Санкт-Петербург, ул. Седова, 80, телефон (812) 449-0770, факс (812) 449-0771, E-mail: info@bilim.com

Network Working Group
Request for Comments: 2453
Obsoletes: 1723, 1388
STD: 56
Category: Standards Track

G. Malkin
Bay Networks
November 1998

Протокол RIP версии 2

RIP Version 2

Статус документа

Этот документ содержит спецификации протокола для сообщества Internet и служит запросом для предложений и дальнейшего обсуждения в целях развития стандарта. Информацию о текущем статусе документа можно найти в Internet Official Protocol Standards (STD 1). Документ может распространяться свободно.

Авторские права

Copyright (C) The Internet Society (1998). All Rights Reserved.

Тезисы

Этот документ содержит расширение протокола RIP (Routing Information Protocol), описанного в работе [1], – новый вариант протокола позволяет передавать больше информации в сообщениях RIP и повышает уровень безопасности.
Дополнением к данному документу является определение объектов SNMP MIB для протокола RIP-2 [2]. Кроме того, в работе [3] рассмотрены вопросы криптографической защиты для протокола RIP-2 [3].

Благодарности

Автор благодарит членов рабочей группы IETF RIP за помощь в разработке протокола RIP-2. Значительные фрагменты обсуждения протоколов на базе вектора расстояния и некоторых других вопросов работы протокола RIP были заимствованы из работы [1] (C. Hedrick). Некоторые части окончательного варианта документа были написаны Скоттом Брэднером (Scott Bradner).

Оглавление

1. Обоснование

После появления протоколов OSPF и IS-IS некоторые специалисты полагали, что протокол RIP должен уйти со сцены. Понятно, что новые протоколы маршрутизации IGP существенно превосходят RIP, но и протокол RIP имеет некоторые преимущества. Прежде всего, в небольших сетях RIP обеспечивает существенно меньший расход полосы, а также снижение затрат времени на обслуживание и настройку. Протокол RIP очень прост в реализации, особенно в сравнении с новейшими протоколами IGP.
Кроме того, в существующих сетях протокол RIP распространен во много раз шире, нежели OSPF и IS-IS. Очевидно, что такая ситуация будет сохраняться еще достаточно долго.
Исходя из широкого распространения протокола RIP, представляется разумным расширение возможностей RIP. С учетом затрат на такое изменение и получаемых результатов такая работа представляется очень перспективной.

2. Текущий вариант RIP

Сообщения RIP-1 содержат минимальное количество информации, требуемой маршрутизаторам для направления сообщений через сеть. Кроме того, в сообщениях содержится много пустого места.
В протоколе RIP-1 не рассматриваются автономные системы и взаимодействие IGP/EGP, подсети (subnetting) [11] и аутентификация, поскольку все это было реализовано после создания RIP-1. Отсутствие масок подсетей является особенно серьезной проблемой для маршрутизаторов, поскольку они используют маски подсетей для определения маршрутов. Если маршрут RIP-1 направлен в сеть (все биты адреса, кроме номера сети, имеют значение 0), маска сети эквивалентна маске подсети. Однако если некоторые биты сверх номера сети имеют значение 1, маршрутизатор уже не может определить маску подсети. Более того, маршрутизатор не может отличить маршруты RIP-1 к хостам и подсетям. Некоторые маршрутизаторы в таких случаях для определения типа маршрута просто используют маску интерфейса, через который была получена информация о маршруте.

3. Базовый протокол

3.1 Введение

RIP представляет собой протокол маршрутизации на основе алгоритма Bellman-Ford (или вектора расстояния - distance vector). Этот алгоритм используется для расчета маршрутов в компьютерных сетях с первых дней существования сети ARPANET[1]. Форматы пакетов и описание протокола, приведенные здесь, базируются на программе **routed** из дистрибутива Berkeley Unix.
В международных сетях типа Internet может использоваться множество разных протоколов маршрутизации. Сеть, скорее, следует рассматривать как множество автономных систем (AS), каждая из которых, в общем случае, администрируется как единое целое. В каждой AS будет использоваться своя технология маршрутизации, которая может отличаться в разных AS. Протокол маршрутизации, используемый в AS, называют внутренним протоколом маршрутизации IGP (Interior Gateway Protocol). Отдельный протокол, называемый протоколом внешней маршрутизации EGP (Exterior Gateway Protocol), служит для обмена маршрутной информации между AS. Протокол RIP был разработан в качестве IGP для автономных систем средних размеров. Этот протокол не предназначен для работы в более сложных средах. Информацию о рабочем контексте RIP-1 можно найти в работе Брадена и Постела [6].
RIP использует один из алгоритмов маршрутизации, известный как алгоритм Distance Vector (вектор расстояния или DV). Первое описание этого класса алгоритмов содержится в работе Форда и Фулкерсона [8], поэтому иногда используют термин «алгоритм Форда-Фулкерсона». Используется также термин «алгоритм Беллмана-Форда», отражающий факт использования в алгоритме расчетов на основе уравнения Беллмана [4]. Описание алгоритма в данном документе основано на работе [5]. Настоящий документ содержит спецификацию протокола. Математические основы алгоритмов маршрутизации описаны в работе [1]. Базовый алгоритм, используемый протоколом, применялся еще в 1969 году в сети ARPANET. Однако основные корни данного протокола находятся в сетевых протоколах Xerox. Для обмена маршрутной информацией используются протоколы PUP [7]. Несколько измененная версия этого протокола была адаптирована для XNS (Xerox Network Systems) под названием Routing Information Protocol [9]. Berkeley routed, по сути, представляет собой Routing Information Protocol, в котором адреса XNS заменены более общим форматом, способным работать с IPv4 и другими типами адресов, а обновления маршрутизации передаются каждые 30 секунд. В силу этого сходства термин Routing Information Protocol (или просто RIP) используется для протоколов, применяемых как XNS, так и routed.
Протокол RIP предназначен для использования в IP-сетях. Сеть Internet организована как множество сетей, соединенных между собой через специальные шлюзы (gateway), называемые маршрутизаторами (router). Сети могут строиться на базе соединений точка-точка или использовать более сложные структуры типа Ethernet или token ring. Хосты и маршрутизаторы обмениваются дейтаграммами, содержащими IP-адрес получателя. Маршрутизация представляет собой метод, на основе которого хост или маршрутизатор принимает решение «куда переслать дейтаграмму». Возможна передача дейтаграммы непосредственно в сеть адресата, если этот адресат находится в одной из сетей, непосредственно подключенных к хосту или маршрутизатору. Однако более интересны случаи, когда адресат недоступен напрямую.
В таких случаях хост или маршрутизатор пытается передать дейтаграмму маршрутизатору, который расположен ближе к адресату. Конечная цель протокола маршрутизации очень проста – он обеспечивает и поддерживает информацию, которая требуется для маршрутизации.

3.2 Ограничения протокола

Этот протокол не решает всех возможных задач маршрутизации. Как отмечено выше, основным назначением протокола является использование в качестве IGP для сетей средних размеров. Кроме того, протоколу присущ еще ряд ограничений:

- Использование протокола ограничено сетями, где самый длинный путь (диаметр сети) не превышает 15 хопов (интервал между двумя маршрутизаторами). Разработчики протокола не предполагали его использование в более крупных сетях. Отметим, что это ограничение основано на допущении стоимости каждого маршрута, равной 1 (обычно для RIP используют именно такие значения). Если администратор сети задаст большие значения стоимости, ограничение числа хопов станет еще более жестким.
- Работа протокола зависит от «вычисления бесконечности» (counting to infinity) для разрешения нештатных ситуаций (эта особенность будет рассмотрена ниже). Если в систему входит несколько сотен сетей и в маршрутизации имеются петли, «разрешение» этих петель потребует дополнительного времени (если ограничена частота обновления маршрутов) или полосы (если обновления передаются по факту их обнаружения). Пока такие петли не будут обнаружены и устранены, они будут поглощать значительную часть полосы сети. Будем надеяться, что в реальных сетях это не создаст дополнительных сложностей (за исключением случаев использования низкоскоростных каналов). Более того, эта проблема возникает достаточно редко, поскольку для ее предотвращения приняты специальные меры.
- Протокол использует «фиксированную» метрику для сравнения альтернативных маршрутов. Такое решение не подходит для систем, где выбор маршрута должен основываться на параметрах, измеряемых в реальном масштабе времени (задержка,

[1] Прообраз сети Internet. *Прим. перев.*

надежность доставки, степень загрузки). Простое добавление учета таких параметров приводит к нестабильности, с которой протокол не может работать.

3.3. Структура документа

Основная часть документа состоит из 2 разделов:

- рассмотрение концепций алгоритмов DV в целом;
- описание реального протокола.

Каждый из этих разделов также делится на фрагменты. В параграфе 3.4 делается попытка неформального представления математической базы алгоритма. Отметим, что это представление следует «спиральному» методу – сначала описывается простейший вариант алгоритма, а потом последовательно добавляется рассмотрение новых возможностей. В параграфе 3.5 дано описание реального протокола. За исключением некоторых специфических аспектов, указанных в параграфе 3.4, реализация протокола RIP полностью основана на приведенной в параграфе 3.5 спецификации.

3.4 Алгоритмы Distance Vector

Задачей маршрутизации является поиск пути между отправителем и получателем. В модели Internet это сводится к определению цепочки маршрутизаторов между сетями отправителя и получателя. Пока сообщение или дейтаграмма остается в пределах одной сети или подсети, все вопросы пересылки определяются используемой в сети технологией. Например, сети Ethernet и ARPANET определяют по-своему способ обмена пакетами между отправителем и получателем. IP-маршрутизация начинается с того момента, когда требуется доставка сообщений от отправителя в одной сети к получателю в другой сети. В таких случаях сообщение может проходить через один или несколько маршрутизаторов, соединяющих сети между собой. Если сети отправителя и получателя не являются соседними, сообщение может проходить через несколько промежуточных (intervening) сетей и подключенных к ним маршрутизаторов. После того, как сообщение попадет в маршрутизатор сети получателя, снова используется локальная сетевая технология для доставки сообщения адресату.

В этом параграфе термин «сеть» используется, прежде всего, для обозначения доменов широковещания (например, сеть Ethernet), каналов «точка-точка» или сети ARPANET. Важно понимать, что сеть в таком случае трактуется протоколом IP как единый объект – т.е., решений о пересылке принимать не требуется или они принимаются прозрачным для IP способом, что позволяет протоколу IP трактовать такую сеть как связную систему (например, Ethernet или ARPANET). Отметим, что при обсуждении адресации IP допускается использование термина «сеть» в иных трактовках (в тех случаях, когда слово «сеть» относится к подсетям при рассмотрении вопросов адресации подсетей).

Существует множество вариантов поиска маршрутов между сетями. Весьма полезно классифицировать эти способы по информации, которой должны обмениваться между собой маршрутизаторы для обеспечения возможности писка пути. Алгоритмы DV основаны на обмене очень незначительным объемом информации. Предполагается, что каждый объект (маршрутизатор или хост), участвующий в обмене информацией, хранит сведения обо всех получателях в системе. В общем случае информация обо всех подключенных к одной сети объектах собирается в единую запись (entry), которая описывает маршрут ко всем получателям в данной сети. Такое обобщение становится возможным потому, что для IP маршрутизация внутри сети невидима (отсутствует). Каждая запись в базе данных о маршрутах включает следующий маршрутизатор, которому должны передаваться дейтаграммы, адресованные объекту. Кроме того, запись включает «метрику» - параметр, определяющий общую протяженность маршрута до объекта. Протяженность маршрута является в значительной мере условным понятием и может учитывать такие параметры, как задержка, стоимость услуг по передаче трафика и т. п. Алгоритмы DV (distance vector – вектор расстояния) получили свое название потому, что в них можно рассчитать оптимальный маршрут, обмениваясь только данными о «протяженности». Более того, обмен информацией ведется только между соседними объектами, т. е., объектами, подключенными к одной сети.

Хотя в общем случае маршрутная информация содержит только данные о сетях, в некоторых случаях может потребоваться сохранение данных о маршрутах к отдельным хостам. Протокол RIP не делает формальных различий между сетями и хостами, просто описывая обмен информацией о получателях, которыми могут быть как сети, так и хосты[2]. Математическое представление удобней для случая маршрутов от одного хоста или маршрутизатора к другому. При абстрактном рассмотрении алгоритма лучше представлять маршрутные записи для сетей как сокращение записи для всех объектов, подключенных к сети. Такое сокращение становится возможным лишь потому, что мы предполагаем отсутствие внутренней структуры сети, видимой на уровне IP. Таким образом, в общем случае предполагается одинаковое расстояние для всех объектов данной сети.

Выше было сказано, что каждый объект хранится в маршрутной базе данных как одна запись для каждого возможного адресата в системе. Очевидно, что в реальных системах требуется хранить для каждого получателя следующие сведения:

- **адрес**: в IP-реализациях этого алгоритма хранится IP-адрес хоста или сети;
- **маршрутизатор**: первый маршрутизатор на пути к получателю;
- **интерфейс**: физический интерфейс, используемый для связи с первым маршрутизатором;
- **метрика**: число, показывающее «удаленность» получателя;
- **таймер**: время, прошедшее с момента последнего обновления записи.

В дополнение к перечисленному могут включаться различные флаги и другая внутренняя информация. Эта база данных инициализируется с описанием объектов, непосредственно подключенных к системе. База данных потом обновляется в соответствии с информацией, получаемой в сообщениях от ближайших соседей.

Наиболее важная информация между маршрутизаторами и хостами передается в обновлениях (update message). Каждый объект, участвующий в схеме маршрутизации, передает обновления, описывающие текущее состояние своей маршрутной базы данных. Можно поддерживать оптимальные маршруты для всей сети, используя лишь информацию, получаемую от ближайших соседей. Используемый для этого алгоритм будет описан ниже.

Как был сказано выше, целью маршрутизации является поиск пути передачи дейтаграмм различным адресатам. Алгоритмы DV основаны на хранящихся в каждом маршрутизаторе таблицах, которые указывают лучший маршрут к каждому получателю. Естественно, что для определения лучшего маршрута должны применяться какие-то измеряемые параметры – метрика маршрута.

В простых сетях в качестве метрики естественно использовать счетчик интервалов (хопов) до получателя. В более сложных сетях метрика может учитывать величину задержки, расходы на доставку (оплата трафика) и другие параметры, которые могут влиять на выбор маршрута. Основным требованием является возможность представления метрики всего маршрута как суммы «стоимостей» отдельных интервалов доставки (хопов).

Формально объект j может получить информацию непосредственно от объекта i (минуя процесс передачи через другие маршрутизаторы по пути), т. е. стоимость d(i,j), связанную с интервалом между i и j. В нормальном случае, когда все объекты данной сети рассматриваются как единое целое, значение d(i,j) будет одинаково для всех получателей данной сети и представляет стоимость использования этой сети. Для получения метрики полного маршрута просто складываются стоимости всех интервалов на пути доставки для данного маршрута. В рамках данного документа мы будем предполагать, что стоимость выражается целыми положительными значениями.

[2] Отметим, что в некоторых реализациях могут существовать различия между хостами и сетями (см. параграф 3.2).

Пусть D(i,j) представляет метрику лучшего маршрута от объекта i к объекту j. Эта метрика должна быть определена для каждой пары объектов. d(i,j) представляет стоимость отдельных шагов. Предположим (формально), что d(i,j) представляет стоимость прямой доставки от i к j. Эта стоимость будет бесконечной, если i и j не являются прямыми соседями[3]. Поскольку стоимость является аддитивной, легко показать, что лучшая метрика должна описываться выражением

```
D(i,i) = 0,
D(i,j) = min [d(i,k) + D(k,j)]
```

и лучший маршрут от i следует к соседу k, для которого d(i,k) + D(k,j) имеет минимальное значение[4]. Отметим, что второе уравнение относится к узлу k, являющемуся непосредственным соседом i. Для других случаев значение d(i,k) бесконечно и никогда не может давать минимума.

Из приведенного доказательства очевидно, что можно построить простой алгоритм расчета стоимости для любого маршрута. Объект i запрашивает у своего соседа k его оценку расстояний до адресата и после этого к каждому из полученных значений просто добавляется d(i,k) – стоимость доставки между i и k. После этого i может сравнить значения, полученные от всех соседей и определить самый «дешевый» путь.

В работе [2] доказана сходимость этого алгоритма к корректному значению D(i,j) за конечное время при отсутствии изменений в топологии. Авторы сделали лишь незначительные допущения о том, в каком порядке в котором объекты передают друг другу свою информацию и когда следует проводить новый расчет минимального значения. В общем случае объект не должен прекращать передачу обновлений и расчет метрики, а сети не могут задерживать сообщения навсегда (крах маршрутизации является изменением топологии сети). В доказательстве не делается каких-либо предположений о начальном значении D(i,j), кроме того, что оно предполагается неотрицательным. Использование в доказательстве лишь незначительных допущений весьма важно. Поскольку мы не можем сделать разумных предположений о том, когда следует передавать обновления, опасно запускать алгоритм в асинхронном режиме. В этом случае каждый объект будет рассылать обновления по своим часам. В результате обновления могут отбрасываться сетью, пока не будут отброшены полностью. Поскольку мы не можем сделать предположений о стартовых условиях, алгоритм может принять изменения. Когда система изменяется, алгоритм маршрутизации начинает переход к новому равновесному состоянию, используя старое состояние как стартовое. Важно, чтобы алгоритм сходился за конечное время независимо от стартового состояния. С другой стороны, некоторые изменения могут привести к невозможности сходимости.

Работа данного алгоритма основана на предположении, что каждый объект хранит копии оценок, полученных от всех соседей, и определяет минимальное значение с использованием этих копий. Фактически, в реальных условиях такой необходимости не возникает. Достаточно просто помнить лучшую метрику и идентифицировать передавшего ее соседа. При получении сведений о более привлекательной метрике в запись вносятся соответствующие коррективы. Такой подход позволяет существенно сократить объем вычислений и размеры сохраняемых таблиц.

Существует еще одно отличие между описанным выше алгоритмом и его реализациями в протоколах типа RIP – в теоретическом варианте каждый объект включает запись для самого себя (с нулевой дистанцией). На практике не возникает необходимости в такой записи – стоимости доставки всем объектам сети выражаются единственной записью для данной сети. Рассмотрим ситуацию, когда хост или маршрутизатор G подключен к сети A. C представляет стоимость использования сети A (обычно эта метрика имеет значение 1). Напомним, что мы предполагаем «невидимость» внутренней структуры сети для IP, таким образом, стоимость доставки будет одинаковой для всех объектов внутри сети. В принципе, узел G должен получить сообщение от каждого объекта H сети A, показывающее нулевую стоимость доставки внутри сети. Тогда G будет считать C + 0 расстоянием до H. Вместо того чтобы просматривать множество идентичных сообщений, узел G просто включает в таблицу одну запись для сети A, содержащую метрику C. Эта запись для сети A должна трактоваться как «квинтэссенция» записей для всех узлов сети A. Единственная информация, которая не может быть включена в суммарную запись для сети A, это сведения о самом объекте G, поскольку стоимость доставки из G в G равна 0, а не C. Но, поскольку такие записи с нулевой стоимостью реально не используются, можно обойтись одной записью для всей сети A. В виду того, что записи с нулевой стоимостью реально не используются, хосты, не выполняющие функций маршрутизации, не должны передавать никаких обновлений таблиц. Очевидно, что хосты, не выполняющие функций маршрутизации (т.е., подключенные к единственной сети) не имеют какой-либо полезной информации для передачи другим хостам, за исключением тривиальной записи D(i,i) = 0. Поскольку такие хосты используют единственный сетевой интерфейс, легко видеть, что маршрут в любую другую сеть через такой хост будет приводить на тот же интерфейс (т. е., назад). В результате стоимость такого маршрута не может быть меньше минимальной стоимости C. Поскольку нет нужды в сохранении записей с нулевой стоимостью, хосты, не являющиеся маршрутизаторами, просто не должны участвовать в работе протокола маршрутизации.

Резюмируем сказанное выше о работе узла G. Для каждого получателя в системе G будет сохранять оценку метрики (т. е., общую стоимость доставки) и сведения о соседнем маршрутизаторе, для которого эта метрика получена. Если получателем является сеть, напрямую подключенная к G, узел G просто использует запись, показывающую стоимость использования данной сети и маршрутизации, фактически, не требуется. Легко показать, что после схождения расчетов метрики, записанный таким путем соседний маршрутизатор является первым на пути к адресату[5]. Такая комбинация получателя, метрики и маршрутизатора обычно используется для указания пути к получателю с данной метрикой и через данный маршрутизатор.

Описанный выше алгоритм включает только возможность снижения метрики, поскольку всегда выбирается наименьший из возможных вариантов. Однако в реальной жизни может потребоваться и увеличение значений, если начальная оценка окажется слишком низкой. Следовательно, должен существовать способ увеличения метрики. Для решения этой задачи достаточно использовать следующее правило - если новый набор информации приходит от другого источника (не G), маршрут обновляется только в тех случаях, когда новая метрика лучше, чем D; если же новая информация приходит от G, значение метрики D обязательно обновляется. Легко показать, что при использовании этого правила для увеличения метрики с сохранением получается такой же результат, который будет достигнут при запоминании информации от всех соседей и повторном расчете маршрута с минимальной стоимостью[6].

Выше был описан алгоритм DV (вектор расстояния). Отметим, что это не является описанием самого протокола RIP, поскольку последний содержит ряд дополнений к базовому алгоритму. Каждый объект, принимающий участие в работе протокола[7], должен выполнять перечисленные ниже процедуры.

- Поддержка таблицы с записью для каждого возможного получателя в системе. Запись содержит сведения о «расстоянии» до адресата D и первом маршрутизаторе (G) на пути к адресату. Концептуально должна также включаться запись о «маршруте к себе» с нулевой метрикой, но на практике такие записи не используются.

[3] Отметим, что стоимость d(i, i) имеет нулевое значение (в оригинале ошибочно сказано «бесконечное» - *прим. перев.*), т. е., мы не учитываем здесь прямую передачу узла самому себе.

[4] Это можно строго доказать с использованием метода математической индукции.

[5] Если несколько маршрутов имеют одинаковую стоимость, маршрутизатор будет относиться к одному из таких путей.

[6] Отметим, что рассмотрение построено на предположении о статическом характере системы и не учитывает возможности «падения».

[7] К таким объектам относятся все маршрутизаторы; хосты, на поддерживающие маршрутизации, также могут выполнять эти процедуры

- Периодическая рассылка каждому соседу сообщений об изменении маршрутов. Такое обновление представляет собой набор сообщений, содержащих всю информацию из маршрутной таблицы. Обновление содержит запись для каждого адресата с указанием дистанции до него.
- Когда приходит обновление от соседа G', в него добавляется стоимость для сети, связанной с G' (это должна быть сеть, через которую принято обновление). После этого вычисляется результирующая дистанция D' и проводится сравнение с записями текущей таблицы маршрутизации. Если новая дистанция D' для адресата N меньше существующего значения D, принимается новый маршрут. Т. е. в запись для адресата N будет включать дистанцию D' и маршрутизатор G'. Если маршрутизатор G' является тем, с которого начинается существующий маршрут (т. е., G' = G), новая метрика D' должна включаться в таблицу даже в тех случаях, когда она превышает старую.

3.4.1 Работа с изменениями топологии

Приведенное выше обсуждение справедливо для фиксированной топологии сети. На практике маршрутизаторы и каналы имеют свойство «падать» и восстанавливать свою работоспособность. Для того чтобы учитывать такие особенности, требуется внести в алгоритм некоторые изменения.

Теоретический вариант алгоритма использует информацию от всех ближайших соседей. При изменении топологии меняется и набор соседей. В результате такого изменения при последующем расчете эти изменения будут учтены. Однако, как было отмечено выше, в реальных приложениях используется вариант минимизации с учетом возрастания (incremental version of the minimization), при которой запоминается только лучший маршрут. Если маршрутизатор, включенный в этот маршрут «падает», или разрывается сетевое соединение, расчет может не отразить таких изменений. В реальных приложениях алгоритм зависит от способа, используемого маршрутизатором для передачи уведомлений об изменении топологии. Если маршрутизатор прекращает работать, у него уже нет возможности уведомить своих соседей об изменении топологии.

Для решения этой проблемы протоколы на основе алгоритмов DV должны использовать понятие тайм-аута для маршрутизаторов. Детали реализации зависят от конкретного протокола. Например, в протоколе RIP каждый участвующий маршрутизатор передает обновления всем своим соседям каждые 30 секунд. Предположим, что текущий путь в сеть N использует маршрутизатор G. Если мы не получаем сообщений от G в течение 180, мы можем предположить что маршрутизатор не работает или разорвано соединение с ним. В результате маршрут помечается как некорректный. Когда от другого соседа приходит информация о корректном маршруте в сеть N, этот маршрут указывается взамен некорректного. Отметим, что для объявления маршрута некорректным мы ждали в течение 180 секунд, тогда, как маршрутизаторы рассылают обновления каждые 30 секунд. Это сделано для того, чтобы избавиться от ложных тайм-аутов в результате потери пакетов при констатации тайм-аута в результате отсутствия одного сообщения.

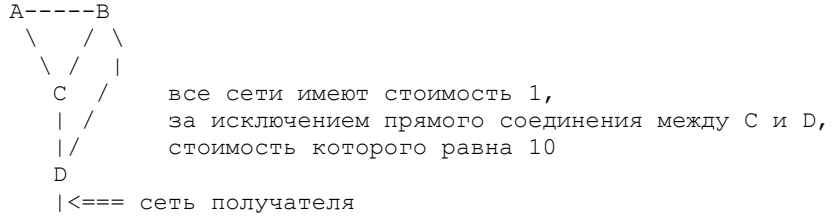
Как было показано выше, полезно иметь способ уведомления соседей об отсутствии маршрута в ту или иную сеть. RIP (как и некоторые другие протоколы этого класса) делает это возможным за счет периодической рассылки обновлений. Для индикации недоступности сети используется специальное значение метрики (16 в текущей реализации протокола RIP). Это значение обычно трактуется как «бесконечность», поскольку оно превышает все остальные значения метрики. Может показаться, что число 16 слишком мало для таких целей. Выбор этого числа в значительной мере определялся простотой работы с ним – число 16 в двоичном выражении удобно использовать как битовый флаг для проверки корректности маршрута.

3.4.2 Предотвращение нестабильности

Представленный выше алгоритм будет всегда обеспечивать хостам и маршрутизаторам возможность расчета корректной таблицы маршрутизации. Однако на практике такой возможности оказывается недостаточно. Приведенные выше варианты алгоритма лишь гарантируют схождение расчетов для таблиц маршрутизации за конечное время. Однако нет никаких гарантий, что это время будет достаточно мало и непонятно, что может произойти с метрикой, когда сеть станет недоступной.

Математическую сторону алгоритма достаточно просто расширить для обработки ставших недоступными маршрутов. Предложенные выше соглашения позволяют это сделать, просто указав достаточно большое значение метрики для недоступных сетей. Это значение должно превышать любую реально используемую метрику. В нашем примере мы выбрали значение 16 в качестве метрики для недоступных сетей. Предположим, что сеть перестала быть доступной. Все соседние с ней маршрутизаторы перестанут видеть сеть и по истечении тайм-аута установят для нее метрику 16. Для нашего анализа мы можем предположить, что все соседние маршрутизаторы получили прямое соединение с исчезнувшей сетью и эти соединения характеризуются метрикой 16. Поскольку с исчезнувшей сетью имеется только одно соединение, все остальные маршрутизаторы в системе будут сходиться к новым маршрутам, проходящим через один из соседних с исчезнувшей сетью маршрутизаторов. Легко видеть, что после схождения расчета все соседние маршрутизаторы получат для исчезнувшей сети метрику не меньше 16. Маршрутизаторы, следующие за ближайшими соседями (1 хоп), получат метрику не менее 17, следующие за ними – не менее 18 и т. д. Поскольку все эти значения превышают максимум, для метрики будет установлено значение 16. Обычно системы будут сходиться на значении 16 в качестве метрики для пропавшей сети.

К несчастью, вопрос о времени схождения расчетов не столь прост. Прежде, чем пойти дальше, рассмотрим пример из работы [2]. Отметим, что приведенный пример не может реально произойти в системах с корректной реализацией RIP – мы просто попытаемся показать необходимость некоторых функций. На приведенном рисунке буквы обозначают маршрутизаторы, а линии - сети.



Каждый маршрутизатор будет иметь таблицу, показывающую маршрут в каждую сеть, однако для упрощения рисунка показаны только пути от каждого маршрутизатора в сеть получателя (в нижней части рисунка).

```
D:  прямое подключение с метрикой 1
B:  маршрут через D, метрика 2
C:  маршрут через B, метрика 3
A:  маршрут через B, метрика 3
```

Предположим, что рвется канал связи между B и D. Маршруты в этом случае должны быть на использование канала между C и D. К несчастью, на такое переключение потребуется время. Изменение маршрутизации начнется после того, как маршрутизатор B обнаружит невозможность использования D. Для простоты будем предполагать, что все маршрутизаторы передают обновления одновременно. Ниже показана метрика для сети получателя с точки зрения каждого маршрутизатора.

```
время ------>

D: dir, 1   dir, 1   dir, 1   dir, 1  ...  dir, 1   dir, 1
B: unreach  C,   4   C,   5   C,   6       C,  11   C,  12
C: B,   3   A,   4   A,   5   A,   6       A,  11   D,  11
A: B,   3   C,   4   C,   5   C,   6       C,  11   C,  12

dir = прямое соединение
unreach = сеть недоступна
```

Здесь возникает проблема – маршрутизатор B может получить информацию о «падении» маршрута с использованием тайм-аута, но достаточно еще долго будет предполагать наличие маршрута в системе. Изначально, маршрутизаторы A и C продолжают думать, что они могут связаться с D по пути через B. Поэтому они будут передавать обновления, содержащие метрику 3. На следующем этапе B будет заявлять, что возможен доступ к D через A или C (естественно, это не так). Маршруты, заявляемые A и C, уже не существуют, но узнать об этом невозможно. И даже после обнаружения недоступности маршрутов через B каждый из маршрутизаторов A и C будет думать о доступности пути через другой маршрутизатор. В конце концов, будут найдены правильные маршруты, но это займет некоторое время. Самая плохая ситуация возникает в тех случаях, когда сеть становится полностью недоступной из какой-то части системы. В таких случаях метрика медленно увеличивается, пока не будет вычислена недоступность. Такие ситуации называются «вычислением бесконечности» (counting to infinity).

Из сказанного можно понять, почему для «бесконечной» метрики выбрано столь малое значение. Если сеть становится полностью недоступной, мы хотим обнаружить недоступность как можно скорей. Значение метрики для недоступной сети должно быть больше метрики любого реального маршрута, но ничто не требует увеличивать его дополнительно. Таким образом, выбор метрики для недоступной сети определяется компромиссом между размером сети и скоростью «вычисления бесконечности». Разработчики протокола RIP предполагали неочевидным практическое использование протокола в сетях, диаметр которых превышает 15.

Для предотвращения проблем, подобных описанной, существует несколько способов. Применительно к RIP - это расщепление горизонта с анонсированием недоступности обратного пути (split horizon with poisoned reverse) и обновления по триггеру (triggered updates).

3.4.3 Split horizon – расщепление горизонта

Отметим, что некоторые из описанных выше проблем вызваны тем, что A и C занимаются обманом друг друга. Каждый из этих маршрутизаторов сообщает о доступности D через другой маршрутизатор. Это можно предотвратить за счет более аккуратного отношения к передаваемой информации. В частности, нет никакой пользы от заявлений о доступности сети получателя для соседей, от которых был получен данный маршрут. Расщепление горизонта (Split horizon) представляет собой схему предотвращения проблем, вызываемых включением маршрута в обновления, передаваемые маршрутизатору, от которого была получена информация об этом маршруте. Простая схема расщепления горизонта (simple split horizon) опускает маршруты, полученные от соседа при передаче обновлений этому соседу. Схема «Split horizon with poisoned reverse» включает такие маршруты в обновления, но устанавливает для них бесконечную метрику (анонсирование недоступности обратного маршрута).

Если маршрутизатор A полагает, что он может связаться с D через узел C, его сообщения маршрутизатору C должны показывать недоступность узла D. Если маршрут через C реален, это говорит о том, что C имеет прямое соединение с D или соединение осуществляется через како-то иной маршрутизатор. Маршрут через C не может возвращаться в A, поскольку это будет порождать петлю. Говоря маршрутизатору C о недоступности D, маршрутизатор A просто пытается предотвратить возможное заблуждение C о доступности маршрута через A. Описанная ситуация достаточно типична для соединений «точка-точка». Рассмотрим теперь ситуацию, когда узлы A и C подключены к широковещательной сети (например, Ethernet) и присутствуют другие маршруты в эту сеть. Если узел A имеет маршрут через C, он должен показывать недоступность D при обмене информацией с любым другим маршрутизатором той же сети. Другие маршрутизаторы в сети могут непосредственно взаимодействовать с C и никогда не будут обращаться к C через узел A. Если лучший маршрут для A реально проходит через C, другим маршрутизаторам той же сети не нужно знать, что узел A имеет доступ к D. Это означает, что обновления, используемые для C, могут передаваться всем остальным маршрутизаторам в той же сети. Таким образом, обновления можно рассылать в широковещательном режиме.

В общем случае использование режима split horizon with poisoned reverse более безопасно по сравнению с простым вариантом split horizon. Если два маршрутизатора имеют указывающие друг на друга маршруты, анонсирование обратных маршрутов с метрикой 16 будет незамедлительно разрывать петли. Если обратные маршруты просто не анонсировать, для ошибочных маршрутов придется ждать тайм-аута. Однако вариант с анонсированием недоступности обратного маршрута (poisoned reverse) имеет недостаток – увеличивается размер маршрутных сообщений. Рассмотрим кампусную магистраль, соединенную со множеством зданий. В каждом здании имеется подключенный к магистрали маршрутизатор, связанный с ЛВС здания. Рассмотрим обновления, которые маршрутизаторы в широковещательном режиме рассылают через магистраль. Все, что требуется знать остальной части сети о каждом маршрутизаторе – это локальная сеть, с которой он соединен. При использовании простого расщепления горизонта в обновлениях будут появляться только те маршруты, которые маршрутизатор передает в магистральную сеть. Если использовать более сложный вариант расщепления горизонта (split horizon with poisoned reverse) маршрутизатор должен упомянуть все полученные из магистрали маршруты с метрикой 16. Если система достаточно велика, размер обновлений существенно возрастает и большинство записей в обновлениях показывают недоступность сетей.

При статическом рассмотрении анонсирование обратных маршрутов с метрикой 16 не дает дополнительной информации. В широковещательных сетях с большим количеством маршрутизаторов дополнительные записи в обновлениях будут занимать существенную часть полосы. Однако такой расход полосы оправдан возможностью учета динамики состояния сети. При изменении топологии упоминание маршрутов, которые не должны проходить через маршрутизатор, вместе с теми, которые проходят через него, может ускорить схождение расчетов метрики. Однако в некоторых случаях администраторы предпочитают более медленное схождение в целях экономии полосы на рассылку обновлений. Таким образом, разработчикам целесообразно давать возможность выбора режима расщепления горизонта (простой или с анонсированием недоступности обратного пути) в качестве опции. Допустима и реализация гибридных схем, когда анонсируется недоступность (метрика 16) лишь для части обратных маршрутов. Примером такой схемы будет использование метрики 16 для обратных маршрутов в течение некоторого времени после изменения маршрутизации и отказ от таких анонсов по истечении времени.

Требования к маршрутизаторам, приведенные в работе [11], указывают, что все реализации RIP должны использовать расщепление горизонта и рекомендуется использовать split horizon with poisoned reverse, с возможностью отказа от этого режима.

3.4.4 Обновления по событию - Triggered updates

Расщепление горизонта с анонсированием недоступности обратного пути (Split horizon with poisoned reverse) будет предотвращать возникновение петель между парой маршрутизаторов. Однако возможны ситуации, когда в процесс «взаимообмана» вовлечены три маршрутизатора. Например, маршрутизатор A может предполагать наличие маршрута через B, B через C, а C через A. Расщепление горизонта не может предотвратить возникновение таких петель. Для разрешения этой проблемы метрика должна

сойтись к бесконечности и соответствующая сеть должна быть объявлена недоступной. Обновления по событию (triggered update) являются попыткой ускорить процесс схождения метрики. Для использования triggered update просто добавляется правило, по которому маршрутизатор, меняя метрику для пути, должен передать уведомление об этом, не дожидаясь времени штатной передачи обновления. Промежуток времени, по истечении которого должно передаваться такое уведомление, зависит от протокола. Некоторые протоколы на базе алгоритмов DV (в частности, RIP) требуют передавать обновление с небольшой задержкой, чтобы избежать значительного роста сетевого трафика. Рассмотрим, как это правило взаимодействует с правилами расчета новой метрики. Предположим, что путь от маршрутизатора в сеть N проходит через маршрутизатор G. Если обновление приходит непосредственно от G, принявший его маршрутизатор должен доверять полученной информации независимо от знака изменения метрики. Если в результате метрика изменяется, принявший обновление маршрутизатор будет передавать в связи с этим событием обновления (triggered updates) всем хостам и маршрутизаторам, непосредственно подключенным к нему. Каждый из получивших обновление узлов может передать его своим соседям. В результате возникает каскад обновлений. Легко увидеть, какие хосту и маршрутизаторы будут вовлечены в этот каскад. Предположим, что маршрутизатор G фиксирует тайм-аут для доступа в сеть N. В результате этого события G будет передавать обновления всем своим соседям. Однако доверятся этому обновлению только те маршрутизаторы, чьи маршруты в сеть N проходят через G. Остальные маршрутизаторы и хосты увидят эту информацию о новом маршруте, который они уже используют, и просто проигнорируют обновление. Соседи, чьи маршруты проходят через G обновят свою метрику и отправят triggered update всем своим соседям. И снова эти обновления будут восприняты только теми маршрутизаторами, для которых путь в сеть N проходит через G. Таким образом, обновления по событию обратно по всем путям, полученным от маршрутизатора G, устанавливая для этого пути бесконечную метрику. Распространение обновлений прекратится как только они попадут в ту часть сети, откуда путь в N идет через другие маршрутизаторы.

Если система сохраняет «спокойное» состояние, пока распространяется каскад обновлений, «вычисление бесконечности» никогда не потребуется. «Плохие» маршруты всегда будут незамедлительно удаляться и петель в маршрутизации возникать не может.

К несчастью, в реальности все обстоит сложнее. Пока передаются обновления по событию, могут быть сгенерированы и переданы штатные (периодические - *прим. перев.*) обновления. Маршрутизаторы, еще не получившие обновления по событию, будут продолжать рассылку информации, содержащей недоступный маршрут. Возможно также после приема обновления по событию получение штатного обновления от маршрутизатора, который еще не получил обновления по событию. В результате этого статус недоступного маршрута может быть некорректно восстановлен (как доступного). Если обновления по событиям рассылаются часто, вероятность такого события становится достаточно большой. Однако возможность «вычисления бесконечности» сохраняется.

В требованиях к маршрутизаторам [11] сказано, что все реализации RIP должны поддерживать обновления по событию для удаленных маршрутов и могут также поддерживать такие обновления для новых или измененных маршрутов. Реализации RIP должны также ограничивать скорость, с которой могут передаваться обновления по событиям (см. параграф 3.10.1).

3.5 Спецификация протокола

Протокол RIP предназначен для обмена информацией, позволяющей рассчитать маршруты через сети на базе IPv4. Предполагается, что любой маршрутизатор, использующий RIP, имеет интерфейсы в одну или несколько [внешних] сетей (в противном случае, это устройство не является маршрутизатором). Будем называть такие сети подключенными напрямую (directly-connected). Протокол основан на доступе к некоторой информации о каждой из таких сетей. Наиболее важной информацией является метрика. RIP-метрика сети выражается целыми числами от 1 до 15, включительно. Ограничения значений метрики не являются частью данной спецификации. Обычно для метрики используется значение 1. Разработчики должны давать администраторам возможность устанавливать метрику для каждой сети. В дополнение к метрике каждая сеть будет иметь связанный с ней адрес [получателя] IPv4 и маску подсети. Эти параметры устанавливаются администратором сети, независимо от данной спецификации.

Предполагается, что любой хост, использующий RIP, имеет интерфейс в одну или несколько сетей, которые называются подключенными напрямую (directly-connected). Протокол опирается на некую информацию о каждой из таких сетей. Наиболее важным параметром является метрика или «стоимость». Метрика сети выражается целым числом от 1 до 15, включительно. Ограничения значений метрики не являются частью данной спецификации. Многие существующие реализации хостов используют метрику 1. В новых разработках администратору должна предоставляться возможность установки метрики для каждой сети. В дополнение к стоимости с каждой сетью связан номер сети IPv4 и маска подсети. Эти значения устанавливаются администратором независимо от данной спецификации.

Отметим, что правила, приведенные в параграфе 3.7, предполагают, что к каждой сети IPv4 применима только одна маска и маски известны только для подключенных напрямую сетей. Могут существовать системы, использующие различные маски подсетей в пределах одной сети. Кроме того, могут встречаться интерфейсы, для которых системе желательно знать маски удаленных сетей. Распространение в масштабах сети маршрутной информации, содержащей разные маски подсетей, допустимо, если все маршрутизаторы в сети поддерживают описанное в данном документе расширение. Однако если некоторые маршрутизаторы не поддерживают это расширение, рассылка информации, содержащей разные маски подсетей, должна быть ограничена во избежание проблем при взаимодействии маршрутизаторов. Правила распространения информации о подсетях приведены в параграфах 3.7 и 4.3.

Предполагается, что каждый маршрутизатор, поддерживающий RIP, имеет таблицу маршрутизации. Эта таблица имеет одну запись для каждого адресата, достижимого через систему, использующую RIP. В каждой записи должны содержаться следующие сведения:

- адрес IPv4 для получателя;
- метрика, которая представляет общую стоимость доставки дейтаграммы от маршрутизатора к адресату (эта метрика представляет собой сумму стоимостей, связанных с каждой сетью на пути к адресату);
- адрес IPv4 для следующего маршрутизатора на пути к адресату (next hop); если адресат находится в подключенной напрямую сети, этот параметр не требуется;
- флаг, показывающий, что информация о маршруте была недавно изменена – его называют флагом изменения маршрута (route change flag);
- различные таймеры, связанные с маршрутом (см. параграф 3.6).

Способ получения информации о подключенных напрямую сетях не оговаривается данной спецификацией. Метрика для непосредственно подключенной сети устанавливается как стоимость этой сети (обычно 1). При использовании единичной стоимости маршрутов метрика RIP сводится к подсчету интервалов (хопов). В более сложных случаях метрика может учитывать параметры предпочтения той или иной сети (например, учитывать доступную полосу или надежность доставки).

Для поддержки предложенный в данной спецификации расширений каждая запись должна содержать также маску подсети. Маски (вместе с адресом IPv4 для получателя) позволяют маршрутизатору идентифицировать различные подсети в одной сети, а также учитывать маски удаленных подсетей.

Разработчики могут также предоставить администратору возможность ввода дополнительных маршрутов. Скорей всего, это будут маршруты к хостам или сетям за пределами видимости системы маршрутизации. Такие маршруты называют статическими (static route). Записи для адресатов сверх тех, которые указаны изначально, добавляются и удаляются с использованием описанных ниже алгоритмов.

Для того, чтобы предоставить полные сведения о маршрутизации, каждый маршрутизатор в автономной системе AS должен участвовать в работе протокола. В тех случаях, когда используется множество протоколов IGP, по крайней мере, один маршрутизатор должен транслировать информацию между разными протоколами.

3.6 Формат сообщения

Работа протокола RIP основана на транспортном протоколе UDP. Каждый маршрутизатор, использующий протокол RIP, включает в себя процесс маршрутизации, который передает и принимает дейтаграммы UDP через порт 520, выделенный для протоколов RIP-1/RIP-2. Весь обмен данными с процессами RIP на других маршрутизаторах осуществляется через порт RIP и через этот же порт передаются все обновления RIP. Незапрошенные анонсы обновления маршрутов передаются с использованием порта RIP у отправителя и получателя. Обновления, передаваемые по запросу, отправляются с использованием того же порта, в который был адресован запрос. Некоторые специфические запросы могут передаваться с использованием портов, отличных от RIP, но они должны быть направлены в порт RIP адресата.

Формат пакетов RIP показан на рисунке[8]:

0								1								2								3				
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	Биты
Команда (1)								Версия (1)								Должно иметь нулевое значение (2)																
Запись RIP (20)																																

Количество записей RIP может меняться от 1 до 25 (включительно). Записи RIP-1 имеют формат:

0								1								2								3				
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	Биты
Идентификатор семейства адресов (2)																Должно иметь нулевое значение (2)																
Адрес IPv4 (4)																																
Должно иметь нулевое значение (4)																																
Должно иметь нулевое значение (4)																																
Метрика (4)																																

Размеры полей приведены в октетах. Если явно не указано иное, значения полей трактуются как беззнаковые целые числа с нормальной (big-endian) записью, когда старший бит указывается слева.

Каждое сообщение содержит заголовок RIP, включающий идентификатор команды и номер версии. В данном параграфе рассматриваются заголовки для первой версии протокола, а заголовки новой версии описаны в параграфе 4. Поле команды показывает цель данного сообщения и может принимать два значения:

1 – запрос (request) на получение таблицы маршрутизации или ее части;

2 – отклик (response), содержащий запрошенную таблицу маршрутизации или ее часть; это сообщение может передано в ответ на запрос или как незапрошенное (unsolicited) обновление таблицы маршрутизации.

Для каждого типа сообщений протокола версии 1 оставшаяся часть дейтаграммы содержит список маршрутных записей (Route Entry или RTE). Каждая RTE-запись в этом списке содержит идентификатор семейства адресов (Address Family Identifier или AFI), адрес получателя (IPv4) и стоимость доставки (метрику).

Поле AFI указывает тип адреса и для RIP-1 может принимать единственное значение AF_INET = 2.

Поле метрики может содержать значения от 1 до 15 (включительно), показывающие «стоимость» доставки информации адресату, или значение 16 (бесконечность), говорящее о недоступности адресата.

3.7 Адресация

Маршрутизация на основе дистанции (стоимости) может использоваться для описания путей к отдельным хостам или сетям. Протокол RIP позволяет использовать любую из этих возможностей. Отправителями (адресами) в запросах и откликах могут быть сети, хосты или специальные коды, используемые для индикации принятых по умолчанию адресов. В общем случае тип реально используемого маршрута будет зависеть от стратегии маршрутизации, принятой в конкретной сети. Во многих сетях устанавливается такой режим, что маршрутная информация для отдельных хостов не требуется. Если каждый узел данной сети или подсети доступен по одному маршруту, нет причин включать информацию о хостах в таблицы маршрутизации. Однако сети на базе каналов «точка-точка» иногда требуют от маршрутизаторов сохранения путей к некоторым узлам. Актуальность этого требования зависит от используемой в системе адресации и маршрутизации. Таким образом, некоторые системы могут не поддерживать маршруты к отдельным хостам. Если маршруты к хостам не поддерживаются при получении откликов такие маршруты будут отбрасываться (см. параграф 3.7.2).

В формате RIP-1 типы адресов не различаются и поле адреса может содержать:

`адрес хоста, номер подсети, номер сети или 0 (принятый по умолчанию маршрут)`

Предполагается, что объекты, использующие RIP-1применяют наиболее конкретную (specific) информацию, которая доступна при маршрутизации дейтаграмм. Т. е., при маршрутизации дейтаграмм адрес получателя должен сначала сравниваться со списком адресов узлов, потом подсетей и, наконец, сетей. Если не найдено соответствия, используется принятый по умолчанию маршрут.

Когда узел оценивает информацию, полученную с помощью RIP-1, интерпретация адреса зависит от наличия информации о применимой маске подсети. Если маска известна, можно определить смысл адреса. Возьмем для примера сеть 128.6 и маску 255.255.255.0. В этом случае 128.6.0.0 будет номером сети, 128.6.4.0 – номером подсети, а 128.6.4.1 – адресом хоста. Однако при отсутствии маски оценка адреса может быть неоднозначной. Если связанная с узлом часть отлична от нуля, не существует способа отличить номер подсети от адреса хоста. Поскольку номер подсети без маски ценности не представляет, в таких ситуациях предполагается, что адрес относится к хосту. Чтобы избавиться от неоднозначностей такого сорта при использовании протокола версии 1, узлы не должны передавать маршруты в подсети тем узлам, про которые нельзя с уверенностью предположить, что им известна маска подсети. Следовательно, при отсутствии специальных мер маршруты в подсети не должны передаваться за пределы сети, частью которой является данная подсеть. Протокол RIP-2 (см. параграф 4) избавляет от неоднозначностей хост/подсеть путем включения маски подсети в маршрутную запись.

Такая «фильтрация подсетей» выполняется маршрутизаторами на «границе» сети, разбитой на подсети. Эти маршрутизаторы подключены к данной сети и каким-то другим сетям. В сети, разбитой на подсети, каждая из подсетей трактуется как отдельная сеть. Обмен маршрутными записи для подсетей осуществляется с помощью RIP. Однако пограничные маршрутизаторы передают только одну запись для все сети во внешние сети. Это означает, что пограничный маршрутизатор передает разную информацию в разные сети. Для соседей, подключенных к сети с подсетями, генерируется список всех подсетей, к которым маршрутизатор

[8] В скобках указано число октетов (байтов) в каждом поле. *Прим. перев.*

подключен напрямую, с использованием номеров подсетей. Соседям, подключенным к другим сетям, передается единственная запись для сети в целом, содержащая метрику, которая связана с этой сетью. Эта метрика обычно меньше метрики для подсетей, к которым маршрутизатор подключен.

Аналогично сказанному, пограничные маршрутизаторы не должны указывать на маршруты к хостам для узлов одной из подключенных напрямую сетей в сообщения для других сетей. Эти маршруты включаются (суммируются) в запись для сети, как целого.

Требования к маршрутизаторам [11] указывают, что все реализации RIP должны поддерживать маршруты к хостам, но если они не делают этого, они должны игнорировать все принимаемые маршруты к хостам.

Специальный адрес 0.0.0.0 используется для обозначения принятого по умолчанию маршрута. Этот маршрут используется в тех случаях, когда неудобно включать каждую подходящую сеть в обновления RIP и когда один или несколько близко расположенных маршрутизаторов в системе подготовлены для обслуживания трафика в сети, которые не указаны явно. Эти маршрутизаторы должны создавать записи RIP для адреса 0.0.0.0, как будто это непосредственно подключенная сеть. Способ, который используется для создания таких записей, могут выбирать разработчики. В общем случае администратору сети предоставляется возможность указать, какой маршрутизатор должен создать запись для 0.0.0.0, однако возможны и другие механизмы. Например, разработчики могут решить, что любой маршрутизатор, использующий BGP, должен декларироваться как используемый по умолчанию. Такой подход может быть полезен, поскольку позволяет администратору выбрать метрику для использования в подобных записях. Если существует несколько принятых по умолчанию маршрутизаторов, можно задать для них предпочтения. Записи для 0.0.0.0 обрабатываются протоколом RIP точно так же, как это происходило бы для реальной сети с таким адресом. Администратор должен принять меры против распространения маршрутов в сеть 0.0.0.0 дальше, чем следует. В общем случае каждая автономная система имеет свой предпочтительный маршрут для использования по умолчанию. Таким образом, маршруты, включающие 0.0.0.0, в общем случае не должны пересекать границ автономной системы. Реализация этого требования не рассматривается в данном документе.

3.8 Таймеры

В этом параграфе рассматриваются все события, активизируемые по таймеру.

Каждые 30 секунд процесс RIP активизируется для передачи незапрашиваемых откликов (Response), содержащих полную таблицу маршрутизации (см. параграф 3.4.3, расщепление горизонта), каждому из соседних маршрутизаторов. При наличии в сети большого числа маршрутизаторов существует тенденция синхронизировать их между собой для одновременной передачи обновлений. Это может произойти в результате воздействия загрузки системы на отсчет 30-секундных интервалов для передачи периодических обновлений. Синхронизация обновлений нежелательна, поскольку она может приводить к многочисленным коллизиям в широковещательных сетях. Следовательно, разработчики должны принять меры предосторожности:

- отсчет 30-секундных интервалов должен вестись по часам, на которые не влияет уровень загрузки системы и время обработки предыдущего обновления;
- 30-секундный интервал изменяется на незначительную величину (+/- 0 - 5 сек) всякий раз при установке таймера[9].

С каждым маршрутом связаны два таймера – timeout (тайм-аут) и garbage-collection (сборка мусора). По истечении тайм-аута маршрут считается недоступным, однако он сохраняется в таблице еще некоторое время, чтобы соседи могли быть уведомлены о недоступности маршрута. При наступлении времени сборки мусора маршрут удаляется из таблицы маршрутизации.

Отсчет тайм-аута начинается при организации маршрута и каждом обновлении, полученном для этого маршрута. Если с момента после последней инициализации тайм-аута прошло 180 секунд, маршрут объявляется недоступным и начинается процесс его удаления, описанный ниже.

Удаление маршрута может произойти в двух случаях – по тайм-ауту и в результате установки для метрики значения 16 на основании обновлений, принятый от текущего маршрутизатора (см. обсуждение вопроса обработки обновлений от других маршрутизаторов в параграфе 3.7.2). В обоих случаях выполняются следующие действия:

- устанавливается таймер сбора мусора (120 секунд);
- для маршрута устанавливается метрика 16 (бесконечность) в результате чего маршрут перестает обслуживаться;
- устанавливается флаг смены маршрута, показывающий изменение записи;
- запускается процесс активизации отклика.

Пока не наступит время сборки мусора (garbage-collection timer), маршрут продолжает включаться во все передаваемые маршрутизатором обновления. По истечении времени (120 сек.) маршрут окончательно удаляется из таблицы.

Если в процессе ожидания времени сбора мусора взамен утраченного организован новый маршрут в ту же сеть, таймер garbage-collection сбрасывается.

Для обновления по событиям используется отдельный таймер, описанный в параграфе 3.9.1.

3.9 Обработка входящей информации

В этом параграфе рассматривается обработка дейтаграмм, принимаемых портом RIP. Процесс обработки зависит от значения поля команды в дейтаграммах. Зависимость процесса обработки от номера версии описана в параграфах 4.6 и 5.1.

3.9.1 Запросы

Запросы (Request) используются для получения от маршрутизатора сообщений, содержащих таблицу маршрутизации или ее часть. Обычно запросы передаются в широковещательном режиме (групповая адресация RIP-2) через порт RIP маршрутизаторами, которые недавно инициализированы и хотят как можно быстрее заполнить свою таблицу маршрутизации. Однако могут возникать ситуации (например, мониторинг маршрутизатора) когда требуется получить таблицу от единственного маршрутизатора. В таких случаях запрос адресуется напрямую нужному маршрутизатору через порт UDP, отличный от порта RIP. При получении такого запроса маршрутизатор отправляет ответ с использованием адреса и номера порта из принятого запроса.

Запросы обрабатываются последовательно – запись за записью. Если запрос не содержит ни одной записи, отклика на такой запрос не передается. Существует один специальный случай для запросов. Если в запросе имеется единственная запись, идентификатор семейства адресов имеет нулевое значение, а метрика бесконечна (16), этот запрос требует передачи всей таблицы маршрутизации. В таких случаях активизируется процесс передачи полной таблицы маршрутизации с использованием адреса и порта из принятого запроса. За исключением этого специального случая обработка запросов достаточно проста. Последовательно просматривается список RTE (маршрутных записей) в запросе и для каждой записи находится получатель в базе данных маршрутизатора. Если искомый маршрут найден, его метрика помещается в соответствующее поле RTE. Если по искомому адресу нет явного маршрута, в поле метрики указывается бесконечное значение (16). После просмотра всех записей значение поля команды изменяется на Response (отклик) и дейтаграмма возвращается отправителю запроса.

Отметим, что при обработке запросов на передачу всей таблицы маршрутизации ситуация несколько меняется. Для таких запросов выполняется обычная обработка, включая Split Horizon (см. параграф 3.4.3, расщепление горизонта). Если же обрабатывается запрос для отдельных записей, они отыскиваются в таблице и возвращаются в исходном состоянии, т. е. Без

[9] Недавние исследования [10] показали допустимость и больших вариаций периода рассылки обновлений.

использования Split Horizon. Причина этого состоит в том, что явно различаются цели запросов. При загрузке маршрутизатора он передает запросы с использованием групповых адресов во все подключенные к маршрутизатору сети для получения полных таблиц маршрутизации. Предполагается, что полученные таблицы будут использоваться для обновления таблицы маршрутизации запрашивающего маршрутизатора. В этом случае требуется выполнение Split Horizon. Для остальных случаев предполагается, что запрос сделан диагностическими программами и не будет использоваться для обновления таблиц. В этом случае запрашивающий маршрутизатор хочет точно знать содержимое таблицы маршрутизации и не хочет, чтобы информация пряталась или изменялась.

3.9.2 Отклики

Получение откликов (Response) может быть вызвано несколькими причинами:

- переданный ранее запрос;
- периодическое обновление (unsolicited response – незапрашиваемый отклик);
- обновление по событию (в результате изменения маршрута).

Обработка откликов не зависит от вызвавшей их причины.

Поскольку обработка отклика может привести к изменению таблицы маршрутизации, для принятых откликов требуется тщательная проверка. Отклики, принятые из порта, отличного от RIP, должны игнорироваться. Адрес отправителя IPv4 должен проверяться на предмет отправки дейтаграммы непосредственным (и допустимым) соседом – адрес отправителя отклика должен принадлежать подключенной напрямую к маршрутизатору сети. Должна также производиться проверка на предмет получения отклика непосредственно от одного из интерфейсов данного маршрутизатора.

Интерфейсы в широковещательные сети могут получать копии своих пакетов, переданных с широковещательным или групповым адресом. Если маршрутизатор будет обрабатывать такие сообщения, это приведет к путанице, поэтому их следует игнорировать.

После проверки дейтаграммы в целом, в отклике последовательно обрабатываются маршрутные записи RTE. Сначала проверяется корректность записи на предмет обнаружения ошибок форматирования (обычно такие ошибки говорят о некорректности настроек и должны привлекать внимание администратора). Например, записи с метрикой, превышающей 16 (бесконечность), должны игнорироваться с записью в журнал. Базовые операции проверки записей перечислены ниже:

- корректность адреса получателя;
- корректность значения метрики (от 1 до 16, включительно)

Если запись не проходи какую-либо из проверок, она должна игнорироваться с переходом к проверке следующей записи. Хорошим тоном является запись в журнал фактов игнорирования записей.

После завершения проверки записи обновляется значение метрики путем добавления стоимости сети, из которой принято обновление. Если результирующая метрика превышает бесконечное значение, устанавливается метрика 16:

```
metric = MIN (metric + cost, infinity)
```

После этого проверяется наличие явного маршрута для данного адресата. Если такой записи нет в таблице, она добавляется, если метрика для нее не имеет бесконечного значения (нет смысла добавлять недоступные маршруты в таблицу). Процесс добавления записи в таблицу включает:

- установку для адреса получателя одноименного значения из RTE;
- установку в поле метрики полученного в результате расчета значения;
- установку в поле next hop (следующий маршрутизатор) адреса маршрутизатора, от которого получено обновление;
- инициализацию отсчета тайм-аута (если уже включен таймер сбора мусора, он останавливается, как описано в параграфе 3.6);
- установку флага изменения маршрута;
- активизацию процесса для инициирования обновления (см. параграф 3.8.1)

Если маршрут уже присутствует в таблице, сравнивается значение поля next hop с адресом маршрутизатора, от которого получена дейтаграмма. Если дейтаграмма пришла от указанного в записи маршрутизатора, отсчет тайм-аута для маршрута начинается заново. После этого проверяется значение метрики. Если дейтаграмма принята от указанного в таблице маршрутизатора и метрика отличается от имеющегося значения, или новая метрика меньше старой (независимо от передавшего ее маршрутизатора), выполняются перечисленные ниже операции:

- в таблицу вносятся изменения (т. е., корректируется метрика и при необходимости изменяется поле next hop);
- Устанавливается флаг изменения маршрута и активизируется процесс инициирования обновления;
- Если новая метрика бесконечна, активизируется процесс удаления записи, описанный выше, в остальных случаях сбрасывается отсчет тайм-аута.

Если новая метрика бесконечна, инициируется процесс удаления маршрута, который больше не используется для пересылки пакетов. Отметим, что процесс удаления инициируется только при первоначальной установке бесконечной метрики. Если метрика уже была бесконечной, новый процесс удаления не запускается.

Если новая метрика совпадает со старой, никаких операций не выполняется, за исключением сброса отсчета тайм-аута, как указано выше. Однако при обработке таких ситуаций существуют определенные тонкости. Обычно не имеет смысла заменять маршрут, если у нового такая же метрика – это будет приводить к неоправданной рассылке обновлений по событию (triggered update). Однако, если для существующего маршрута близок к завершению отсчет тайм-аута, разумно будет заменить этот маршрут равным по стоимости маршрутом, не дожидаясь тайм-аута. Следовательно, при совпадении метрики нового и старого маршрутов целесообразно проверить счетчик тайм-аута. Если прошло более половины заданного времени, имеет смысл переключиться на новый маршрут. Такое решение не является обязательным, но рекомендуется его использовать.

Любая запись, не прошедшая перечисленных проверок, игнорируется (считается, что она не имеет преимуществ перед используемым маршрутом).

3.10 Обработка исходящей информации

В этом параграфе описываются процессы создания откликов, которые содержат всю таблицу маршрутизации или ее часть. Генерация отклика может быть инициирована любым из перечисленных ниже событий:

- получение запроса (Request), как описано в параграфе 3.7.1;
- регулярное обновление (рассылается с использованием групповых или широковещательных адресов каждые 30 секунд);
- обновление по событию (рассылка изменений с использованием группового или широковещательного адреса).

Когда отклик (Response) посылается всем соседям (периодическое обновление или обновление по событию), сообщение адресуется удаленному маршрутизатору для каждого соединения «точка-точка» или передается в широковещательном режиме (групповая адресация в RIP-2) во все подключенные сети, поддерживающие широковещание. Таким образом, один отклик подготавливается для каждой подключенной напрямую сети и передается по соответствующему адресу (прямая адресация или широковещательный/групповой адрес). В большинстве случаев этого достаточно для доставки отклика всем соседним маршрутизаторам. Однако в некоторых ситуациях могут потребоваться дополнительные действия. В числе адресатов могут оказаться сети, не поддерживающие широковещания (например, the ARPANET) или «тупые» (dumb) маршрутизаторы. В таких

случаях может потребоваться указание списка соседних маршрутизаторов и явная адресация дейтаграмм каждому из них. Разработчики должны сами определять необходимость использования такого механизма и способ указания списка соседних маршрутизаторов.

3.10.1 Обновления по событию - Triggered Updates

Обновления по событию (Triggered update) требуют специальной обработки по двум причинам. Во-первых, опыт показывает, что такие обновления могут приводить к перегрузке сетей с низкоскоростными каналами или большим числом маршрутизаторов. Следовательно, разработчики должны принимать меры по ограничению частоты обновлений по событию. После передачи такого обновления требуется установить таймер на случайный промежуток времени в диапазоне от 1 до 5 секунд. Если до истечения этого времени произойдут события, которые должны инициировать дополнительные обновления по событию, передается единственное обновление по истечении заданного времени и таймер снова устанавливается для отсчета случайного времени от 1 до 5 секунд. Обновления по событиям не должны передаваться, если наступило время генерации периодического обновления.

Во-вторых, в обновления по событию не требуется включать таблицу маршрутизации целиком. Следовательно, сообщения, генерируемые как часть обновления по событию, должны включать по крайней мере те записи, для которых установлен флаг изменения маршрута. В обновления могут также включаться другие записи по усмотрению разработчика, однако передача таблиц целиком строго запрещена. При обработке обновлений по событию должно генерироваться сообщение для каждой подключенной напрямую сети. Операции Split Horizon для генерации обновлений по событию используются так же, как для периодических обновлений (см. параграф 3.9). Если после обработки Split Horizon для данной сети обновленный маршрут будет появляться в сети неизмененным (например, с бесконечной метрикой), такой маршрут не должен передаваться. Если в сеть не требуется передавать измененных маршрутов, обновление для этой сети можно опустить. После генерации всех обновлений по событию флаг изменения маршрута следует снять.

Если во время обработки исходящей информации допускается обработка входящих сообщений, должна быть организована соответствующая блокировка. Флаг изменения маршрута не должен меняться в результате обработки входной информации, пока не будет завершена генерация обновлений по событию.

Единственной разницей между обновлениями по событию и прочими обновлениями является возможность исключения неизмененных маршрутов из обновлений по событию. Остальные механизмы, описанные ниже, применимы ко всем типам обновлений.

3.10.2 Генерация откликов

В этом параграфе описана генерация отклика для отдельной сети с прямым подключением:

- установка номера версии 1 или 2 (механизм выбора номера версии зависит от конкретной реализации, однако в откликах на запрос номер версии должен совпадать с номером версии в запросе);
- установка значения Response (отклик) в поле команды;
- установка нулевого значения для следующего поля;
- заполнение записей RTE (число записей в отклике не должно превышать 25 – если реально требуется передать большее число записей, следует генерировать несколько обновлений).

При заполнении полей RTE проверяется каждый маршрут в таблице. Если генерируется обновление по событию, следует включать только записи с установленным флагом изменения маршрута. Если после обработки Split Horizon маршрут не следует включать в обновление, опустите его. Если маршрут включается в обновление, значения метрики и адреса получателя должны быть скопированы в RTE. Маршруты с бесконечной метрикой также должны включаться в дейтаграммы.

4. Расширения протокола

В этом параграфе описываются не изменения протокола RIP, а расширение формата сообщений, которое позволяет маршрутизаторам совместно использовать важную информацию.

В сообщениях RIP-1 и RIP-2 используются заголовки одного формата (см. параграф 3.4). Формат 20-октетных записей RTE для RIP-2 показан ниже:

0										1										2										3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	Биты
Идентификатор семейства адресов (2)										Тег маршрута (2)																						
Адрес IP (4)																																
Маска подсети (4)																																
Следующий маршрутизатор – Next Hop (4)																																
Метрика (4)																																

Поля идентификатора семейства адресов (Address Family Identifier), IP-адреса и метрики описаны в параграфе 3.4. Поле версии должно содержать значение 2 для сообщений RIP, использующих аутентификацию или содержащих информацию в полях, которые не были определены для версии 1.

4.1 Аутентификация

Поскольку функции аутентификации должны выполняться на уровне каждого сообщения, а в заголовке сообщения имеется лишь два свободных октета, тогда как для любой схемы аутентификации требуется больший объем сведений, система аутентификации протокола RIP версии 2 использует для хранения информации записи RIP. Если идентификатор семейства адресов (Address Family Identifier) первого (и только первого) элемента в сообщении имеет значение 0xFFFF, оставшаяся часть сообщения содержит сведения для аутентификации. Это означает, что для аутентификации может использоваться до 24 записей 24 RIP в оставшейся части сообщения. Если аутентификация не используется, поле идентификатора семейства адресов не должно принимать значение 0xFFFF. Сообщения RIP с использованием аутентификации будут иметь следующий формат:

0										1										2										3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	Биты
Команда (1)								Версия (1)								Не используется (2)																
0xFFFF																Тип аутентификации (2)																
Аутентификация (16)																																

В настоящее время поддерживается только один тип аутентификации – простой пароль – и для его обозначения в поле типа устанавливается значение 2. Оставшиеся 16 октетов могут содержать пароль в виде строки текста (plain text password). Если пароль содержит менее 16 октетов, он должен дополняться соответствующим количеством нулей (0x00) справа.

4.2 Тег маршрута

Поле тега маршрута (Route Tag или RT) является атрибутом маршрута, который должен сохраняться и анонсироваться вместе с маршрутом. Поле RT позволяет отличать «внутренние» RIP-маршруты (маршруты для сетей внутри области RIP-маршрутизации) от «внешних», которые импортируются от EGP или других IGP.

Для маршрутизаторов, поддерживающих отличные от RIP протоколы, должна обеспечиваться возможность настройки RT для маршрутов, импортируемых из других источников. Например, маршруты, импортируемые от EGP или BGP, должны обеспечивать возможность установки для них произвольного значения тега маршрута RT или (по крайней мере) номера автономной системы (AS), из которой был получен маршрут.

Возможны и другие варианты использования поля RT, если трактовка этого поля остается одинаковой во всем домене RIP. Это обеспечивает возможность разработки спецификации взаимодействия BGP-RIP, описывающей методы синхронизации маршрутизации в транзитной сети.

4.3 Маска подсети

Поле Subnet Mask содержит маску подсети, которая применяется к IP-адресу для того, чтобы выделить из него номер сети (подсети). Если запись не включает маску подсети, поле должно иметь нулевое значение.

Для интерфейсов, где маршрутизатор RIP-1 может принимать и обрабатывать информацию RIP-2, действуют следующие правила:

1) внутренняя информация одной сети никогда не должна анонсироваться в другие сети;
2) информация о специфической подсети может не анонсироваться, если маршрутизаторы RIP-1 будут воспринимать это как путь к хосту;
3) supernet-маршруты (маршруты, в которых маска подсети содержит меньше битов, нежели «естественная маска сети) не должны анонсироваться, если существует возможность их некорректной интерпретации маршрутизаторами RIP-1.

4.4 Следующий маршрутизатор

Это поле содержит IP-адрес интерфейса соседнего маршрутизатора, по которому пересылаются пакеты для получателей, указанных в маршрутной записи. Значение 0.0.0.0 в этом поле говорит о том, что маршрутизация должна осуществляться через отправителя RIP-анонса. Указанный в поле next hop (следующий маршрутизатор) адрес должен быть доступен непосредственно через логическую сеть в которой выполняется анонсирование.

Поле Next Hop позволяет избавиться от лишней пересылки пакетов в системе. Особенно полезна такая возможность в тех случаях, когда протокол RIP поддерживается не всеми маршрутизаторами в сети. Простой пример приведен в Приложении A. Отметим, что поле Next Hop является «информационным» (advisory), т. е., его можно игнорировать (это может привести к снижению производительности, но маршрутизация будет работать нормально). Если адрес, казанный в поле Next Hop недоступен напрямую, значение поля должно трактоваться как 0.0.0.0.

4.5 Групповая адресация

Для снижения избыточной нагрузки на хосты, которые не слушают сообщений RIP-2, при передаче периодических обновлений может использоваться групповой адрес IP 224.0.0.9. Отметим, что IGMP не требуется, поскольку эти сообщения передаются между маршрутизаторами и не требуют дальнейшей пересылки.

В сетях NBMA должна использоваться конкретная (unicast) адресация. Однако отклики, переданные с использованием группового адреса RIP-2, должны восприниматься.

Для обеспечения обратной совместимости использование групповой адресации должно быть настраиваемым в соответствии с требованиями параграфа 5.1. При включении групповой адресации она должна использоваться для всех поддерживающих такую адресацию интерфейсов.

4.6 Запросы

Если маршрутизатор RIP-2 принимает запрос RIP-1, он должен возвращать отклик в формате RIP-1. Если передача откликов поддерживается только для формата RIP-2, запросы RIP-1 должны игнорироваться.

5. Совместимость

В первой версии протокола [1] предусмотрены некоторые возможности обработки номера версии. Спецификация говорит, что сообщения RIP с номером версии 0 должны отбрасываться, а сообщения версии 1 отбрасываются только при ненулевом значении любого из полей MBZ (Must Be Zero – должно быть нулевым). Для сообщений RIP с номером версии, превышающим не должны отбрасываться лишь потому, что имеют ненулевое значение в поле MBZ. Это означает, что новая версия RIP полностью совместима с существующими реализациями RIP, которые соответствуют этой части спецификации.

5.1 Ключ совместимости (Compatibility Switch)

Включение ключа совместимости обусловлено двумя причинами. Во-первых, существуют реализации протокола RIP-1, в которых поля сообщений отличаются от спецификации [1]. Во-вторых, использование групповой адресации будет предохранять системы RIP-1 от получения обновлений RIP-2 (это может быть весьма желательно для некоторых случаев). Ключ совместимости должен устанавливаться независимо для каждого интерфейса.

Ключ совместимости может принимать 4 значения:

- **RIP-1** – передаются только сообщения RIP-1;
- **RIP-1 compatibility** – сообщения RIP-2 являются широковещательными;
- **RIP-2** – сообщения RIP-2 используют групповую адресацию;
- **None** – запрет передачи сообщений RIP.

Рекомендуется использовать для ключа совместимости значения RIP-1 или RIP-2, но не RIP-1 compatibility, поскольку последнее значение может вызывать проблемы для некоторых вариантов топологии. Значение RIP-1 compatibility следует использовать только в тех случаях, когда все последствия такого применения осознаны сетевым администратором.

Для полноты маршрутизаторы также должны поддерживать ключ управления приемом, который будет определять, какие сообщения воспринимаются – только RIP-1, только RIP-2, оба типа или никакие. Это ключ также должен устанавливаться независимо для каждого интерфейса. Рекомендуется устанавливать используемые по умолчанию значения ключа совместимыми с используемым форматом передачи обновлений.

5.2 Аутентификация

Ниже описан алгоритм, который должен использоваться для аутентификации сообщений RIP. Если в маршрутизаторе отключена аутентификация сообщений RIP-2, должны восприниматься сообщения RIP-1 и сообщения RIP-2 без аутентификации, а сообщения RIP-2 с аутентификацией должны отбрасываться. Если аутентификация сообщений RIP-2 на маршрутизаторе включена, должны восприниматься сообщения RIP-1 и сообщения RIP-2 после проверки аутентификации. Сообщения RIP-2 без

аутентификации или не прошедшие проверку должны отбрасываться. В целях обеспечения максимальной безопасности сообщения RIP-1 следует игнорировать при использовании аутентификации (см. параграф 4.1), поскольку в противном случае сообщения с аутентификацией (пароль) будут передаваться маршрутизаторами RIP-1 без аутентификации.
Поскольку сообщения с аутентификацией помечаются значением идентификатора семейства адресов 0xFFFF, системы RIP-1 будут игнорировать такие записи (в первой версии поддерживается только семейство адресов IP). Следует отметить, однако, что использование аутентификации не запрещает системам RIP-1 видеть сообщения RIP-2. Если нужно, такую недоступность можно организовать путем использования групповой адресации (см. параграфы 4.5 и 5.1).

5.3 Увеличение «бесконечности»

Говоря о совместимости, нельзя не упомянуть, что часто приходится слышать запросы увеличения «бесконечной метрики» (16). Основной причиной отказа от такого увеличения является необходимость обеспечения обратной совместимости. Большее значение бесконечной метрики будет приводить к неоднозначности трактовки поля метрики старыми системами RIP. В лучшем случае маршруты с метрикой больше 16 будут игнорироваться. Были также предложения сделать поле метрики 1-октетным, а три старших октета использовать для других целей – такой вариант будет несовместим с реализациями, трактующими поле метрики как 4-октетное значение.

5.4 Безадресные соединения

Как и RIP-1, протокол RIP-2 не поддерживает безадресные соединения.

6. Взаимодействие между версиями 1 и 2

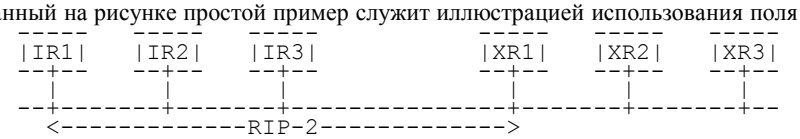
Поскольку в пакетах первой версии протокола не содержится информации о подсетях, семантика, используемая в сетях с маршрутизаторами обеих версий, должна быть ограничена возможностями версии 1. При нарушении этого требования возможно появление в системе «черных дыр» (blackhole route) в виде несуществующих сетей или порождение избыточной маршрутной информации в средах версии 1.
В некоторых реализациях предпринимаются попытки представления групп соседних маршрутов как единого объекта в целях снижения общего числа объектов. Такой подход называют auto-summarization.
При использовании обеих версий протокола в одной сети, во всей такой сети должна использоваться общая маска подсети. Кроме того, для таких сетей должен быть отключен механизм auto-summarization, а разработчики должны обеспечивать возможность такого отключения.

7. Вопросы безопасности

Базовый вариант протокола RIP не обеспечивает безопасности. При переходе к RIP-2 и более современным протоколам маршрутизации обеспечиваются дополнительные возможности аутентификации, описанные в параграфах 4.1 и 5.2. Дополнительные меры безопасности описаны в работе [3].

Приложение А

Показанный на рисунке простой пример служит иллюстрацией использования поля next hop в маршрутной записи.



Предположим, что маршрутизаторы IR1, IR2 и IR3 являются «внутренними» и для них обеспечивается единое администрирование (например, кампусная сеть) с использованием протокола RIP-2 в качестве IGP. Маршрутизаторы XR1, XR2 и XR3, с другой стороны, относятся к сфере другого администратора (например, региональная сеть, в которую входит кампус) и работают с другим протоколом маршрутизации (например, OSPF). XR1, XR2 и XR3 обмениваются маршрутной информацией между собой и знают, что лучшие маршруты в сети N1 и N2 проходят через XR1, в сети N3, N4, N5 – через XR2, а в сети N6 и N7 - через XR3. При корректной установке поля Next Hop (XR2 для сетей N3/N4/N5, XR3 для N6/N7) только маршрутизатору XR1 потребуется обмен сообщениями RIP-2 с IR1/IR2/IR3 для того, чтобы при маршрутизации не использовалось лишнего интервала через XR1. Если Next Hop не используется (например, при работе на базе RIP-1) маршрутизаторам XR2 и XR3 также потребуются сообщения RIP-2 для предотвращения избыточных пересылок.

Литература

[1] Hedrick, C., "Routing Information Protocol", STD 34[10], RFC 1058, Rutgers University, June 1988.
[2] Malkin, G., F. Baker, "RIP Version 2 MIB Extension", RFC 1389[11], January 1993.
[3] Baker, F., R. Atkinson, "RIP-II MD5 Authentication", RFC 2082, January 1997.
[4] Bellman, R. E., "Dynamic Programming", Princeton University Press, Princeton, N.J., 1957.
[5] Bertsekas, D. P., Gallaher, R. G., "Data Networks", Prentice-Hall, Englewood Cliffs, N.J., 1987.
[6] Braden, R., Postel, J., "Requirements for Internet Gateways", STD 4, RFC 1009[12], June 1987.
[7] Boggs, D. R., Shoch, J. F., Taft, E. A., Metcalfe, R. M., "Pup: An Internetwork Architecture", IEEE Transactions on Communications, April 1980.
[8] Ford, L. R. Jr., Fulkerson, D. R., "Flows in Networks", Princeton University Press, Princeton, N.J., 1962.
[9] Xerox Corp., "Internet Transport Protocols", Xerox System Integration Standard XSIS 028112, December 1981.
[10] Floyd, S., V. Jacobson, "The synchronization of Periodic Routing Messages," ACM Sigcom '93 symposium, September 1993.
[11] Baker, F., "Requirements for IP Version 4 Routers." RFC 1812, June 1995.

Адрес автора

Gary Scott Malkin
Bay Networks
8 Federal Street
Billerica, MA 01821

Phone: (978) 916-4237
E-mail: gmalkin@baynetworks.com

[10] В настоящее время этот документ не имеет статуса стандарта (см. STD 1). *Прим. перев.*
[11] В настоящее время этот документ утратил силу и заменен RFC 1724. *Прим. перев.*
[12] В настоящее время этот документ утратил силу и заменен RFC 1812. Перевод имеется на сайте http://www.protocols.ru. *Прим. перев.*

Перевод на русский язык
Николай Малых
nmalykh@bilim.com

Полное заявление авторских прав

Copyright (C) The Internet Society (1998). All Rights Reserved.
This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.
The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.
This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.