распространением маршрутов и истечением. Эти временные константы могут быть установлены системным администратором. Однако существуют значения по умолчанию. Эти константы времени следующие:

- Время трансляции — Обновления переданы всеми шлюзами на всех связанных интерфейсах это часто. По умолчанию – один раз каждые 90 секунд.
- Недопустимое время — Если никакое обновление не было получено для данного пути в этом периоде времени, это, как полагают, испытало таймаут. Это должно несколько раз быть время трансляции, для получения возможности возможности, что пакеты, содержащие обновление, могли быть отброшены сетью. По умолчанию широковещательный канал опрашивается 3 раза.
- Время удержания — Когда назначение стало недостижимым (или метрика увеличился достаточно для порождения отравления), назначение входит в "удержание". Во время этого состояния никакой новый путь не будет принят для того же назначения для этого периода времени. Значение времени удержания указывает на длительность этого состояния. Периодов широковещания должно быть несколько. Значение по умолчанию в три раза больше времени трансляции плюс десять секунд. (Как описано в разделе "Отключение удержаний", удержания можно отключить.).)
- Время очистки удержания — Если никакое обновление не было получено для заданного

получателя в этом периоде времени, записи для него, удалено из таблицы маршрутизации. Обратите внимание на различие между недопустимым временем и временем очистки удержания: По истечении недопустимого периода путь считается устаревшим и удаляется. Если не осталось других путей к месту назначения, оно становится недоступным. Однако запись в базе данных об этом направлении остается. Остается принудительно установить удержание. После выполнения сброса запись базы данных удаляется из таблицы. Это должно быть дольше, чем недопустимое время плюс время удержания. Значение по умолчанию – в 7 раз больше времени широковещания.

Эти рисунки предполагают следующие главные структуры данных. Для каждого протокола, поддерживаемого шлюзом, хранится отдельный набор таких структур данных. В рамках любого протокола для каждого поддерживаемого типа обслуживания существует отдельный набор структур данных.

Для всех известных системе назначений указывается список ведущих к ним путей (возможно, пустой), время истечения удержания и время последнего обновления. Время последнего обновления указывает время, когда в последний раз путь для этого пункта назначения был включен в обновление из другого шлюза. Обратите внимание на то, что для каждого пути имеется время обновления. Когда удален последний путь к месту назначения, место назначения помещается в захват, если захваты не отключены (дополнительные сведения см. в разделе "Отключение захватов"). Время истечения удержания указывает время окончания срока действия удержания. Факт, что это является ненулевым, указывает, что назначение находится в удержании. Для сохранения времени конфигурации рекомендуется хранить "лучшую метрику" для каждого получателя. Это минимальное количество составных метрик для всех маршрутов к адресу назначения.

Для каждого пути к месту назначения существует адрес следующего перехода на пути, используемый интерфейс, вектор системы показателей, характеризующих путь, включая топологическую задержку, пропускную способность, надежность и занятость канала. Другая информация также привязана к каждому пути, включая счетчик переходов, MTU, источник информации, удаленную составную метрику и составную метрику, вычисленную от этих номеров согласно уравнению 1. Существует также время последнего обновления. Источник информации показывает, откуда пришли наиболее свежие обновления для этого пути. На практике это совпадает с адресом следующего перехода. Последнее время обновления – это время, когда для данного пути было получено последнее обновление. Используется для окончания истекших по времени путей.

Обратите внимание, что сообщение обновления IGRP состоит из трех частей: внутренняя, система (в смысле "эта автономная система", но не внутренняя) и внешняя. Внутренняя часть предназначена для маршрутов к подсетям. Не все сведения о подсети включены. Только подсети одной сети включены. Это сеть, связанная с адресом, на который отправляется обновление. Обычно обновления передаются на каждый интерфейс, поэтому это просто сеть, в которую посылается широковещательный сигнал. (Другие случаи возникают для ответов на IGRP - запрос и IGRP "точка-точка".) Крупные сети (например, неподсети) помещены в системную часть обновленного сообщения, пока они в частности не отмечены как внешний вид.

Сеть будет отмечена как внешний вид, если это было изучено из другого шлюза, и информация поступила во внешнюю часть обновленного сообщения. Реализация Cisco также позволяет системному администратору объявить определенные сети внешними. Внешние маршруты называются также "кандидатами по умолчанию". Они - маршруты, которые переходят или через шлюзы, которые, как полагают, являются соответствующими как настройки по умолчанию, используются, когда нет никакого явного маршрута

назначению. Например, в Rutgers мы настраиваем шлюз, который соединяет Rutgers с нашей региональной сетью таким образом, что он помечает маршрут к магистрали NSFnet, как внешний. При развертывании Cisco выбирается маршрутизатор по умолчанию при помощи выбора внешнего маршрута с наименьшей метрикой.

Следующие разделы предназначены для разъяснения некоторых фрагментов рисунков 4 - 8.

Маршрутизация пакетов

Рис. 4 поясняет общий процесс обработки входящих пакетов. Это нужно только для прояснения терминологии. Очевидно, что это не полное описание функций IP шлюза.

Этот процесс использует список поддерживаемых протоколов и сведения про интерфейсы, введенные во время инициализации шлюза. Подробности обработки пакетов зависят от протокола, используемого пакетом. Это определяется на шаге A. Шаг A является единственной частью рисунка 4, общей для всех протоколов. Как только тип протокола известен, реализация рисунка 4, соответствующего типу протокола, используется. Подробная информация о содержимом пакета приведена в спецификации протокола. Спецификации протокола включают в себя процедуру определения пункта назначения пакета, процедуру сравнения пункта назначения с собственным адресом шлюза для определения того, является ли сам шлюз пунктом назначения, процедуру определения, является ли пакет пакетом широковещания и процедуру определения, является ли пункт назначения частью указанной сети. Эти процедуры используются в шагах B и C рисунка 4. Тест в шаге D требует поиска назначений, перечисленных в таблице маршрутизации. Тест удовлетворен, существует ли запись в таблице маршрутизации для назначения, и что назначение привязало к нему по крайней мере один доступный путь. Обратите внимание на то, что назначение и данные пути, используемые в этом и следующем шаге, поддержаны отдельно для каждого поддерживаемого типа сервиса. Таким образом, этот шаг начинается с определения типа обслуживания, указанного в пакете, и выбора соответствующего набора структур данных для использования во время этого и следующего шага.

Если его удаленная составная метрика является меньше, чем его составная метрика, путь применим в целях шагов D и E. Маршрут, чья удаленная составная метрика больше его составной метрики, является маршрутом со следующим переходом, находящимся на значительном удалении от места назначения (на основании измерений метрик). Этот путь называется "восходящим маршрутом". Как правило, использование метрик исключает выбор путей от абонента к оператору. Легко видеть, что путь в направлении оператора никогда не будет оптимальным. Однако, если большое различие позволено, пути кроме лучшего могут использоваться. Некоторые из тех могли быть восходящими.

В шаге E вычисляется используемый путь. Пути, удаленная комбинированная метрика которых меньше комбинированной метрики, не рассматриваются. Если несколько путей приемлемы, такие пути используются во взвешенной форме циклической дизъюнкции. Частота использования пути обратно пропорциональна его суммарной метрике.

Получение обновления маршрута

Рисунок 5 описывает обработку обновления маршрута, полученного от соседнего шлюза. Такие обновления состоят из списка записей, каждая из которых дает информацию для одного места назначения. Более одной записи для одного пункта назначения в одной операции обновление маршрутизации может возникнуть для того, чтобы обеспечить

различные типы обслуживания. Каждая из этих записей обработана индивидуально, как описано на рисунке 5. Если запись находится во внешнем разделе обновления, внешний флаг будет настроен на назначение, если он добавляется как результат этого процесса.

Весь процесс, описанный на рисунке 5, следует повторить для каждого типа службы, поддерживаемого шлюзом, используя сведения set of destination / path, сопоставленные данному типу службы. Это показывают в наиудаленном цикле на рисунке 5. Все обновление маршрута должно быть обработано однажды для каждого типа сервиса. (Обратите внимание, что текущее выполнение IGRP не поддерживает множественность типов сервиса, поэтому самая ближняя внешняя петля в реальности не выполняется).)

В действии A выполняется базовая проверка приемлемости параметров пути. Следует включить тесты на обоснованность для направления. Невозможные ("марсианские") номера сетей должны быть отклонены. (См. RFC 1009 и RFC 1122 для получения дополнительной информации.) Обновления также отклонены, если назначение, к которому они обращаются, находится в удержании, т.е. время истечения удержания является ненулевым и позже, чем текущее время.

В действии B осуществляется поиск в таблице маршрутизации с целью определить, описывает ли данная запись уже известный путь. Путь в таблице маршрутизации определен назначением, с которым это привязано, следующий переход, перечисленный как часть пути, выходной интерфейс, который будет использоваться для пути и источника информации (адрес, из которого обновление прибыло — на практике обычно то же как следующий переход). Запись из пакета обновления описывает путь, адрес назначения которого указан в другой записи, выходной интерфейс которой совпадает с интерфейсом получения обновления, а адрес следующего узла и источника данных совпадает с адресом шлюза-отправителя обновления (источник S).

На шаге H и шаге T составляется расписание процесса обновления, описанного на рис. 7. Этот процесс будет запущен после полного завершения процесса, изображенного на рис. 5. Т.е. процесс обновления, описанный на рисунке 7, только произойдет однажды, даже если это будет инициироваться несколько раз во время обработки, описанной на рисунке 5. Кроме того, необходимо принять меры предосторожности, чтобы исключить слишком частый выпуск обновлений, если сеть быстро изменяется.

Если назначение, описанное текущей записью в обновленном пакете уже, существует в таблице маршрутизации, шаг K сделан. На шаге K новая композитная метрика, рассчитанная на основании данных в пакете обновления, сравнивается с лучшей композитной метрикой назначения. Заметьте, что лучшая составная метрика не пересчитывалась в этот раз, поэтому, если рассматриваемый путь уже находится в таблице маршрутизации, этот тест поможет сравнить старые и новые метрики одного пути.

Шаг L выполнен для путей, которые хуже, чем существующая лучшая составная метрика. Это включает и новые пути, которые хуже, чем существующие и существующие пути, составная метрика которых увеличилась. На этапе L выполняется проверка допустимости нового пути. Обратите внимание на то, что этот тест внедряет и тест для того, достаточно хорош ли новый путь для хранения, и искажение маршрута. Допустимое значение задержки не должно быть тем особым значением, которое отображает недоступный узел назначения (в текущей реализации протокола IP – все единицы в 24-битном поле), а композитная метрика (расчет как на рисунке 8) также должна быть допустимой. Чтобы определить, приемлема ли составная метрика, сравните ее с составными метриками всех других путей к назначению. Позвольте M быть минимумом их. Новый маршрут приемлем, если он < V X M ГДЕ V ДИСПЕРСИЯ, УСТАНОВЛЕННАЯ ПРИ ИНИЦИАЛИЗАЦИИ ШЛЮЗА. ЕСЛИ V = 1

(ЧТО ВСЕГДА ВЕРНО НАЧИНАЯ С РЕЛИЗА CISCO RELEASE 8.2), ТО ЛЮБАЯ МЕТРИКА МЕНЬШЕ ИМЕЮЩЕЙСЯ НЕПРИЕМЛИМА. СУЩЕСТВУЕТ ОДНО ИСКЛЮЧЕНИЕ: Если путь уже существует и является единственным путем к назначению, он будет сохранен, если метрика не была увеличена более чем на 10% (или удержания отключаются, если значение счетчика переходов не увеличилось).

Шаг V выполняется, когда новые сведения для пути указывают, что составная метрика будет уменьшена. Составные метрики всех путей к назначению Д сравнены. Данное сравнение использует новую композитную метрику для P, а не ту, что отображается в таблице маршрутизации. Минимальная составная метрика M вычислена. Затем все пути к D проверяются снова. Если составная метрика для любого пути > M x V, этот путь удален. "V" - дисперсия, указанная при инициализации шлюза. (Как и в Cisco версии 8.2, дисперсия имеет постоянное значение "1".).)

Периодическая обработка

Процесс, описанный на рисунке 6, инициирован однажды секунда. Он изучает показатели таймеров в таблице маршрутов, чтобы найти таймеры, для которых истек срок действия. Описание этих таймеров приведено выше.

На шаге U активизируется процесс, описанный на рис. 7.

Шаг R и шаг S являются необходимыми, поскольку составные метрики, хранящиеся в таблице маршрутизации, зависят от величины занятости канала, которая изменяется в результате измерений. Загруженность канала периодически пересчитывается с использованием скользящее среднее измеренного трафика через интерфейс. Если заново рассчитанное значение отличается от существующего, необходимо настроить соответствующим образом все составные метрики, включающие этот интерфейс. Проверяется каждый путь, показанный в таблице маршрутизации. Для любого пути, следующий переход которого использует интерфейс "I", осуществляется пересчет составной метрики. Это сделано в соответствии с уравнением 1, используя в качестве заполнения канала максимальное значение, находящееся в таблице маршрутизации как часть метрики пути, а также заново рассчитанное заполнение канала интерфейса.

Генерирование сообщений об обновлении

На рисунке 7 объясняется, каким образом шлюз генерирует сообщения обновлений, которые должны быть переданы другим шлюзам. Отдельное сообщение генерируется для каждого сетевого интерфейса, подключенного к шлюзу. Это сообщение затем передается всем остальным шлюзам, доступным через интерфейс (шаг J). В общем случае это реализуется через отправку сообщения как широковещательного. Однако, если сетевая технология или протокол не допускает широковещательных рассылок, может потребоваться отправить сообщение индивидуально на каждый шлюз.

В целом сообщение создано путем добавления записи для каждого назначения в таблице маршрутизации в Шаге G. Обратите внимание на то, что назначение/данные пути, привязанное к каждому типу сервиса, должно использоваться. В самом худшем случае, новая запись добавляется в обновление для каждого пункта назначения каждого типа службы. Однако перед добавлением записи в сообщение обновления на шаге G проверяются уже добавленные записи. Если новая запись уже присутствует в обновленном сообщении, она не добавлена снова. Когда назначения и шлюзы следующего перехода являются тем же, новая запись копирует существующий.

Ради простоты псевдокод опускает одну вещь — сообщения обновления IGRP имеют три части: внутренняя часть, система и внешний вид, что означает, что существует фактически три петли по назначениям. Первое включает только подсети сети, которой передается обновление. Второе включает все крупные сети (например, неподсети), которые не отмечены как внешний вид. Третье включает все крупные сети, которые отмечены как внешний вид.

На шаге E выполняется тест на расщепленный горизонт. Данный тест обычно не проходят маршруты, оптимальный путь которых начинается с интерфейса, с которого отсылается обновление. Однако, если обновление передается в конкретную точку назначения (например в ответ на запрос IGRP от другого шлюза или как часть двухточечного соединения IGRP), сбой при разделении горизонтов происходит только в том случае, если оптимальный путь получен из точки назначения ("источник информации" совпадает с точкой назначения) и выходной интерфейс совпадает с интерфейсом, от которого был получен запрос.

Расчет метрических данных

Рисунок 8 описывает, как обрабатывается метрическая информация из сообщений обновления, полученных шлюзом, и как она генерируется для сообщений обновления, посланных шлюзом. Обратите внимание на то, что запись основана на одном конкретном маршруте до конечной точки. Если существует несколько путей к назначению, выбран путь, составная метрика которого минимальна. Если больше одного пути имеет минимальную композитную метрику, то используется правило арбитра конфликта. (Для большинства протоколов это основано на адресе шлюза следующего узла.).)

Рисунок 4 — обработка входящих пакетов

```
Data packet arrives using interface I

A   Determine protocol used by packet

    If protocol is not supported
      then discard packet

B   If destination address matches any of gateway's addresses
    or the broadcast address
      then process packet in protocol-specific way

C   If destination is on a directly-connected network
      then send packet direct to the destination, using
      the encapsulation appropriate to the protocol and link type

D   If there are no paths to the destination in the routing
    table, or all paths are upstream
      then send protocol-specific error message and discard the packet

E   Choose the next path to use. If there are more than
    one, alternate round-robin with frequency proportional
    to inverse of composite metric.

    Get next hop from path chosen in previous step.

    Send packet to next hop, using encapsulation appropriate
    to protocol and data link type.
```

Рисунок 5 — обработка обновлений входящей маршрутизации

```
Routing update arrives from source S

    For each type of service supported by gateway
      Use routing data associated with this type of service

      For each destination D shown in update

A       If D is unacceptable or in holddown
          then ignore this entry and continue loop with next destination D

B       Compute metrics for path P to D via S (see Fig 8)

        If destination D is not already in the routing table
        then Begin

          Add path P to the routing table, setting last
          update times for P and D to current time.

H         Trigger an update

          Set composite metric for D and P to new composite
          metric computed in step B.

          End

        Else begin (dest. D is already in routing table)

K         Compare the new composite metric for P with best
          existing metric for D.

           New > old:

L            If D is shown as unreachable in the update,
             or holddowns are enabled and
              the new composite metric >
                (the existing metric for D) * V
                [use 1.1 instead of V if V = 1,
                as it is as of Cisco release 8.2]
O            or holddowns are disabled and
             P has a new hop count > old hop count
             then Begin

              Remove P from routing table if present

              If P was the last route to D
                then Unless holddowns are disabled
                     Set holddown time for D to
                      current time + holddown time
T                   and Trigger an update

              End

             else Begin

              Compute new best composite metric for D

              Put the new metric information into the
              entry for P in the routing table

              Add path P to the routing table if it
              was not present.

              Set last update times for P and D to
              current time.
```

```
              End

           New <= OLD:

V            Set composite metric for D and P to new
             composite metric computed in step B.

             If any other paths to D are now outside the
             variance, remove them.

             Put the new metric information into the
             entry for P in the routing table

             Set last update times for P and D to
             current time.

          End

        End of for

      End of for
```

Рисунок 6 — периодическая обработка

```
Process is activated by regular clock, e.g. once per second

    For each path P in the routing table (except directly
    connected interfaces)

      If current time < P'S LAST UPDATE TIME + INVALID TIME
        THEN CONTINUE WITH THE NEXT PATH P

      Remove P from routing table

      If P was the last route to D
        then Set metric for D to inaccessible
          Unless holddowns are disabled,
            Start holddown timer for D and
          Trigger an update

        else Recompute the best metric for D

      End of for

    For each destination D in the routing table

      If D's metric is inaccessible
        then Begin

        Clear all paths to D

        If current time >= D's last update time + flush time
          then Remove entry for D

        End

      End of for

    For each network interface I attached to the gateway

R     Recompute channel occupancy and error rate

S     If channel occupancy or error rate has changed,
```

```
        then recompute metrics

      End of for

    At intervals of broadcast time

U     Trigger update
```

Рисунок 7 — генерирует обновление

```
Process is caused by "trigger update"

    For each network interface I attached to the gateway

      Create empty update message

      For each type of service S supported

        Use path/destination data for S

        For each destination D

E         If any paths to D have a next hop reached through I
            then continue with the next destination

          If any paths to D with minimal composite metric are
          already in the update message
            then continue with the next destination

G         Create an entry for D in the update message, using
          metric information from a path with minimal
          composite metric (see Fig. 8)

          End of for

        End of for

J     If there are any entries in the update message
       then send it out interface I

      End of for
```

Рисунок 8 — подробные данные метрических расчетов

В этом разделе описываются процедуру для вычислений метрики и счетчиков переходов от поступающего обновления маршрута. Ввод к этой функции является записью для определенного назначения в пакете обновления маршрутизации. Выходные данные являются вектором метрик, которые могут использоваться для вычислений составной метрики и счетчика переходов. Если этот путь будет добавлен в таблицу маршрутизации, в таблицу будет внесен весь вектор метрики. Параметры интерфейса, используемые в следующих определениях, установлены таковыми при инициализации шлюза для интерфейса, на который поступило обновление маршрутизации, за исключением того, что заполнение и надежность канала основаны на скользящем среднем измеренного трафика, прошедшего через интерфейс.

- Задержка = задерживается от пакета + интерфейсная топологическая задержка
- Пропускная способность = max (пропускная способность пакета, пропускная способность интерфейса)
- Reliability = min (надежность из пакета, надежность интерфейса)
- Channel occupancy = max (занятость канала от пакета, занятость канала

интерфейса)(Max используется для пропускной способности, потому что метрика пропускной способности сохранена в обратной форме. Концептуально, мы хотим минимальную пропускную способность.) Обратите внимание на то, что исходное заполнение канала от пакета должно быть сохранено, так как будет необходимо повторно вычислить эффективное заполнение канала каждый раз, когда изменяется заполнение канала интерфейса.

Приведенное ниже не является частью вектора метрики, но также хранится в таблице маршрутизации в качестве характеристики пути:

- Счетчик переходов = счетчик переходов от пакета.
- MTU = min (MTU из пакета, интерфейс MTU).
- Удаленная составная метрика = вычислила от Уравнения 1 использование значений метрики от пакета. То есть, компонентами метрики являются компоненты из пакета, они не обновляются, как показано выше. Очевидно, это должно быть вычислено, прежде чем корректировки, показанные выше, сделаны.
- Составная метрика определяется по уравнению 1 с использованием значений метрики, которые были вычислены способом, описанным в данном разделе.

Последняя часть раздела описывает процедуру вычисления метрики и числа переходов для отправляемых обновлений маршрутов.

Данная функция определяет данные метрик и число переходов, добавляемые в исходящий обновленный пакет. Если существуют какие-либо доступные пути, это основывается на определенном пути к назначению. Если путей нет или все пути идут в направлении, противоположном основному трафику, то получатель называется "недостижимым".

```
If destination is inaccessible, this is indicated by using a specific
        value in the delay field. This value is chosen to be larger
        than the largest valid delay. For the IP implementation this is
        all ones in a 24-bit field.

   If destination is directly reachable through one of the interfaces, use
        the delay, bandwidth, reliability, and channel occupancy of the
        interface. Set hop count to 0.

   Otherwise, use the vector of metrics associated with the path in the
        routing table. Add one to the hop count from the path in the
        routing table.
```

Подробности IP-установки

Данный раздел содержит краткое описание форматов пакетов, используемых Cisco IGRP. IGRP передается с помощью дейтаграмм IP с Протоколом "IP" 9 (IGP). Пакет начинается с заголовка. Это сразу запускается после IP - заголовка.

```
unsigned version: 4;  /* protocol version number */
    unsigned opcode: 4;   /* opcode */
    uchar edition;        /* edition number */
    ushort asystem;       /* autonomous system number */
    ushort ninterior;     /* number of subnets in local net */
    ushort nsystem;       /* number of networks in AS */
    ushort nexterior;     /* number of networks outside AS */
    ushort checksum;      /* checksum of IGRP header and data */
```

Сведения маршрутизации в сообщениях обновления следуют непосредственно после заголовка.

Номер версии в настоящее время равняется 1. Пакеты, имеющие другие номера версий, проигнорированы.

Код операции может быть 1 = обновление или 2 = запрос.

Это указывает на тип сообщения. Формат двух типов сообщений будет приведен ниже.

Версия – порядковый номер, увеличивающийся с каждым изменением таблицы маршрутизации. (Это сделано в тех условиях, в которых псевдокод выше говорит для инициирования обновления маршрута.) Номер издания позволяет шлюзам избегать обрабатывать обновления, содержащие информацию, которую они уже видели. (Это в настоящее время не внедряется. Т.е. номер издания генерируется правильно, но он проигнорирован на вводе. Поскольку существует вероятность отбрасывания пакетов, нет уверенности в том, что номер издания достаточен для избежания двойной обработки. Необходимо убедиться, что были обработаны все пакеты, соответствующие изданию).)

Asystem — это номер автономной системы. Во внедрении Cisco шлюз может участвовать в нескольких автономных системах. В каждой такой системе используется собственный протокол IGRP. Концептуально, существуют абсолютно отдельные таблицы маршрутизации для каждой автономной системы. Маршруты, которые поступают через IGRP от одной автономной системы, передаются только в обновлениях для этой AS. Это поле позволяет шлюзу выбрать набор таблиц маршрутизации, используемый для обработки этого сообщения. Если шлюз получает сообщение IGRP для AS, которая не настроена для этого, сообщение игнорируется. Фактически, реализация Cisco допускает "утечку" информации с одного AS на другой. Однако я считаю эту AS средством администрирования, а не частью протокола.

Ninterior, nsystem и nexterior указывают на количество записей в каждой из трех частей сообщений об обновлении. Эти разделы описаны выше. Между разделами нет никакого другого разграничения. Первая запись ninterior рассматривается как внутренняя, запись nsystem – как системная, а запись nexterior – как внешняя.

Контрольная сумма IP рассчитывается с помощью того же алгоритма, что и контрольная сумма UDP. Контрольная сумма вычисляется в заголовке IGRP и любая информация маршрутизации, которая следует за ней. Поле контрольной суммы обнулено при вычислениях контрольной суммы. Контрольная сумма не включает IP - заголовок, и при этом нет никакого виртуального заголовка как в UDP и TCP.

Запросы

IGRP - запрос просит, чтобы получатель передал его таблицу маршрутизации. Сообщение запроса имеет только заголовок. Только версия, opcode и asystem поля используются. Все другие поля являются нулем. Получатель должен отправить запрашивающей стороне обычное сообщение об обновлении IGRP.

Обновления

Сообщение обновления IGRP содержит заголовок, придерживавшийся сразу записями маршрутизации. Включается столько записей маршрутизации, сколько поместится в датаграмме размером 1500 байт (включая IP-заголовок). С объявлениями текущей структуры это позволяет до 104 записей. При необходимости дополнительных записей выполняется отправка нескольких сообщений об обновлении. Поскольку сообщения об

обновлениях обрабатываются по записям, использование одного фрагментированного сообщения вместо нескольких независимых сообщений не дает преимущества.

Вот структура записи маршрутизации:

```
uchar number[3];       /* 3 significant octets of IP address */
    uchar delay[3];        /* delay, in tens of microseconds */
    uchar bandwidth[3];    /* bandwidth, in units of 1 Kbit/sec */
    uchar mtu[2];          /* MTU, in octets */
    uchar reliability;     /* percent packets successfully tx/rx */
    uchar load;            /* percent of channel occupied */
    uchar hopcount;        /* hop count */
```

Поля, определяемые как uchar[2] и uchar[3], являются просто 16 и 24- разрядными двоичными целыми числами, в обычном порядке сети IP.

Число определяет описываемый адрес назначения. Это IP-адрес. Для экономии места даются только первые три байта IP-адреса, за исключением внутреннего раздела. Во внутреннем разделе приведены последние три байта. Для систем и внешних маршрутов недоступны подсети, поэтому младший байт всегда равен нулю. Внутренними маршрутами всегда являются подсети известной сети, следовательно, предоставляется первый байт номера этой сети.

Задержка исчисляется 10 микросекундами. Таким образом, будет выделен достаточный временной интервал от 10 мкс до 168 с. Задержка для всех пакетов указывает, что сеть недоступна.

Bandwidth - величина обратная пропускной способности в битах в секунду, умноженная на масштабирующий коэффициент 1,0e10. Диапазон скорости канала составляет от 1200 бит/с до 10 Гбит/с. (Т. е. при пропускной способности N кбит/с, используется число 10000000/N.).)

MTU в байтах.

Надежность дана как часть 255. Т.е. 255 100%.

Нагрузка задается в долях от 255.

Счетчик переходов является простым количеством.

Поскольку для пропускной способности и задержки использовались несколько необычные единицы измерения, для наглядности приведены некоторые примеры. Это значения по умолчанию, используемые для нескольких стандартных средств.

	Delay	Bandwidth
Satellite	200,000 (2 sec)	20 (500 Mbit)
Ethernet	100 (1 ms)	1,000
1.544 Mbit	2000 (20 ms)	6,476
64 Kbit	2000	156,250
56 Kbit	2000	178,571
10 Kbit	2000	1,000,000
1 Kbit	2000	10,000,000

Расчет метрик

Здесь представлено описание способа фактического вычисления составной метрики в Cisco версии 8.0(3).

```
metric = [K1*bandwidth + (K2*bandwidth)/(256 - load) + K3*delay] *
                        [K5/(reliability + K4)]

    If K5 == 0, the reliability term is not included.

    The default version of IGRP has K1 == K3 == 1, K2 == K4 == K5 == 0
```

Дополнительные сведения

- Страница поддержки IP-маршрутизации
- Страница поддержки IGRP
- Техническая поддержка - Cisco Systems