RIP. Версия 2.

Данный документ содержит описание версии 2 протокола маршрутизации RIP для сетей IP, основанное на RFC 2453. Автор RFC 2453: G. Malkin, Bay Networks. Дата: Ноябрь 1998.

Содержание

1 От переводчика

Данный документ не является попыткой изложить материал в удобоваримом виде, т. е. таким образом, чтобы его было удобно читать и понимать неспециалисту. Такая обработка не проводилась. Документ является, если так можно выразиться, литературным переводом первоисточника. Последовательность изложения материала и терминология по возможности сохранены. В то же время некоторые организационные данные (не относящиеся к собственно описанию) не включены в данный документ. Желающие получить эти данные могут обратиться к оригиналу.

Англоязычные названия параметров будут переводиться, но использоваться далее в тексте как правило в первоначальном виде. Это объясняется тем, что вероятность встретить русскоязычные обозначения при настройке какого-либо устройства достаточно мала.

При чтении данного документа необходимо иметь общие знания о функционировании протокола IP, а также о том, что IP используется для передачи данных такого протокола, как UDP.

2 Введение

RIP является протоколом маршрутизации, использующимся в сетях протокола IP. Отметим, что протокол с таким же названием используется и в IPX-сетях. Принцип действия протоколов одинаковый, но работают они в сетях разных протоколов и путать их не надо.

Протокол RIP принадлежит к классу так называемых IGP протоколов – Interior Gateway Protocol. Протоколы класса IGP, такие, как RIP или OSPF, используются как правило внутри **AS (Autonomous System – Автономная система)**. Под AS понимается совокупность устройств, принимающих участие в работе одного протокола и находящихся под единым административным управлением. Как следует из определения, в пределах AS используется как правило один тип протокола маршрутизации.

Примером AS может быть сеть одной организации, использующей несколько маршрутизаторов для связи офисов. Внутри такой сети может использоваться RIP. Другой пример – построение крупной локальной сети на основе маршрутизирующих коммутаторов, на которых поднят OSPF. Выход в глобальные сети осуществляется через маршрутизатор, работающий с внешними сетями с помощью других протоколов маршрутизации. Сеть, образованная маршрутизирующими коммутаторами и маршрутизатором тоже является AS.

3 Основы протокола

3.1 Введение

RIP является протоколом маршрутизации, основанным на алгоритме Беллмана-Форда (Bellman-Ford algorithm), или distance-vector алгоритме (distance-vector – вектор-дистанция). Алгоритм на данный момент является уже, так сказать, заслуженным. Он использовался для вычисления маршрутной информации в сети ARPANET с первых дней ее существования.

Необходимо заметить, что для больших сетей использование для всех маршрутизаторов одного и того же протокола маршрутизации не совсем оправдано. Например, Internet. Использование всеми маршрутизаторами RIP'а может привести к прекращению функционирования Internet'а, поскольку вся сеть будет перегружена служебными сообщениями.

Вследствие данного утверждения крупные сети как правило разделяются на некие области, так называемые AS (Autonomous System – Автономная система), каждая из которых администрируется отдельно. Каждая AS может внутри себя использовать свой протокол маршрутизации. Протоколы маршрутизации, используемые внутри AS, относятся к классу Interior Gateway Protocol - IGP. Протоколы маршрутизации, которые используются AS для обмена маршрутной информацией между собой, относятся к классу Exterior Gateway Protocol – EGP. RIP относится к классу IGP-протоколов.

По принципу принятия решения RIP относится к классу протоколов distance-vector. Принцип работы таких протоколов будет рассмотрен в разделе 3.4. :Желающие знать историю развития алгоритма могут обратиться к RFC 2453 и соответствующей литературе.

RIP разрабатывался для работы в IP-сетях, т. е. в сетях, использующих в качестве протокола сетевого уровня IP и объединяемых с помощью активных сетевых устройств, определяемых как маршрутизаторы. IP-сети могут использовать различные сетевые технологии канального уровня, такие, как Ethernet, Token Ring, линии связи точка-точка (PPP) и так далее. Предполагается, что хосты и маршрутизаторы IP-сетей генерируют и/или пересылают IP-пакеты. **Маршрутизация есть метод, с помощью которого хост или маршрутизатор решает, куда должен быть послан пакет для достижения своего места назначения (получателя).** Если получателем является хост в той же сети, в которой находится IP-интерфейс источника, то пакет посылается непосредственно получателю. Если же хост находится в другой IP-сети, хост или маршрутизатор должен послать пакет тому маршрутизатору, который обеспечит кратчайший путь передачи данных. Цель протокола маршрутизации состоит в том, чтобы хосты и маршрутизаторы имели информацию, необходимую для принятия правильного маршрутного решения.

3.2 Ограничения протокола

RIP не является универсальным протоколом маршрутизации и не может использоваться в IP-сети любого размера и сложности. Т. е. использовать его можно, но при определенных размерах сети это может стать проблемой. Напомним, что RIP есть IGP-протокол. RIP имеет следующие, характерные именно для него ограничения:

- Протокол ограничивает диаметр сети 15-тью хопами (hop) самого длинного маршрута. Что такое hop? В данном случае этот термин можно определить как сеть – в том случае, если cost каждой сети равен 1. Что значит вышеприведенное утверждение? Оно значит, что между двумя любыми хостами сети не должно быть более 15 сетей. Таким образом, для больших сетей RIP не годиться. Что такое cost? Cost есть параметр, назначаемый каждой сети сетевым администратором. Использование cost'а рассмотрено ниже, раздел *3.4.1.4 Функции участника алгоритма*.
- Для больших сетей может стать проблемой образование циклических соединений (loop), или **циклов**. Более подробно эта ситуация рассмотрена в следующем разделе. RIP имеет механизмы разрешения ситуации с циклами, но в больших сетях эти механизмы могут отрабатываться достаточно большое время, при этом занимая полосу пропускания сети. Несмотря на то, что подобная утилизация будет заметна только на линиях с небольшим скоростями передачи данных, таких, как модемные, возникновение таких ситуаций крайне нежелательна.
- Для сравнения двух маршрутов к одной и той же сети используется «метрика» (metrics). В тех ситуациях, когда выбор маршрута должен основываться на таких параметрах, как скорость передачи, надежность линка, доступная полоса пропускания - такой критерий помогает не сильно. В принципе, метриками можно «поиграть», но в любом случае протокол не поддерживает механизмов отслеживания реальных параметров линии при выборе маршрута передачи данных.

3.3 Структура третьей главы

Данная глава содержит две части:

- Описание distance-vector алгоритма.
- Описание работы протокола.

Перечисленные разделы друг от друга не зависят и могут изучаться отдельно. Однако необходимо отметить, что некоторые механизмы, на которых основан RIP, обсуждаются именно в первом разделе.

3.4 Описание distance-vector алгоритма

Необходимо отметить, что данный раздел не совсем соответствует аналогичному в RFC 2453. RFC 2453 содержит некое количество текста, собственно к алгоритму отношения не имеющего – как то немного истории, немного общих рассуждений о том, что есть маршрутизация. Здесь такие изречения опущены.

Термин «сеть», используемый в данном разделе, как правило будет использоваться для определения broadcast-домена (широковещательного домена). Реально говоря, под «сетью» будем понимать совокупность устройств, работающих в одной IP-сети/подсети, имеющих общую маску и способных отрабатывать между собой протокол распознавания адреса – ARP. Последний использует broadcast-сообщения. Примерами «сетей» могут стать сеть Ethernet, Token Ring, сеть точка-точка и т. д.

3.4.1 Общие понятия

3.4.1.1 Определение

Маршрутизаторы могут использовать несколько подходов, или алгоритмов для вычисления маршрутов передачи данных между сетями. Один из возможных способов классификации таких алгоритмов основывается на том, какой информацией обмениваются маршрутизаторы в целях поиска правильных маршрутов. Distance-vector алгоритм основан на обмене сравнительно небольшим количеством информации. Каждый маршрутизатор внутри AS хранит информацию о всех сетях внутри AS. Эта информация организована в отдельную маршрутную базу данных. Каждая запись в этой базе данных содержит информацию об одной из сетей AS. Каждая запись включает в себя адрес следующего маршрутизатора, т. е. маршрутизатора, которому должна быть послана информация для того, чтобы она достигла соответствующей сети. Кроме того, каждая запись содержит так называемую «метрику». Метрика указывает «дистанцию» до сети. Дистанция является понятием довольно общим, т. е. может выражать время, необходимое пакету для достижения сети, или стоимость передачи пакета – при использовании платных линий связи. **Алгоритм distance-vector получил свое название из-за того, что рассчитывает маршрут к назначению (vector) на основе дистанции (distance).**

Как правило, концепция маршрутизации рассматривает передачу информации между сетями. Иногда может появиться необходимость хранить информацию о маршруте доступа к конкретному хосту. RIP в принципе не делает различия между сетями и хостами. Определяется маршрут к назначению, которое может быть или сетью, или хостом.

3.4.1.2 Таблица маршрутизации. Определение.

Как говорилось ранее, каждый маршрутизатор поддерживает базу данных маршрутов, содержащую информацию о том, как можно передать данные каждой сети внутри AS. Такая база данных называется таблицей маршрутизации и содержит следующие поля для каждой записи (одна запись – одна сеть назначения):

- Адрес. Адрес IP-адрес хоста или сети назначения.
- Маршрутизатор. Первый маршрутизатор «по дороге» к сети/хосту назначения. Тот маршрутизатор, которому должны быть посланы данные для того, чтобы они достигли сети/хоста назначения.
- Интерфейс. Физический интерфейс, через который должны быть отправлены данные *маршрутизатору*.
- Метрика (metric). Число, указывающее дистанцию до сети назначения.
- Таймер. Время, указывающее как давно обновлялась данная запись.

Записи таблицы маршрутизации могут содержать дополнительную информацию, такую, как различного рода флаги. Таблица маршрутизации инициализируется информацией о локально (непосредственно) подключенных сетях. Обновление/добавление записей происходит по мере получения сообщений от других маршрутизаторов.

3.4.1.3 Формирование таблицы маршрутизации.

Информация, необходимая для формирования в таблице маршрутизации записей о удаленных сетях, содержится в сообщениях, которыми обмениваются хосты и маршрутизаторы. Эти сообщения будем называть update-сообщениями (сообщения корректировки). Каждое устройство, участвующее в протоколе маршрутизации, рассылает такие сообщения, несущие информацию о всех известных устройству сетях. Получая такую информацию от всех своих соседей, маршрутизатор в состоянии вычислить оптимальный маршрут к любой из сетей. В случае RIP'а и distance-vector алгоритма оптимальный маршрут выбирается на основании метрики.

В простых и относительно небольших сетях метрика может рассчитываться как количество маршрутизаторов от источника до получателя. В более комплексных сетях метрики желательно должны отражать большее количество параметров, таких, как скорость линков между устройствами, стоимость передачи данных и т. д.

3.4.1.4 Функции участника алгоритма.

Напомним, что несмотря на то, что RIP основан на distance-vector алгоритме, между ними существуют некоторые отличия. Об этом необходимо помнить при прочтении материала данного раздела.

Ниже перечислены функции, которые должен выполнять участник distance-vector алгоритма. В качестве таких участников должны выступать все маршрутизаторы AS. Как говорилось выше, в качестве участников могут выступать также хосты.

Итак, участник алгоритма должен:

- Поддерживать таблицу маршрутизации, имеющую запись для каждой сети в AS. В принципе, должна даже поддерживаться запись с указанием на себя самого с метрикой 0. Но в реальной жизни эта концепция не нашла отражения.
- Периодически посылать update-сообщения всем своим соседям. Update состоит из одного сообщения (или из нескольких, если в одно все не влезло) и содержит все записи таблицы маршрутизации. Для каждой записи должны указываться IP-адрес назначения и соответствующая метрика.
- При получении update'а от маршрутизатора **R** добавлять к метрикам update'а некий параметр cost (цена), ассоциированный с сетью, которая разделяется с оным маршрутизатором. Другими словами, с какой-либо непосредственно подключенной сети получена информация с соответствующими метриками. Перед анализом и обработкой информации ко всем метрикам прибавить cost сети, с которой информация получена.
  Полученную информацию сравнить с содержащейся в таблице маршрутизации. Если полученная информация указывает, что для какой-либо существующей сети получена меньшая метрика, изменить таблицу маршрутизации. Если в таблице маршрутизации содержится запись, полученная от маршрутизатора R, и от R получена новая информация по поводу этой записи (та же сеть, но метрика изменилась), то запись в таблице маршрутизации будет изменена в соответствии с полученной информацией, даже если метрика стала больше.

3.4.2 Обработка изменений в топологии

Вышеизложенное подразумевает, что мы установили маршрутизаторы, соединили их некими линиями связи, и все остается стабильным. В реальной жизни это не совсем так. Маршрутизаторы могут падать (имеется в виду прекращаться функционировать), линии связи тоже, так что необходимо предусмотреть возможность таких ситуаций.

В идеале все хорошо. При изменении топологии сети изменяется набор соседей (соседних маршрутизаторов). Например, один из них больше не работает. При следующем перерасчете маршрутов те маршруты, которые вели через него, надо просто пересчитать. В идеале.

Возвращаясь в реальность, необходимо учесть, что каждый маршрутизатор хранит в памяти только **один** маршрут к какой-либо сети, лучший с его точки зрения. Алгоритм предполагает пересчет маршрута в случае изменения метрики. Т. е. маршрутизатор, через который более нельзя достичь какой-либо сети назначения, должен прислать маршрут с изменившейся метрикой. А если этот маршрутизатор упал? Нет информации, нет изменения метрики, не пересчета маршрута.

Дабы избежать подобной ситуации, протокол, основанный на distance-vector алгоритме должен содержать механизм устаревания (timing out) маршрутов. Детали такого механизма относятся к спецификации собственно протокола. Например, RIP посылает update'ы каждые 30 секунд. Следовательно, маршрутизатор получает информацию о сети N каждые 30 секунд. Если что-то случилось, и он не получит информацию о сети N в течении 180 секунд, он будет считать, что сеть более не достижима. Такой относительно большой таймаут выбирается по тому, что update'ы могут быть потеряны линком между маршрутизаторами, или быть по каким-либо причинам задержанными.

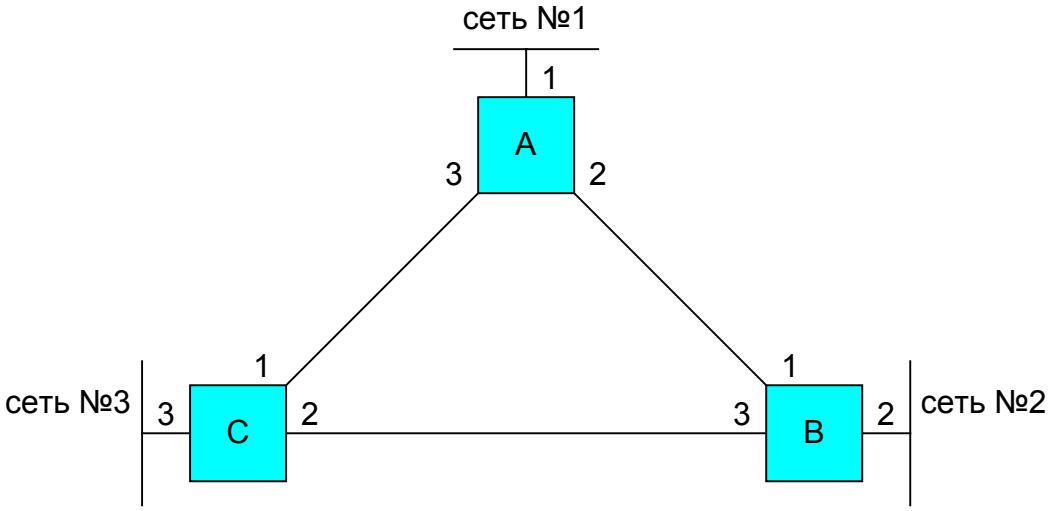
Маршрутизатор может не только определить на основании механизма устаревания, что сеть более недоступна, но и поделиться этой информацией с другими маршрутизаторами. RIP делает это с помощью стандартных update-сообщений. Сети, более не доступные, помечаются в таких сообщениях как «недоступные» (unreachable). Для индикации «недоступности» сети используется метрика 16. Эта метрика в RIP называется «бесконечность» (infinity) и является большей, чем наибольшая доступная к использованию в протоколе метрика. Другими словами, сеть, имеющая метрику 16, является недоступной и пакеты в такую сеть посланы быть не могут.

3.4.3 Обеспечение стабильности

Пока еще вышеописанный механизм немного далек от реальной жизни и не обеспечивает своевременного и корректного пересчета таблиц маршрутизации. Вышеописанный механизм обеспечивает корректировку таблиц маршрутизации в некое конечное время. Однако неизвестно, действительно ли это конечное время будет настолько малым, что это будет приемлемо для реально существующих сетей. Кроме того, большое время пересчета таблиц может привести к несоответствиям в работе алгоритма в масштабе достаточно большой сети.

В случае отключения какой-либо сети или маршрутизатора, как мы видели выше, по происшествии некоторого времени (таймаута) соседние с отключенной компонентой маршрутизаторы помечают ее в своих таблицах маршрутизации как недоступную, т. е. присваивается метрика 16. Далее эти маршрутизаторы будут рассылать update'ы с записью, содержащей метрику 16. Получив update, содержащий запись с метрикой 16, маршрутизатор не будет, как обычно, добавлять к полученной метрике cost сети (см. раздел *3.4.1.4 Функции участника алгоритма*). Метрика и так уже «недостижима», чего же более.

Таким образом все маршрутизаторы сети через некоторое время узнают, что одной из сетей больше не существует, т. е. она не доступна. Однако это некое время может оказаться слишком большим. Вообще, все это может привести к неприятностям. Каким? Рассмотрим на примере.



В нашем примере будет рассматриваться случай «падения» сети №2. Каким образом это произойдет технически – дело двадцать пятое и здесь не рассматривается.

В нижеприведенной таблице показаны те записи таблиц маршрутизации, которые относятся к сети №2 (в соответствии с пунктом *3.4.1.3 Таблица маршрутизации. Определение.*).

Маршрутизатор	Адрес	Следующий маршрутизатор	Интерфейс	Метрика
A	сеть №2	B	2	2
B	сеть №2	-	2	1
C	сеть №2	B	2	2

Таймер для рассматриваемого примера значения не имеет.

Рассмотрим последовательность событий. Для простоты будем считать, что все маршрутизаторы посылают свои update'ы одновременно. Cost всех сетей равен 1.

- Сеть 2 отваливается, маршрутизатор B это событие отлавливает и устанавливает в записи для сети 2 метрику 16.

- Все маршрутизаторы посылают update'ы. B посылает: сеть №2, метрика 16. A и C посылают: сеть №2, метрика 2.
- Получение update'ов.
  - Маршрутизатор A:
    получено от B сеть №2, метрика 16. Через B сеть №2 более не достижима.
    получено от C сеть №2, метрика 2. Добавить 1, получим 3. Это лучше, чем 16. Таким образом, сеть №2 теперь достижима через C.
  - Маршрутизатор C:
    получено от B сеть №2, метрика 16. Через B сеть №2 более не достижима.
    получено от A сеть №2, метрика 2. Добавить 1, получим 3. Это лучше, чем 16. Таким образом, сеть №2 теперь достижима через A.
  - Маршрутизатор B:
    получено от C сеть №2, метрика 2. Добавить 1, получим 3. Следовательно, сеть №2 теперь не достижима через непосредственно подключенный интерфейс, но достижима через C с метрикой 3! Очень хорошо, теперь будем посылать пакеты в сеть №2 через C.
    получено от A сеть №2, метрика 2. Добавить 1, получим 3. Такая сеть с такой метрикой уже есть, оставим то, что есть.
- В результате мы имеем

Маршрутизатор	Адрес	Следующий маршрутизатор	Интерфейс	Метрика
A	сеть №2	C	3	3
B	сеть №2	C	3	3
C	сеть №2	A	1	3

При дальнейшем обмене update'ами метрики будут увеличиваться. Поясним это на примере маршрутизатора C.

Запись о сети №2 сформирована на основании информации от маршрутизатора A и содержит метрику 3. Однако и у маршрутизатора A метрика для сети №2 равна 3. Теперь пошел update от A к C, содержащий запись: сеть №2, метрика 3. C получил, прибавил cost (1), получил 4. По правилам алгоритма он должен заменить запись для сети №2 полученной информацией (*3.4.1.4 Функции участника алгоритма*). Метрика возрастает.

Вышесказанное будет справедливо для остальных участников событий – маршрутизаторов B и A. Таким образом, метрики для сети №2 будут возрастать по мере обмена update'ами, и в конце концов достигнут 16. Маршрутизаторы наконец-то поймут, что сеть отключена. Однако это займет слишком долгое время.

Для предотвращения подобных ситуаций используются некоторые механизмы. RIP использует следующие: «split horizon with poisoned reverse» (разделение горизонта с отравлением обратного направления.) и «triggered update» (запуск update'а). Перевод здесь приведен дословный и далее будет использоваться англоязычное определение механизмов.

3.4.4 Split horizon

Часть вышерассмотренных проблем может быть разрешена с помощью более тщательного отношения к тому, куда какая информация посылается. Т. е. нет необходимости посылать информацию о сети тому маршрутизатору, который информацию об этой же сети и прислал. В нашем примере маршрутизатор A не должен посылать маршрутизатору B от него же полученную информацию о сети №2. **«Split horizon» - это механизм, препятствующий посылке информации тому маршрутизатору, от которого эта информация получена.**

Механизм имеет два варианта реализации. Первый «simple split horizon», или «split horizon» заключается в том, что информация не посылается тому маршрутизатору, от которого она получена. В нашем примере: маршрутизатор A не будет посылать информацию о сети №2 маршрутизатору B.

Второй вариант называется «split horizon with poisoned reverse». Отличается от первого тем, что информация посылается тому маршрутизатору, от которого она была получена, но! В качестве метрики используется 16 – т. е. «недостижима». В нашем примере: маршрутизатор A посылает информацию о сети №2 маршрутизатору B с метрикой 16.

Выбор используемого в той или иной ситуации способа остается на совести сетевого администратора. Однако необходимо отметить два момента, которые следует учитывать при выборе:

1. При наличии в сети циклов и использовании механизма «split horizon with poisoned reverse» отработка изменений в топологии будет происходить быстрее в силу того, что маршрутизаторы будут получать информацию о «недостижимости» сети друг от друга достаточно быстро. При использовании «split horizon» неправильные маршруты будут исключаться только по происшествии таймаута.
2. Механизм «split horizon with poisoned reverse» подразумевает внесение дополнительной информации в update-сообщения по сравнению со«split horizon». В большой сети с большим количеством сетей и маршрутизаторов и наличием линков с небольшой скоростью это может оказаться ощутимым. Второй вопрос, стоит ли использовать в такой сети RIP?

3.4.5 Triggered update

Механизм «triggered update» разработан в целях ускорения процесса отработки изменений сетевых маршрутов. Механизм прост. В том случае, если маршрутизатор получает информацию о изменении конфигурации сети – перестал функционировать собственный интерфейс, пришел update, из-за которого пришлось изменить таблицу маршрутизации, мор, глад, семь казней египетских (последние три не учитываются) – в таких случаях маршрутизатор не ожидает очередного срока посылки update'а, а посылает update немедленно. Т. е. не то чтобы совсем немедленно, маршрутизатор ожидает некое небольшое случайное время. Делается это для того, чтобы избежать одновременного шторма update'ов в пределах сети.

Реальная жизнь может вносить коррективы в безупречную работу алгоритма. Например, даже в случае использования «triggered update» может оказаться так, что у каких-либо маршрутизаторов, еще не получивших новую информацию, наступает время посылки регулярного update'а. Ничего не зная о происходящем, такой маршрутизатор выдает всем окружающим уже устаревшую информацию. Несмотря на то, что при «triggered update» задержка посылки update'а достаточно мала, такая ситуация возможна. Ничего плохого в ней нет, однако время отработки изменений в сети в таком случае увеличиться. Такие случаи надо даже не то, чтобы учитывать, но помнить о том, что они возможны.

3.5 Спецификация протокола

Итак, что есть RIP? RIP есть протокол маршрутизации, используемый для расчета маршрутов в сетях IPv4. Подразумевается, что любой маршрутизатор – участник RIP имеет один или более функционирующих интерфейсов (к которым подключены IP-сети). Такие сети определяются как непосредственно подключенные – **directly connected**. Каждый маршрутизатор рассылает информацию о известных ему сетях, и соответственно, получает аналогичную информацию от других маршрутизаторов. Таким образом все маршрутизаторы сети через некоторое время имеют информацию о всех IP-сетях AS (автономной системы). Каждой сети ставится в соответствие метрика (metric). Метрика является параметром, с помощью которого маршрутизаторы оценивают «расстояние» до той или иной сети. Метрика может отражать количество маршрутизаторов, которое должен пересечь пакет, посланный в сеть назначения, базироваться на скорости промежуточных линий связи, и т. д. RIP в качестве метрики использует целое число (integer), находящееся в диапазоне от 1 до 15 (включительно). Реализация протокола (например, операционная система маршрутизатора) должна позволять администратору устанавливать метрики произвольно. Каждая сеть назначения в среде RIP должна иметь IP-адрес, маску и метрику.

Кроме маршрутизаторов, в протоколе RIP могут участвовать также хосты. Они подчиняются тем же правилам, что и маршрутизаторы.

Забегая вперед, скажем, что RIP версии 1 (RIPv1) не способен передавать информацию о маске сети назначения. RIPv2 способен. Соответственно, в среде RIPv1 можно использовать сети только с натуральной адресацией, т. е. нельзя использовать подсети (subneting).

Каждый маршрутизатор, использующий RIP, должен иметь таблицу маршрутизации. Таблица маршрутизации имеет отдельную запись для каждой сети, которой данный маршрутизатор способен передавать данные. Каждая запись содержит (по минимуму) следующую информацию:

- IPv4-адрес сети назначения (адрес назначения).
- Метрика. Метрика является суммой cost'ов (cost - цена) сетей, промежуточных между маршрутизатором и сетью назначения.
- IPv4-адрес маршрутизатора, которому должны быть переданы данные для того, чтобы они достигли сети назначения (так называемый **next hop** – следующий участок). Если сеть назначения является directly connected, то данный параметр не имеет значения.
- Флаг, указывающий, изменялась ли эта запись. Такой флаг может определяться как «route change flag» (флаг изменения маршрута/записи).

- Различного рода таймеры, соответствующие данной записи. Рассматриваются далее.
- Маска подсети.

Записи для directly connected сетей заносятся по мере поднятия интерфейсов, к которым эти сети подключены. Метрики этих сетей устанавливаются в соответствии с параметром cost для этих сетей. **По умолчанию cost = 1.** В том случае, если cost всех сетей в среде RIP равен 1, то метрика будет отражать просто количество сетей от маршрутизатора до сети назначения. Назначение сетям различных cost'ов оправдано в том случае, если эти сети имеют разную скорость передачи, или различные стоимостные характеристики.

Реализации протокола должны позволять администратору сети вносить записи в таблицу маршрутизации вручную. Такая запись называется «static route» (статический маршрут). Остальные записи появляются и обновляются автоматически.

RIP является IGP-протоколом, т. е. действует внутри AS. Если сеть содержит несколько AS, и поддерживает EGP-протокол, позволяющий AS обмениваться маршрутной информацией, соответственно в каждой AS должен быть маршрутизатор, участвующий и в RIP, и в соответствующем EGP-протоколе.

3.6 Формат сообщений

Для передачи сообщений RIP использует протокол **UDP**. Для посылки и приема сообщений в обоих версиях RIP используется **UDP порт 520**. Это означает, что сообщения содержат в качестве порта источника и назначения 520. Сообщения RIP, являющиеся ответом на запрос, в качестве порта назначения имеют порт, с которого был получен запрос. Запросы могут посылаться с порта, отличного от стандартного для RIP, но всегда должны адресоваться на стандартный порт (520).

Ниже показан формат пакета RIP версии 1.

Биты

8	16	24	32
команда	версия	должны быть установлены в 0	
RTE (Запись маршрута - Route entry) 20 байт Пакет RIP может содержать от 1 до 25 RTE			

Формат пакета RIPv1

Поле RTE имеет следующий формат.

Биты

8	16	24	32
address family identifier (AFI)		должны быть установлены в 0	
IPv4-адрес			
должно быть установлено в 0			
должно быть установлено в 0			
метрика			

Формат поля RTE для RIPv1

Каждое сообщение содержит RIP-заголовок. RIP-заголовок содержит идентификаторы команды и версии. Данный раздел рассматривает RIPv1; RIPv2 рассматривается в разделе 4.

Версия

Поле содержит номер версии.

Команда

Поле команды указывает функциональное назначение сообщения. Версии 1 и 2 используют следующие виды команд:

1	запрос (Request)	Запрос, направляемый какой-либо системе для получения полной таблицы маршрутизации или ее части.
2	ответ (Response)	Сообщение, содержащее полную таблицу маршрутизации или ее часть. Данный тип сообщения может быть ответом на запрос, или может посылаться на регулярной основе.

Для обоих типов сообщений в случае использования версии 1 остаток сообщения содержит перечень RTE. Каждый RTE содержит поле AFI, IPv4-адрес и метрику.

AFI

Поле AFI указывает тип используемого адреса. Для RIPv1 поддерживается только AF_INET (2).

Метрика

Поле содержит целое значение от 1 до 15 (включительно). Значение 16 означает, что сеть «недостижима», т. е. пакеты, предназначенные этой сети, переданы быть не могут.

3.7 Адресация

Маршрутизация с использованием distance vector алгоритма может служить для расчета маршрутов как к сетям, так и к отдельным устройствам (хостам). RIP поддерживает обе возможности. «Назначение» (IP-адрес), содержащийся в сообщении, может указывать сеть, хост или специальный код, используемый для указания маршрута «по умолчанию» - default address. Как правило, в большинстве сетей расчет маршрутов к конкретным хостам не требуется. Однако сети, содержащие линии связи точка-точка могут потребовать от маршрутизаторов высчитывать и хранить маршруты к конкретным хостам. Реализация протокола должна поддерживать использование хостов. В том случае, если использование хостов не поддерживается, информация о них просто игнорируется.

В случае использования RIPv1 протокол не в состоянии отличить различные типы адресов – нет маски. Поле IP-адрес может содержать следующие значения:

- Нулевой адрес – **default route**.
  Для default route используется специальный тип адреса – 0.0.0.0. default route используется маршрутизатором в том случае, когда адрес назначения пакета не соответствует ни одному из адресов, содержащихся в таблице маршрутизации. Как правило, default route поднимается на одном из маршрутизаторов вручную. Например это может быть маршрутизатор, через который осуществляется доступ в глобальную сеть. Все остальные маршрутизаторы могут получать информацию о default route с помощью RIP-update'ов. С точки зрения RIP default route ничем не отличается от любого другого сетевого адреса.
- Адреса сетей или хостов. При пересылке информации маршрутизатор будет использовать наиболее точное соответствие между адресом назначения в пакете и полем «адрес» в таблице маршрутизации. Приведем пример. Маршрутизатор должен переслать пакет с адресом назначения 115.168.14.13. Таблица маршрутизации содержит записи с полем адреса, равным 115.168.14.0 и 115.168.0.0. Для передачи пакета будет использоваться запись, содержащая поле адреса, равное 115.168.14.0. Соответственно, если таблица маршрутизации будет содержать адрес хоста 115.168.14.13, то пакет будет отправлен с использованием информации, соответствующей данному хосту.

При получении информации через RIPv1-маршрутизатор может интерпретировать полученные данные в зависимости от того, знает ли он маску подсети, соответствующую полученным сетям. Если знает, то в состоянии правильно интерпретировать полученную информацию.

Например, рассмотрим сеть 128.6, имеющую подсетевую маску 255.255.255.0. Таким образом, адрес сети 128.6.0.0, адрес подсети 128.6.4.0 и 128.6.4.1 адрес хоста. Однако если получивший информацию маршрутизатор не знает масок, оценка полученных данных может быть некорректной. Например, если полученный адрес не содержит нулевых октетов, то невозможно определить, указывает ли он на хост или на какую-либо подсеть. Поскольку подсеть без маски определить невозможно, в таких ситуациях подразумевается, что получен адрес хоста. Для того, чтобы избежать подобных ситуаций, необходимо учесть, что в среде RIPv1 узлы знают маски только тех подсетей, которые непосредственно к ним подключены. Таким образом, не следует рассылать информацию о подсетях за пределы области, в которой маска для этих подсетей известна. RIPv2 разрешает эту проблему, используя в своих пакетах поле маски.

В среде RIP можно использовать «граничные» - «border» маршрутизаторы. Такие маршрутизаторы одной стороной подключены к сетям, использующим подсетевые маски. Внутри области, использующей подсетевые маски, такой маршрутизатор будет рассылать сообщения, содержащие отдельные RTE для каждой подсети. Другой стороной border-маршрутизатор подключен к оставшейся части сети. В эту оставшуюся часть сети border-маршрутизатор будет посылать сообщения, содержащие один общий RTE

для всех подсетей, т. е. будет подразумеваться использование натуральной маски. В результате - border-маршрутизатор будет посылать на разные интерфейсы разные версии update'ов.

3.8 Таймеры

Раздел «Таймеры» описывает события, запуском которых управляют таймеры, а также значения и установку таймеров.

3.8.1 Таймер периодической рассылки.

Первый таймер, который стоит отметить – **30-секундный интервал**, через который RIP посылает свои сообщения типа Response (это же сообщение мы ранее называли update). Такие сообщения содержат полную таблицу маршрутизации (за исключением случаев, описанных в разделе *3.4.4 – Split horizon*). Response может посылаться как на периодической основе, так и в ответ на сообщение типа Request (запрос).

В сети с большим количеством маршрутизаторов или сетей выполнение всеми маршрутизаторами правила 30-секундного интервала между посылками RIP-update'ов может привести к периодическим перегрузкам сети. Эти перегрузки будут вызываться broadcast или multicast – штормами RIP-update'ов. Это объясняется тем, что 30-секундные интервалы маршрутизаторов могут синхронизироваться. Смещение интервалов может происходить, например, из-за перегрузки процессора маршрутизатора и «растягивания» таймера. В целях предотвращения таких ситуаций реализация протокола должна поддерживать следующие характеристики:

- Источник отсчета времени (clocking) для 30-секундного интервала не должен зависеть от загрузки процессора маршрутизатора или от смещения времени, по каким-либо причинам случившегося при срабатывании предыдущего таймера.
- 30-секундный интервал не должен быть точно 30-секундным. Значение таймера каждый раз должно смещаться на значение (+/- от 0 до 5 секунд), что должно препятствовать синхронизации update'ов маршрутизаторов.

3.8.2 Таймауты.

Каждой записи в таблице маршрутизации соответствуют два таймера – «таймаут» (timeout) и «уборка ненужных данных» (garbage-collection).

Если запись в таблице маршрутизации не обновляется в течении timeout, то такая запись объявляется «непригодной» к использованию, но из таблицы маршрутизации не удаляется. Соответственно, соседние маршрутизаторы могут быть уведомлены о изменении ситуации. После того, как истечет и таймер garbage-collection, запись удаляется из таблицы маршрутизации.

Timeout инициализируется/запускается в момент занесения записи в таблицу маршрутизации, и сбрасывается каждый раз, когда маршрутизатор получает update, содержащий информацию о этой записи. Если запись не обновляется **180 секунд** с последней инициализации timeout'а, то маршрутизатор считает, что запись более «негодна». Далее для записи запускается процесс уничтожения, описанный ниже.

Процесс уничтожения записи в таблице маршрутизации.

Записи в таблице маршрутизации может быть уничтожена по двум причинам: по происшествии timeout'а, или метрика для записи была установлена равной 16-ти. В любом случае, маршрутизатор должен совершить следующие действия:

- Таймер **garbage-collection** устанавливается в **120 секунд**.
- Метрика для записи устанавливается равной 16.
- Устанавливается флаг route change (маршрут изменен), что указывает на то, что параметры маршрута были изменены.
- Передается информация процессу вывода (см. ниже) о том, что можно посылать update (напомним – это механизм triggered update).

До тех пор, пока таймер garbage-collection не закончился, удаляемая запись включается во все пакеты типа Response, посылаемые маршрутизатором. После истечения garbage-collection запись удаляется из таблицы маршрутизации.

Если во время работы garbage-collection поступает информация о удаляемой сети, удаляемая запись заменяется новой. garbage-collection сбрасывается.

3.8.3 Таймер triggered update.

Обсуждается в разделе *3.9.1*.

3.9 Процесс ввода (получения информации) – Input Processing.

Данный раздел описывает процесс обработки получаемых RIP-сообщений. Процедура обработки зависит от значения поля **команда** в заголовке пакета RIP.

3.9.1 Сообщения типа Request

Сообщения типа Request используются для запроса на получения полной таблицы маршрутизации или ее части. Как правило Request посылается как broadcast (multicast для RIPv2). В качестве порта-источника сообщения используется стандартный порт UDP для RIP – 520. Таким образом поступают маршрутизаторы, которые только загрузились и хотят узнать о окружающих их сетях. Что нужно сделать в этом случае? «Заорать» во все стороны (со всех интерфейсов): «Люди добрые, дайте, кто чего знает!»

С другой стороны, может возникнуть ситуация, когда необходимо получить таблицу маршрутизации только одного конкретного маршрутизатора. В этом случае Request посылается непосредственно такому маршрутизатору, но в качестве UDP-порта источника используется не-RIP значение (не 520). При получении такого запроса маршрутизатор отвечает непосредственно на адрес и порт запрашивающего.

Запрос обрабатывается последовательно запись за записью. Если запрос не содержит записей – соответственно ответ не формируется и не отсылается.

RIP поддерживает один специализированный тип запроса: запрос содержит только одно поле RTE, в котором поле AFI установлено в 0 и метрика установлена в infinity (16). Данный специализированный тип означает запрос на полную таблицу маршрутизации. В этом случае управление передается процессу вывода (Output processing) с указанием того, что нужно выслать полную таблицу маршрутизации на указанный адрес/порт.

Исключая вышеописанный случай, обработка запроса проста и однотипна. Как говорилось ранее, обработка запроса ведется запись за записью (RTE за RTE). Для каждого RTE проверяется собственная таблица на предмет того, есть ли там соответствующая запись. Если есть, в метрику обрабатываемого RTE помещается метрика из таблицы маршрутизации. Если нет, в метрику обрабатываемого RTE помещается 16 (infinity). После того, как все RTE обработаны, поле команды в заголовке изменяется с Request на Response и пакет отсылается обратно запрашивающему.

Обратим внимание на некоторую разницу при формировании ответов на запросы всей таблицы маршрутизации или только ее части. При запросе полной таблицы маршрутизации срабатывает обычный Output processing (процесс вывода), который включает в себя механизм split horizon. При запросе информации о конкретных маршрутах информация в Response помещается в том виде, в котором она присутствует в таблице маршрутизации, т. е. никакого split horizon. Почему так? Ну… в общем потому, что запрос полной таблицы необходим, как правило, для получения маршрутной информации. Здесь split horizon необходим. Запросы на конкретные сети формируются, как правило, в диагностических целях, поэтому информация в ответе должна быть точной.

3.9.2 Сообщения типа Response

Сообщение Response может быть получено в следующих случаях.

- Ответ на конкретный запрос.
- Регулярный update.
- Triggered update, вызванный изменением таблицы маршрутизации.

Независимо от того, чем вызван Response, механизм его обработки остается одним и тем же.

Поскольку в результате обработки Response'а существует возможность изменения таблицы маршрутизации, сам Response должен быть тщательно проверен на корректность. Если в качестве UDP-порта назначения пакета используется не RIP-порт, сообщение должно игнорироваться. Проверяется IP-адрес источника: если источник пакета находится не на непосредственно подключенной сети, то такой пакет игнорируется. Кроме того, производится проверка на то, является ли адрес источника одним из собственных адресов маршрутизатора – это возможно, если несколько интерфейсов маршрутизатора подключены к одной broadcast-сети.

После проверки Response на корректность RTE пакета обрабатываются запись – за – записью. Каждая RTE проверяется на корректность. В случае некорректности какой-либо записи она должна

игнорироваться. Сообщение о том, что была получена некорректная запись, должно быть внесено в логфайл (log file) маршрутизатора. Базовые проверки должны содержать:

- Корректность адреса назначения – поле IP-адреса. Адрес должен быть unicast'ным, не быть нулевым или 127-ым.
- Корректность метрики. Метрика должна находиться в диапазоне от 1 до 16, включительно.

В том случае, если RTE не прошла какую-либо проверку, она должна игнорироваться, известие об этом должно помещаться в логфайл.

Итак, все проверки для RTE пройдены. Теперь для дальнейшей обработки необходимо добавить к метрике cost той сети, с которой RTE была получена. Если результат получается больше, чем 16, использовать в качестве результата 16 (infinity). Т. е.

метрика = MIN (метрика + cost, infinity)

Далее, проверяем таблицу маршрутизации на предмет наличия записи, адрес которой в точности совпадает с полученным в RTE. Если такового там не найдено, создать новую запись в соответствии с информацией, полученной в RTE (исключая те случаи, когда метрика = 16). Добавление записи в таблицу маршрутизации состоит в следующем:

- Установить поле «Адрес назначения» в адрес, содержащийся в полученном RTE.
- Установить поле «метрика» в полученное в результате расчетов значение.
- Установить поле «next hop» в адрес маршрутизатора, от которого получен update.
- Инициализировать для записи timeout. Если для данного маршрута работает garbage-collection, сбросить его.
- Установить флаг изменения маршрута, или записи – route change.
- Передать информацию в Output processing на trigger update.

Далее. Предполагаем, что мы нашли запись в таблице маршрутизации с адресом назначения, равным адресу назначения в полученном RTE. Если эта запись получена от того же маршрутизатора, что и RTE, и метрики равны, реинициализировать timeout.

Если эта запись получена от того же маршрутизатора, что и RTE, и метрики разные; или полученная для записи метрика меньше, чем содержащаяся в таблице маршрутизации:

- Поместить в запись таблицы маршрутизации новую метрику – а при необходимости и сменить адрес next hop – если он отличается от предыдущего.
- Установить флаг изменения маршрута, или записи – route change и передать информацию в Output processing на trigger update.
- Если новая метрика – 16 (infinity), запустить процесс уничтожения записи, если нет – реинициализировать timeout.

Процесс удаления записи запускается только в том случае, если метрика была изменена и приняла значения 16. Если метрика уже была равна 16, процесс удаления не запускается.

Если получена запись, метрика и адрес которой совпадают с существующим в таблице маршрутизации, в принципе нет необходимости заменять существующую запись новой. Однако если существующая запись в таблице маршрутизации некоторое время не обновлялась (более нескольких периодов посылки update'ов), существует вероятность того, что с ней не все в порядке. В этом случае если получена информация о той же сети и с той же метрикой хорошей практикой будет воспользоваться новой информацией. RFC 2453 рекомендует переключение на новую информацию в том случае, если запись в таблице маршрутизации не обновлялась половину времени от установленного timeout'а.

3.10 Процесс вывода (Output processing)

В данном разделе описывается процесс формирования сообщений Response при выдаче всей таблицы маршрутизации или ее части. Процесс формирования Response и его транслирования в сеть может быть вызван следующими причинами:

- Передачей управления от Input processing в случае получения сообщения Request.
- Срабатыванием 30-секундного таймера регулярной посылки update'ов.
- Механизмом triggered update (т. е. в случае изменения таблицы маршрутизации).

В том случае, если сообщение Response посылается всем соседним маршрутизаторам, оно посылается broadcast'ом (multicast для RIPv2) на всех сетях, которые поддерживают механизмы broadcast'инга, или на дальний конец линии связи в том случае, если тип подключенной сети – точка-точка. Для каждой непосредственно подключенной сети формируется свое Response-сообщение, затем посылается на соответствующий адрес (broadcast/multicast или напрямую на другой конец линка). Однако существуют случаи сетей, не поддерживающих broadcast'инг. В таком случае маршрутизатор должен иметь список маршрутизаторов, которым он должен направлять Response напрямую. Каким образом формируется данный перечень, остается на совести разработчика аппаратуры/программного обеспечения.

3.10.1 Triggered Updates

Механизм Triggered Update требует специального подхода по нескольким причинам.

Во-первых опыт показывает, что в немаленькой сети сообщения, посылка которых вызывается этим механизмом, могут стать причиной перегрузки сети и сокращения полосы пропускания (если будут посылаться слишком часто). Протокол RIP требует наличия механизма, который не позволил бы генерировать update'ы с излишней частотой.

После посылки сообщения, вызванной Triggered Update, маршрутизатор должен ожидать случайное время (от 1 до 5 секунд) до посылки следующего сообщения, чем бы оно не было вызвано. Если же triggered update должен быть послан в тот момент, когда подошло время посылки регулярного update'а, то он не посылается. Посылается регулярный.

Во-вторых, как правило нет необходимости включать в triggered update полную таблицу маршрутизации. Достаточно включить информацию только о той записи, которая изменилась, т. е. только для тех записей, у которых установлен флаг «запись (маршрут) изменена». Это первое. Второе. В triggered update должны включаться все непосредственно подключенные сети. Третье. При генерации triggered update должен использоваться механизм split horizon. Если после применения механизма split horizon оказывается, что метрика маршрута не изменилась (т. е. был 16, и осталась 16), то запись о таком маршруте в triggered update не включается. Если после применения split horizon новых записей для посылки не оказалось, triggered update не посылается. Если triggered update был послан, и в него были включены записи из таблицы маршрутизации, то для таких записей флаг «запись изменена» снимается. Если input processing и output processing наложились по времени, то флаг «запись изменена» не должен изменяться процессом input processing до завершения output processing.

Реально говоря разницы между генерацией обычного – регулярного – update'а и triggered update'а нет, за исключением того, что первый включает всю таблицу маршрутизации, второй – только часть ее. Механизм генерации сообщения, описанный в следующем разделе, применим и к тому, и к другому.

3.10.2 Генерация сообщения Response

Данный раздел описывает механизм генерации сообщений Response. Сообщение Response создается отдельно для каждой непосредственно - подключенной сети.

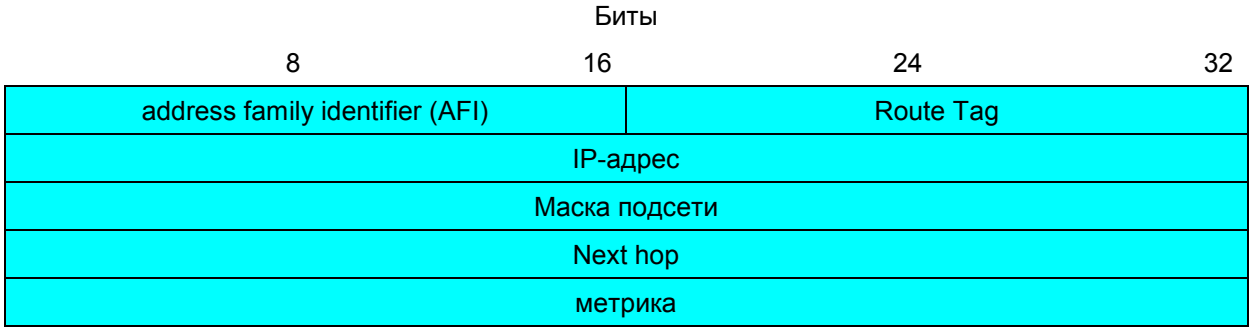
Ниже изложена последовательность действий по созданию Response.

- Установить номер версии. Номер устанавливаемой версии определяется конкретной реализацией протокола и настройками устройства; однако если Response формируется в ответ на конкретный Request, версии Response и Request должны совпадать.
- Установить код команды в «Response».
- Установить байты, которые должны быть равны нулю, в ноль.
- Начать заполнение RTE. Напомним, что в один пакет возможно поместить не более 25 RTE. Если вся посылаемая информация не помещается в один RTE, послать полностью заполненный пакет и затем начать формировать новый, с дополнительной информацией. Количество пакетов с RTE не ограничивается.
- При заполнении RTE обрабатывать каждую запись в таблице маршрутизации. В случае генерации triggered update обрабатывать только те записи, у которых установлен флаг «запись изменена» (route change). В процессе обработки записи реализуется алгоритм split horizon. Если после работы последнего выясняется, что запись не должна помещаться в Response, RTE для нее не формируется. Если запись должна быть включена в Response, ее адрес назначения и метрика включаются в RTE. Записи включаются в Response даже в том случае, если их метрика равна 16.

4 Расширение протокола. RIPv2.

Под расширением протокола RIPv1 понимается RIPv2. Почему «расширение»? Потому что по сравнению с RIPv1 RIPv2 не внес в протокол каких-либо серьезных изменений в механизме или формате сообщения. RIPv2 обеспечивает передачу дополнительной информации, пользуясь пакетами того же формата, что и RIPv1.

Формат заголовка пакета RIP во второй версии не изменился, исключая, естественно, поле «версия». Те поля RTE, которые ранее не использовались, теперь содержат дополнительную информацию. Формат RTE для RIPv2 показан на следующем рисунке:



Формат поля RTE для RIPv2

Поля AFI, IP-адрес и метрика имеют то же значения и формируются тем же образом, что и для RIPv1. Значение остальных полей будет рассмотрено ниже.

4.1 Аутентификация

Для аутентификации используется целый отдельный RTE. Это сделано по следующим причинам:

- Аутентификация требуется для каждого отдельного сообщения.
- В заголовке пакета RIP недостаточно места для реализации нормальной схемы аутентификации.

Для аутентификации может использоваться первый, и только первый RTE пакета. Если используется аутентификация, AFI первого RTE будет равно **0xFFFF**, а оставшаяся часть RTE будет содержать данные аутентификации. Таким образом, на данные остается 24 RTE. Если аутентификация не используется, ни один из RTE не должен содержать AFI, равное 0xFFFF. Начало сообщения RIP, использующего аутентификацию, будет выглядеть следующим образом:

Начальная часть пакета RIPv2 при использовании аутентификации

RFC 2453 специфицирует использование только одной схемы аутентификации – использование простого нешифруемого пароля. Тип аутентификации – 2. 16 байт поля «данные аутентификации» содержат пароль. Пароль начинается с первого байта поля. Выравнивается (т. е. остаток поля заполняется) нулями.

4.2 Route Tag

Route tag (RT) – таг маршрута. Поле RT является неким атрибутом, назначаемым записи в таблице маршрутизации. RT используется для того, чтобы отличить «внутренние» маршруты, т. е. маршруты, выученные через RIP, от маршрутов «внешних», т. е. полученных от других протоколов маршрутизации, таких, как EGP.

Маршрутизаторы, на которых подняты несколько различных протоколов маршрутизации, должны обеспечивать возможность конфигурации RT для маршрутов, которые получены от «внешних» по отношению к RIP'у протоколов маршрутизации.

RFC 2453 допускает иное использование поля RT, что может позволить EGP-протоколам взаимодействовать с RIP в целях передачи, например, транзитной информации через RIP-области.

4.3 Маска подсети

Поле «маска подсети» содержит маску подсети. Если это поле равно нулю, следовательно подсетевая маска отсутствует. Используется натуральная.

Если на каком-либо интерфейсе RIPv2-маршрутизатора существует вероятность того, что update'ы может получать RIPv1-маршрутизатор, то должны выполняться следующие правила (на этом интерфейсе):

1. Информация, являющаяся внутренней для одной сети, не должна посылаться в другую сеть.
2. Информация о подсетях не должна посылаться, поскольку RIPv1-маршрутизатор может принять ее за посылку информации о адресе хоста.
3. Supernet routes (суперсетевые маршруты, т. е. маршруты с маской сети, более короткой, чем натуральная) не должны рассылаться, поскольку они могут быть неправильно интерпретированы RIPv1-маршрутизаторами.

4.4 Next hop

Поле указывает адрес маршрутизатора, которому должны быть посланы данные для того, чтобы они достигли сети назначения, которая указана в поле IP-адрес данного RTE. Для чего это? Например, маршрутизатор рассылает информацию о сети 10.10.10.0, но по каким либо причинам считает, что пакеты для этой сети должны посылаться не через него, а через другой маршрутизатор. В этом случае в поле next hop он указывает адрес этого «другого» маршрутизатора.

Next hop должен содержать адрес, принадлежащий той же сети, на которой получен данный update. Если next hop не находиться на той же сети, его значение должно рассматриваться как равное 0.0.0.0. Пример использования поля next hop приведен в Приложении А.

4.5 Multicasting, или адрес назначения при посылке сообщений RIPv2

В целях уменьшения использования полосы пропускания сетей RIPv2 вместо broadcast'ового адреса использует multicast'овый – **224.0.0.9**. При этом нет необходимости говорить о использовании IGMP, поскольку RIP сообщения не должны пересылаться маршрутизаторами.

В сетях NBMA (Non-broadcast multi-access) для рассылки сообщений может использоваться unicast-адрес. Однако в том случае, если на такой сети маршрутизатором получено RIP-сообщение, адресованное на адрес 224.0.0.9, оно также должно быть обработано.

В целях сохранения совместимости использование multicast'ового адреса должно конфигурироваться. При использовании multicast'а он должен использоваться на всех интерфейсах.

4.6 Запросы

Если RIPv2-маршрутизатор получает RIPv1 Request, он должен формировать и отправить RIPv1 Response. Если же такой маршрутизатор сконфигурирован на отправку только RIPv2-сообщений, он не должен отвечать на такой запрос.

5 Совместимость

Данная секция рассматривает некоторые вопросы, относящиеся к совместимости.

Должны игнорироваться следующие пакеты:

- Содержащие номер версии, равный 0.
- Содержащие ненулевые значения в пакете RIPv1 в тех полях, которые должны быть установлены в 0.

5.1 Compatibility switch

Compatibility switch (переключатель совместимости) необходим по двум причинам.

Первое, существуют разработки RIPv1, которые не полностью соответствуют RFC 1058 (в котором описан RIPv1). Второе, multicast-сообщения RIPv2 будут игнорироваться RIPv1-интерфейсами, что в некоторых случаях нежелательно.

Compatibility switch должен конфигурироваться отдельно для каждого интерфейса и должен поддерживать следующие четыре «положения»:

- RIP-1. Посылаются только RIPv1-сообщения.
- RIP-1-совместимый, при котором формируются сообщения формата RIPv2, но посылаются broadcast'ом.
- RIP-2, в котором формируются сообщения формата RIPv2 и посылаются multicast'ом.
- Положение «none», т. е. «нет». Запрет на посылку любых RIP-сообщений.

RFC 2453 рекомендует по умолчанию устанавливать режим RIP-1 или RIP-2, но не RIP-1-совместимый. Хотя не все производители этого придерживаются. По-моему, они правы – судя по практике. RFC сообщает, что режим RIP-1-совместимый должен использоваться сетевым администратором в том случае, если он понимает, что делает со своими маршрутизаторами.

Устройство, работающее с RIP, также должно поддерживать параметр receive control switch (ключ управления получением) - RCS. RCS управляет тем, сообщения какой версии принимать. Имеет положения RIP-1, RIP-2, оба и ничего. Должен конфигурироваться отдельно для каждого интерфейса. Рекомендуется, чтобы по умолчанию установки переключателей посылки и приема совпадали.

5.2 Аутентификация

При использовании аутентификации выполняются следующие алгоритмы:

- В том случае, если маршрутизатор не использует аутентификацию. Сообщения RIPv1 и не аутентифицированные RIPv2 должны приниматься и обрабатываться; аутентифицированные сообщения RIPv2 должны игнорироваться.
- В том случае, если маршрутизатор использует аутентификацию сообщений RIPv2. Сообщения RIPv1 и сообщения RIPv2, прошедшие аутентификацию, принимаются и обрабатываются. Сообщения RIPv2, не прошедшие аутентификацию, игнорируются. При повышенных требованиях к ограничению доступа в данном режиме сообщения RIPv1 также должны игнорироваться.

Отметим, что RIPv1 воспринимает аутентифицированное сообщение RIPv2 как содержащее неверный первый RTE (неверный AFI), но при этом обрабатывает информацию, содержащуюся в остальных RTE. Т. е. аутентификация не предохраняет маршрутизаторы с RIPv1 от получения аутентифицированной информации RIPv2.

5.3 Увеличение infinity

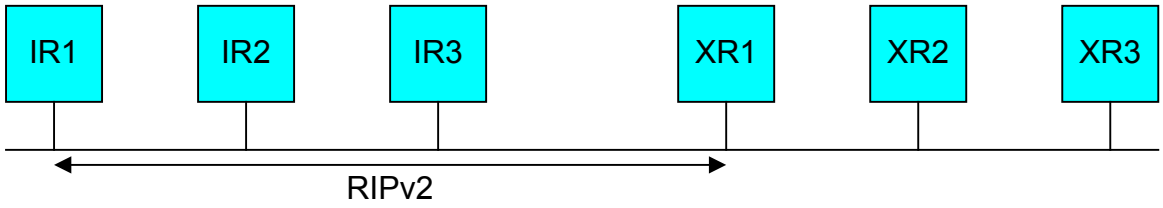
Во второй версии не предусмотрено из соображений совместимости.

5.4 Addressless links

Не поддерживаются ни первой, ни второй версиями RIP.

6 Приложение А. Использование поля next hop

Здесь приведен простой пример того, зачем может быть понадобиться использование поля next hop.



Примем, что IR1, IR2 и IR3 «внутренние» маршрутизаторы, использующие RIP в качестве IGP-протокола. Маршрутизаторы XR1, XR2 и XR3 находятся административно в другой сети и используют другой протокол маршрутизации, например, OSPF. Обмениваясь маршрутной информацией, они знают, что лучший маршрут к сетям N1 и N2 – через XR1, к N3, N4 и N5 – через XR2 и к N6 и N7 – через X3. Если корректно установить поле next hop, то только XR1 должен посылать RIPv2-сообщения, в которых будет указываться, какая из сетей через какой маршрутизатор доступна. Без этого все маршрутизаторы должны были бы посылать информацию о сетях, которые через них доступны.