Network Working Group
Request for Comments: 1771
Obsoletes: 1654
Category: Standards Track

Y. Rekhter
T.J. Watson Research Center, IBM Corp.
T. Li
Cisco Systems
Editors

March 1995

A Border Gateway Protocol 4 (BGP-4)

Протокол BGP-4

Статус документа

Этот документ[1] содержит проект стандарта протокола Internet и служит приглашением к дискуссии в целях развития протокола. Сведения о текущем состоянии стандартизации вы можете найти в документе Internet Official Protocol Standards (STD 1). Документ может распространяться свободно.

Тезисы

Этот документ вместе с Application of the Border Gateway Protocol in the Internet определяет протокол маршрутизации между автономными системами Internet.

1. Благодарности

Первый вариант этого документа был опубликован в RFC 1267 (октябрь 1991), разработанном Kirk Lougheed (Cisco Systems) и Yakov Rekhter (IBM).

Авторы выражают свою благодарность Guy Almes (ANS), Len Bosack (Cisco Systems) и Jeffrey C. Honig (Cornell University) за их вклад в подготовку предварительных вариантов документа.

Отдельно отметим Bob Braden (ISI) за обзор предварительных вариантов документа и конструктивные замечания.

Благодарим также Bob Hinden (Director for Routing of the Internet Engineering Steering Group) и команду специалистов, подготовивших обзор предыдущей версии документа (BGP-2). Эта команда, в состав которой входят Deborah Estrin, Milo Medin, John Moy, Radia Perlman, Martha Steenstrup, Mike St. Johns и Paul Tsuchiya, показала в работе высокий уровень профессионализма, упорство и такт.

Обновленный вариант документа является результатом работы IETF IDR Working Group. Редакторами документа являются Yakov Rekhter и Tony Li. Некоторые разделы этого документа заимствованы из спецификации IDRP [7], описывающей протокол BGP в рамках OSI. Подготовку этой спецификации обеспечила группа ANSI X3S3.3 под руководством Lyman Chapin (BBN) и Charles Kunzinger (IBM Corp.), который выполнял в группе функции редактора IDRP. Благодарим также Mike Craren (Proteon, Inc.), Dimitry Haskin (Bay Networks, Inc.), John Krawczyk (Bay Networks, Inc.) и Paul Traina (Cisco Systems) за полезные и важные комментарии.

Особо отметим значительный вклад в работу Dennis Ferguson (MCI).

Работа Yakov Rekhter поддерживалась фондом National Science Foundation в рамках гранта NCR-9219216.

2. Введение

BGP (Border Gateway Protocol - протокол граничного шлюза) является протоколом маршрутизации между автономными системами (AS). Протокол разработан с учётом опыта создания протокола EGP (RFC 904 [1]) и применения EGP на магистрали NSFNET (см. RFC 1092 [2] и RFC 1093 [3]).

Основной функцией понимающей BGP системы является обмен информацией о доступности сетей с другими системами BGP. Информация о доступности сетей включает список автономных систем (AS), через которые проходит эта информация. Этих сведений достаточно для построения графа связности AS, из которого могут исключаться маршрутные петли (routing loop), а также приниматься некоторые решения на уровне политики AS.

BGP-4 обеспечивает новые механизмы поддержки бесклассовой междоменной маршрутизации (CIDR). Эти механизмы включают поддержку анонсирования префикса IP и позволяют обойтись без концепции «класса» сетей в рамках BGP. BGP-4 также включает механизм объединения маршрутов, включающий объединение путей AS. Эти изменения в совокупности обеспечивают поддержку схемы «суперсетей» (supernetting scheme) [8, 9].

Для того чтобы охарактеризовать набор решений, которые могут быть реализованы с использованием BGP, следует принять правило, по которому узел BGP может анонсировать узлам-партнёрам (peer) в соседних AS только те маршруты, которые этот узел использует сам. Это правило отражает парадигму поэтапной (hop-by-hop) маршрутизации, используемую в сети Internet для большинства случаев. Отметим, что некоторые правила не могут поддерживаться в рамках парадигмы "hop-by-hop" и, следовательно, требуется использовать другие методы маршрутизации (такие, как source routing). Например, BGP не позволяет AS передавать в соседнюю AS информацию, показывающую маршрут, отличающийся от того, который будет использоваться для трафика, происходящего из соседней AS. С другой стороны, BGP может поддерживать любые правила, соответствующие парадигме поэтапной маршрутизации. Поскольку в современной сети Internet используется только парадигма поэтапной маршрутизации и

[1] В январе 2006 г. выпущен документ RFC 4271, в котором данная спецификация признана утратившей силу. *Прим. перев.*

BGP может поддерживать любые правила, соответствующие этой парадигме, протокол BGP очень распространён для маршрутизации между AS в современной сети Internet.

Более полное обсуждение возможности использования тех или иных правил с протоколом BGP выходит за пределы данного документа и рассматривается в работе [5], посвящённой использованию BGP.

BGP использует транспортные протоколы с гарантированной доставкой. Это позволяет избавиться от необходимости поддержки механизмов фрагментации, передачи, подтверждения доставки и нумерации пакетов. Любая схема аутентификации, обеспечиваемая транспортным протоколом, может использоваться в дополнение к встроенным средствам аутентификации BGP. Используемый в BGP механизм уведомлений об ошибках предполагает, что транспортный протокол поддерживает «элегантное завершение сеансов ("graceful" close), т. е. все остающиеся данные будут доставлены до разрыва соединения.

BGP использует на транспортном уровне протокол TCP [4]. TCP удовлетворяет транспортным требованиям BGP и поддерживается практически всеми современными маршрутизаторами и хостами. При последующем обсуждении слова «соединение транспортного уровня» (transport protocol connection) следует понимать как соединение TCP. Протокол BGP использует для организации соединений порт TCP с номером 179.

В этом документе постоянно будет встречаться термин «автономная система» (AS). По классическому определению автономная система представляет собой множество маршрутизаторов с единым техническим администрированием, использующих один протокол внутренней маршрутизации (IGP) и единую метрику для маршрутизации пакетов внутри AS, а для передачи пакетов в другие автономные системы применяющих протокол внешней маршрутизации (exterior gateway protocol или EGP). Со временем классическое определение было расширено и в современном понимании AS может использовать несколько протоколов внутренней маршрутизации, а в некоторых случаях даже несколько наборов метрик в рамках одной AS. Использование термина AS в таких случаях обусловлено тем, что даже при использовании множества метрик и протоколов IGP администрирование такой AS с точки зрения других автономных систем выглядит как единый план внутренней маршрутизации и показывает согласованную картину доступности адресатов с использованием данной AS.

Планируемое использование BGP в среде Internet включает такие вопросы, как топология, взаимодействие между BGP и протоколами IGP, а также исполнение правил политики маршрутизации, рассмотренных в работе [5], являющейся дополнением к настоящей спецификации. Этот документ является первым из серии планируемых работ по различным аспектам применения протокола BGP. Вы можете отправлять свои комментарии к этому документу по адресу списка рассылок BGP (bgp@ans.net).

3. Обзор

Две системы организуют между собой соединение на транспортном уровне. После этого системы обмениваются сообщениями для согласования и подтверждения параметров соединения. Первоначальный поток данных включает всю таблицу маршрутизации BGP. При изменении таблиц маршрутизации передаются обновления. Протокол BGP не требует периодического обновления всей таблицы маршрутизации BGP. Следовательно, узел BGP должен сохранять текущие версии полных таблиц маршрутов BGP всех соседей одного уровня (peer) в течение всего соединения. Для обеспечения сохранности соединения периодически передаются сообщения KeepAlive. При возникновении ошибок и в иных специальных случаях передаются специальные уведомления. При возникновении ошибок в соединении передаётся уведомление об этом и соединение закрывается.

Хосты, использующие протокол BGP, не обязаны быть маршрутизаторами. Не являющиеся маршрутизаторами хосты могут обмениваться маршрутной информацией с маршрутизаторами с помощью протокола EGP или даже протокола внутренней маршрутизации. В таких случаях эти хосты могут использовать протокол BGP для обмена маршрутными данными с граничным маршрутизатором в другой AS. Реализация и использование этой архитектуры является предметом последующего изучения.

Если отдельная AS имеет множество узлов BGP и обеспечивает транзит для других AS, внутри этой системы должна обеспечиваться согласованная картина маршрутизации, обеспечиваемая протоколом внутренней маршрутизации (IGP). Согласованная картина путей за пределы AS может обеспечиваться за счёт прямых соединений между всеми узлами BGP в масштабе данной AS. Используя общий набор правил, узлы BGP согласуют политику обслуживания точек входа-выхода для конкретных адресатов за пределами данной AS. Эта информация передаётся внутренним маршрутизаторам AS (возможно с использованием протокола внутренней маршрутизации). Должны быть приняты меры по обеспечению обновления транзитной информации на внутренних маршрутизаторах до того, как узлы BGP анонсируют другим AS возможность обеспечения транзитного сервиса.

Соединения между узлами BGP разных AS будем называть внешними каналами, а соединения между узлами BGP в одной AS - внутренними. Узлы одного уровня (peer) из других AS будем называть внешними, а узлы в той же AS - внутренними.

3.1 Маршруты: анонсирование и хранение

При описании этого протокола маршрут определяется как единица информации, содержащая пару «отправитель-получатель» с атрибутами каждого из них.

a)Маршруты анонсируются между парами узлов BGP как сообщения UPDATE - получателем является система, чей IP-адрес указывается в поле NLRI (Network Layer Reachability Information - информация о доступности на сетевом уровне), а путь указывается содержимым полей атрибутов пути в том же сообщении UPDATE.

b)Маршруты сохраняются в базах данных RIB (Routing Information Base - база маршрутной информации) с использованием формата Adj-RIBs-In, Loc-RIB, Adj-RIBs-Out. Маршруты, которые будут анонсироваться другим узлам BGP, должны быть включены в Adj-RIB-Out; маршруты, используемые локальным узлом BGP, должны быть указаны в Loc-RIB, а следующий маршрутизатор (next hop) для каждого из этих маршрутов должен быть представлен в базе рассылки маршрутной информации локального узла BGP; маршруты, полученные от других узлов BGP, включаются в Adj-RIBs-In.

Если узел BGP решает анонсировать маршрут, он может добавить или изменить атрибуты пути для этого маршрута перед анонсированием маршрута другому узлу (peer).

Протокол BGP обеспечивает механизм, с помощью которого узел BGP может информировать узел-партнёр (peer) о том, что анонсированный ранее маршрут больше не может использоваться. Для такого оповещения могут служить три метода:

- в поле WITHDRAWN ROUTES сообщения UPDATE может быть передан префикс IP, указывающий получателя для анонсированного маршрута,- это говорит, что связанный с адресатом маршрут больше не доступен;
- может быть анонсирован новый маршрут с тем же значением NLRI в качестве замены прежнего маршрута;
- соединение между двумя узлами BGP может быть закрыто - это ведёт к полному удалению всех маршрутов, которые эта пара узлов BGP анонсировала друг другу.

3.2 Базы данных о маршрутах

База маршрутной информации (RIB) узла BGP состоит из трёх частей:

a) **Adj-RIBs-In**: маршрутная информация, полученная во входящих сообщениях UPDATE. Эти данные представляют маршруты, которые могут использоваться в качестве входных данных для принятия решения (Decision Process).

b) **Loc-RIB**: локальные маршрутные данные, которые узел BGP выбирает на основе своих локальных правил из данных Adj-RIBs-In.

c) **Adj-RIBs-Out**: данные, которые локальный узел BGP выбрал для анонсирования peer-узлам. Маршрутные данные из Adj-RIBs-Out будут передаваться локальным узлом BGP в сообщениях UPDATE.

Adj-RIBs-In содержит необработанные маршрутные данные, которые получены локальным узлом BGP от узлов-партнёров; Loc-RIB содержит маршруты, выбранные с помощью Decision Process локального узла BGP; Adj-RIBs-Out включает маршрут, которые анонсируются в другие системы с помощью сообщений UPDATE.

Хотя концептуальная модель различает Adj-RIBs-In, Loc-RIB и Adj-RIBs-Out, совершенно не обязательно хранить три отдельных копии маршрутной информации. Выбор числа копий для конкретной реализации (например, 3 копии или 1 копия с указателями на каждую часть) не рассматривается в данной спецификации.

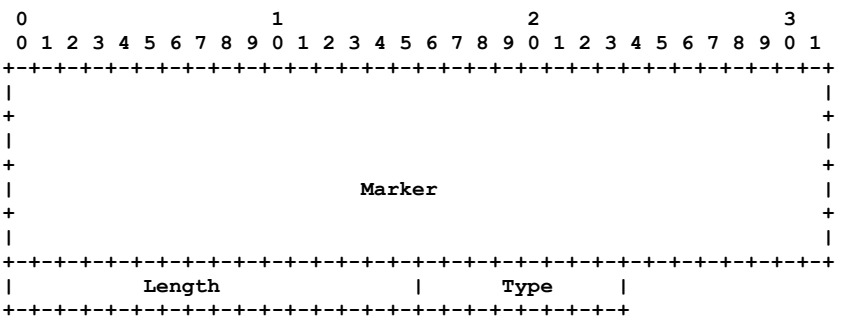
4. Формат сообщений

В этом разделе описаны форматы сообщений, используемых протоколом BGP.

Сообщения передаются с использованием транспортных соединений с гарантированной доставкой. Обработка сообщения происходит после того, как оно будет полностью принято. Максимальный размер сообщения составляет 4096 октетов. От всех реализаций требуется поддержка указанного значения для максимального размера сообщений. Минимальное сообщение, которое может быть передано, содержит только заголовок BGP (19 октетов).

4.1 Формат заголовка сообщений

Каждое сообщение имеет заголовок фиксированного размера. В зависимости от типа сообщения после заголовка может следовать то или иное количество полей данных. Схема размещения полей показана ниже:



Marker - маркер

Это 16-октетное поле содержит значение, которое может быть предсказано получателем сообщения. Если Type = OPEN или сообщение OPEN не содержит Authentication Information (как дополнительного параметра - Optional Parameter), все биты поля Marker должны иметь значение 1. В остальных случаях значение маркера может быт предсказано путём расчётов, являющихся частью используемого механизма аутентификации (часть Authentication Information). Поле Marker может использоваться для обнаружения потери синхронизации между парой узлов BGP и аутентификации входящих сообщений BGP.

Length - длина

Это 2-октетное поле содержит беззнаковое целое число, указывающее общий размер сообщения (с учётом заголовка) в октетах. Эта информация позволяет находить следующее сообщение (поле Marker) в транспортном потоке. Значение поля Length должно лежать в диапазоне от 19 до 4096 (в зависимости от типа сообщения возможны дополнительные ограничения). Заполнение сообщений пустыми полями не допускается, поэтому поле длины должно иметь минимальное значение, достаточное для передачи сообщения.

Type - тип

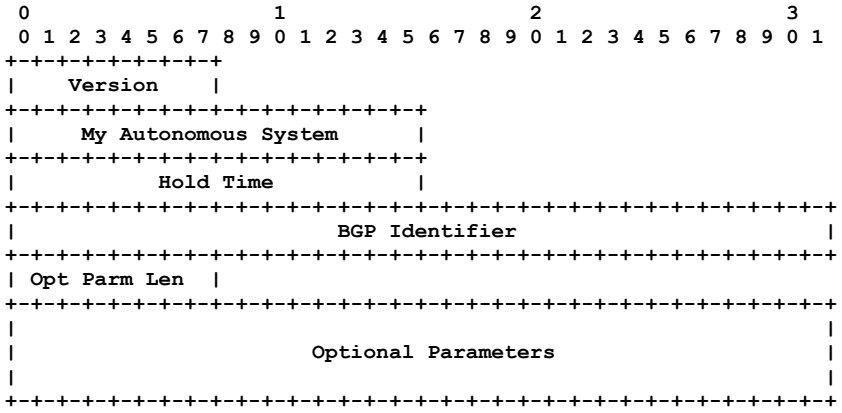
Это 1-октетное поле содержит беззнаковое целое число, определяющее тип сообщения:

1 - OPEN (соединение)
2 - UPDATE (обновление)
3 - NOTIFICATION (уведомление)
4 - KEEPALIVE (подтверждение)

4.2 Формат сообщений OPEN

После организации соединения на транспортном уровне, первым сообщение от каждой из сторон имеет тип OPEN. После восприятия сообщения OPEN узел возвращает сообщение KEEPALIVE, которое подтверждает приём сообщения OPEN. После подтверждения OPEN может происходить обмен сообщениями UPDATE, KEEPALIVE, и NOTIFICATION.

В дополнение к стандартному заголовку BGP сообщение OPEN содержит следующие поля:



Version - версия

Однооктетное беззнаковое целое, указывающее номер версии протокола BGP (для данной версии - 4).

My Autonomous System - моя AS

2-октетное беззнаковое целое, указывающее номер автономной системы отправителя.

Hold Time - время удержания

2-октетное беззнаковое целое, определяющее время (в секундах), которое отправитель предлагает в качестве значения таймера удержания (Hold Timer). Получив сообщение OPEN, узел BGP **должен** выбрать значение Hold Timer (меньшее из настроенного и полученного в сообщении OPEN значений Hold Time). Если значение Hold Time не равно 0, оно **должно** быть не меньше 3 секунд. Реализации протокола могут отвергать соединения на основе значения Hold Time. Выбранное значение показывает максимальный интервал (в секундах) между передачей последовательных сообщений KEEPALIVE и/или UPDATE.

BGP Identifier - идентификатор BGP

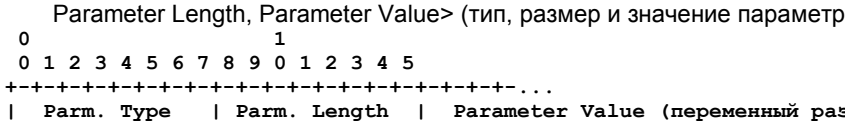
4-октетное беззнаковое целое, указывающее идентификатор BGP для отправителя. Узел BGP устанавливает в качестве значения идентификатора BGP IP-адрес, присвоенный этому узлу BGP. Значение BGP Identifier определяется при загрузке и совпадает для всех локальных интерфейсов и всех узлов BGP одного ранга.

Optional Parameters Length - размер дополнительных параметров

Это 1-октетное поле показывает размер поля дополнительных параметров (Optional Parameters) в октетах. Нулевое значение поля говорит об отсутствии дополнительных параметров.

Optional Parameters - дополнительные параметры

Это поле может содержать список дополнительных параметров, представленный в формате <Parameter Type, Parameter Length, Parameter Value> (тип, размер и значение параметра).

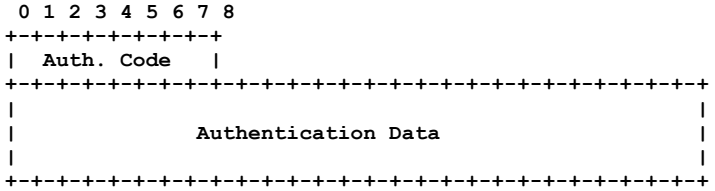


Поле Parameter Type занимает один октет и однозначно идентифицирует отдельно взятый параметр. Поле Parameter Length имеет размер 1 октет и определяет число октетов в поле Parameter Value. Поле Parameter Value не имеет фиксированного размера и интерпретируется в зависимости от значения поля Parameter Type.

В данной спецификации определяются следующие типы дополнительных параметров:

а) **Authentication Information** (данные аутентификации, тип 1):

Этот параметр может использоваться для подтверждения полномочий партнера BGP (peer). Поле Parameter Value содержит 1-октетный код аутентификации, за которым следует поле данных переменной длины.



Authentication Code (код аутентификации)

Это 1-октетное поле содержит беззнаковое целое число, которое указывает используемый механизм аутентификации. При использовании протокола BGP с аутентификацией в спецификацию должны включаться три вещи:

- значение кода (Authentication Code), показывающее используемый механизм
- форма или значение поля Authentication Data
- алгоритм расчёта значений полей Marker.

Отметим, что при организации соединений на транспортном уровне может использоваться отдельный механизм аутентификации.

Authentication Data (данные аутентификации)

Форм и значение этого поля переменной длины зависят от кода аутентификации (Authentication Code).

Минимальный размер сообщения OPEN составляет 29 октетов с учётом заголовка.

4.3 Формат сообщений UPDATE

Сообщения UPDATE используются для передачи маршрутной информации между узлами BGP. Данные из пакетов UPDATE могут использоваться для построения графа, описывающего связи между различными AS. Применение обсуждаемых здесь правил позволяет избавиться от петель и некоторых других аномалий в маршрутизации между AS.

Сообщение UPDATE служит для анонсирования одного возможного маршрута к узлу-партнёру или для отзыва группы анонсированных ранее маршрутов (см. 3.1). Сообщение UPDATE может одновременно анонсировать доступный маршрут и отзывать группу недоступных более маршрутов. Сообщения UPDATE всегда включают заголовок BGP фиксированного размера и могут содержать дополнительные поля, показанные ниже:

Unfeasible Routes Length (2 октета)
Withdrawn Routes (переменный размер)
Total Path Attribute Length (2 октета)
Path Attributes (переменный размер)
Network Layer Reachability Information (переменный размер)

Unfeasible Routes Length (размер недоступных маршрутов)

Это 2-октетное беззнаковое целое число указывает общий размер поля Withdrawn Routes в октетах. Значение этого поля должно позволять определение размера поля Network Layer Reachability Information в соответствии с приведённым ниже описанием.

Нулевое значение говорит об отсутствии отзываемых маршрутов и поля Withdrawn Routes в сообщении UPDATE.

Withdrawn Routes (отзываемые маршруты)

Это поле переменной длины содержит список префиксов IP-адресов, маршруты к которым исключаются из обслуживания. Каждый префикс IP-адреса представляется в 2-компонентном формате <length, prefix>, показанном ниже:

Length (1 октет)
Prefix (переменный размер)

Поля имеют следующие значения:

a) ***Length*** (длина):
показывает размер префикса IP-адреса в битах; нулевое значение поля длины указывает префикс, соответствующий всем адресам IP (сам префикс содержит 0 октетов).

b) ***Prefix*** (префикс):
содержит префикс адреса IP, за которым следует несколько битов, используемых для выравнивания по границе октета (отметим, что значение битов заполнения не играет роли).

Total Path Attribute Length (общий размер атрибутов пути)

Это 2-октетное беззнаковое целое задаёт общую длину поля атрибутов пути (Path Attributes) в октетах. Значение этого поля должно позволять определение длины поля Network Layer Reachability в соответствии с описанной ниже процедурой.

Значение 0 говорит об отсутствии поля Network Layer Reachability Information в данном сообщении UPDATE.

Path Attributes (атрибуты пути)***:***

Последовательность атрибутов пути (переменной длины) присутствует в каждом сообщении UPDATE. Каждый атрибут представляет собой триплет формата <attribute type, attribute length, attribute value> размер которого может меняться.

2-октетное поле Attribute Type содержит октет флагов (Attribute Flags) за которым следует октет типа (Attribute Type Code).

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Attr. Flags  |Attr. Type Code|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Старший бит (0) октета Attribute Flags является битом Optional и определяет тип атрибута - дополнительный (1) или хорошо известный (0).

Второй бит (1) поля Attribute Flags является битом Transitive, определяющим назначение дополнительного атрибута - переходный (1) или непереходный (0). Для хорошо известных атрибутов бит Transitive должен всегда иметь значение 1 (дополнительное рассмотрение переходных атрибутов приводится в главе 5).

Третий бит (2) поля Attribute Flags является флагом Partial. Этот бит показывает полноту информации, содержащейся в дополнительном переходном атрибуте - информация полная (0) или неполная (1). Для хорошо известных атрибутов бит Partial должен иметь значение 0.

Четвёртый флаг (бит 3) задаёт бит Extended Length, определяющий размер поля Attribute Length - 1 октет (0) или 2 октета (1). Флаг Extended Length может использоваться только в тех случаях, когда размер поля атрибутов превышает 255 октетов.

Четыре младших бита поля Attribute Flags не используются и должны иметь нулевые значения (на приёмной стороне эти биты игнорируются).

Октет Attribute Type Code содержит код типа атрибута. Определённые в настоящее время типы рассматриваются в главе 5.

Если бит Extended Length поля Attribute Flags имеет значение 0, третий октет поля Path Attribute содержит размер данных атрибута в октетах.

Если бит Attribute Flags поля флагов имеет значение 1, третий и четвёртый октеты содержат размер данных атрибута в октетах.

Оставшиеся октеты поля Path Attribute представляют значение атрибута и интерпретируются в соответствии со значениями полей Attribute Flags и Attribute Type Code:

a) **ORIGIN** (тип 1):
Атрибут ORIGIN относится к числу хорошо известных и является обязательным - он определяет источник информации о пути. Октет данных атрибута может принимать следующие значения:
0 IGP - информация NLRI является внутренней для AS, из которой пришло сообщение
1 EGP - информация NLRI получена с помощью EGP
INCOMPLETE - данные NLRI получены каким-либо иным способом.
Использование этого атрибута описано в параграфе 5.1.1

b) **AS_PATH** (тип 2):

AS_PATH является хорошо известным обязательным атрибутом, который содержит последовательность сегментов пути AS. Каждый сегмент пути AS представляется тройкой значений <path segment type, path segment length, path segment value>.

1-октетное поле типа сегмента может содержать следующие значения:

1 AS_SET: неупорядоченный набор AS, через которые проходит маршрут из сообщения UPDATE

2 AS_SEQUENCE: упорядоченный набор AS, через которые проходит маршрут из сообщения UPDATE.

Поле path segment length размером 1 октет указывает число AS в сегменте пути.

Значение path segment value содержит один или несколько номеров AS, каждый из которых представлен 2-октетным полем. Использование этого атрибута описано в параграфе 5.1.2.

c) **NEXT_HOP** (тип 3):
Этот хорошо известный обязательный атрибут указывает IP-адрес граничного маршрутизатора, который следует использовать как следующий интервал (next hop) для адресата, указанного в поле Network Layer Reachability сообщения UPDATE. Использование этого атрибута описано в параграфе 5.1.3.

d) **MULTI_EXIT_DISC** (тип 4):
Этот необязательный непереходный атрибут представляет собой 4-октетное неотрицательное целое число. Значение атрибута может использоваться в процессе принятия узлом BGP решения по вопросу избавления от множественных точек выхода в соседнюю автономную систему. Использование этого атрибута описано в параграфе 5.1.4.

e) **LOCAL_PREF** (тип 5):
LOCAL_PREF является хорошо известным необязательным атрибутом, содержащим 4-октетное неотрицательное целое число. Значение атрибута используется узлом BGP для информирования других узлов BGP в своей автономной системе об уровне предпочтения для анонсируемого маршрута. Использование этого атрибута описано в параграфе 5.1.5.

f) **ATOMIC_AGGREGATE** (тип 6)
ATOMIC_AGGREGATE является хорошо известным необязательным атрибутом нулевой длины. Этот атрибут используется узлом BGP для информирования других узлов BGP о том, локальная система выбрала менее специфичный маршрут без выбора более специфичного маршрута, который включён в неё. Использование этого атрибута описано в параграфе 5.1.6.

g) **AGGREGATOR** (тип 7)
AGGREGATOR представляет собой дополнительный атрибут длиной 6 октетов. Этот атрибут содержит номер последней AS, которая формирует агрегированный маршрут (2 октета), за ним следует IP-адрес узла BGP, который формирует агрегированный маршрут (4 октета). Использование этого атрибута описано в параграфе 5.1.7

Network Layer Reachability Information (информация о доступности на сетевом уровне):

Это поле переменной длины содержит список префиксов IP-адресов. Размер поля NLRI в октетах не задаётся в явном виде, но может быть рассчитан как:

```
UPDATE message Length - 23 - Total Path Attributes Length - Unfeasible Routes Length
```

Значение UPDATE message Length берётся из заголовка BGP, значения полей Total Path Attribute Length и Unfeasible Routes Length определяются из сообщения UPDATE, а значение 23 является суммарной длиной заголовка BGP и полей Total Path Attribute Length, Unfeasible Routes Length.

Информация о доступности представляется в виде одной или нескольких последовательностей <length, prefix>, описанных ниже:

Length (1 октет)
Prefix (переменный размер)

a) ***Length*** (размер):
Это поле задаёт число битов префикса IP-адреса. Нулевое значение задаёт префикс, соответствующий любому адресу IP (сам префикс в этом случае содержит 0 октетов).

b) ***Prefix*** (префикс):
Это поле содержит префикс адреса IP, за которым могут следовать биты заполнения для выравнивания по границе октета. Значение битов заполнения не играет роли.

Минимальный размер сообщения UPDATE составляет 23 октета - 19 октетов занимает стандартный заголовок, 2 октета - поле Unfeasible Routes Length, 2 октета - поле Total Path Attribute Length (Unfeasible Routes Length = 0 и Total Path Attribute Length = 0).

Сообщение UPDATE может анонсировать по крайней мере один маршрут, который может быть описан несколькими атрибутами пути. Все атрибуты пути, содержащиеся в данном сообщении UPDATE, применимы к адресатам, указанным в поле Network Layer Reachability Information сообщения UPDATE.

Сообщение UPDATE может включать список маршрутов, обслуживание которых прекращено. Каждый из таких маршрутов указывается префиксом адресата, который однозначно идентифицирует маршрут в контексте узла BGP - соединение узла BGP, для которого это маршрут был анонсирован ранее.

Сообщение UPDATE может анонсировать только отзыв маршрутов. В этом случае сообщение не содержит атрибутов пути и данных NLRI. И наоборот, сообщение может анонсировать только доступный маршрут и не включать поля WITHDRAWN ROUTES.

4.4 Формат сообщения KEEPALIVE

BGP не использует на транспортном уровне каких-либо механизмов keep-alive для проверки доступности других узлов (peer). Вместо этого используются сообщения KEEPALIVE, которыми партнёры обмениваются достаточно часто, чтобы между двумя сообщениями не истекло время, заданное таймером удержания (Hold Timer). Разумным значением максимального интервала между передачей двух последовательных сообщений KEEPALIVE является треть интервала, заданного значением Hold Time. **Недопустимо** передавать сообщения KEEPALIVE чаще одного раза в секунду. Разработчики **могут** установить интервал между передачей сообщений KEEPALIVE как функцию значения Hold Time.

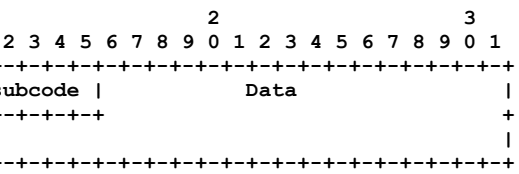
Если Hold Time = 0, периодическая передача сообщений KEEPALIVE **недопустима**.

Сообщение KEEPALIVE состоит только из заголовка, следовательно, размер такого сообщения равен 19 октетам.

4.5 Формат сообщения NOTIFICATION

Сообщения NOTIFICATION передаются в случаях обнаружения ошибок. Соединение BGP незамедлительно закрывается после передачи такого сообщения.

В дополнение к стандартному заголовку BGP сообщение NOTIFICATION содержит следующие поля:



Error Code (код ошибки)

Это 1-октетное поле содержит беззнаковое целое число, указывающее тип сообщения NOTIFICATION:

Код	Название	Дополнительная информация
1	Message Header Error (ошибка в заголовке сообщения)	6.1 Обработка ошибок в заголовке.
2	OPEN Message Error (ошибка сообщения OPEN)	6.2 Обработка ошибок в сообщениях OPEN.
3	UPDATE Message Error (ошибка сообщения UPDATE)	6.3 Обработка ошибок в сообщениях UPDATE.
4	Hold Timer Expired (закончилось время удержания)	6.5 Обработка ошибок Hold Timer Expired.
5	Finite State Machine Error (ошибка конечного автомата)	6.6 Обработка ошибок конечного автомата.
6	Cease (разрыв соединения)	6.7 Разрыв соединения.

Error subcode (субкод ошибки)

Это 1-октетное поле содержит беззнаковое целое число, которое предоставляет дополнительные сведения о природе произошедшей ошибки. Каждое значение Error Code может быть связано с одним или несколькими значениями Error Subcode.

Если не определено подходящего субкода, устанавливается Error Subcode = 0 (неуказанная ошибка).

Субкоды ошибок Message Header Error:

1 - Connection Not Synchronized - соединение не синхронизировано.
2 - Bad Message Length - некорректный размер сообщения.
3 - Bad Message Type - некорректный тип сообщения.

Субкоды ошибок OPEN Message Error:

1 - Unsupported Version Number - неподдерживаемый номер версии.
2 - Bad Peer AS - некорректная AS партнера.
3 - Bad BGP Identifier - некорректный идентификатор BGP.
4 - Unsupported Optional Parameter - неподдерживаемый дополнительный параметр.
5 - Authentication Failure - отказ при аутентификации.
6 - Unacceptable Hold Time - недопустимое время удержания.

Субкоды ошибок UPDATE Message Error:

1 - Malformed Attribute List - некорректный список атрибутов.
2 - Unrecognized Well-known Attribute - нераспознанный атрибут из числа хорошо известных.
3 - Missing Well-known Attribute - отсутствует хорошо известный атрибут.
4 - Attribute Flags Error - ошибка во флагах атрибута.
5 - Attribute Length Error - некорректный размер атрибута.
6 - Invalid ORIGIN Attribute - некорректный атрибут ORIGIN
7 - AS Routing Loop - петля в маршрутизации AS.
8 - Invalid NEXT_HOP Attribute - некорректный атрибут NEXT_HOP.
9 - Optional Attribute Error- ошибка в дополнительном атрибуте.
10 - Invalid Network Field - некорректное поле сети.
11 - Malformed AS_PATH - некорректный формат AS_PATH.

Данные

Это поле переменной длины служит для диагностики причин генерации сообщений NOTIFICATION. Содержимое поля данных зависит от значений полей Error Code и Error Subcode. Дополнительная информация приведена в главе 6. Обработка ошибок BGP..

Отметим, что размер поля данных может быть определён по приведённой ниже формуле:

```
Message Length = 21 + Data Length
```

Минимальный размер сообщений NOTIFICATION составляет 21 октет (с учётом стандартного заголовка).

5. Атрибуты пути

В этой главе рассматриваются атрибуты пути, используемые в сообщениях UPDATE. Атрибуты делятся на 4 категории:

1. Well-known mandatory - общеизвестные, обязательные.
2. Well-known discretionary - общеизвестные, необязательные.
3. Optional transitive - дополнительные, переходные.
4. Optional non-transitive - дополнительные, непереходные.

Хорошо известные атрибуты должны распознаваться всеми реализациями BGP. Некоторые из этих атрибутов являются обязательными и должны присутствовать в каждом сообщении UPDATE. Остальные атрибуты являются необязательными и могут отсутствовать в некоторых сообщениях UPDATE. Все хорошо известные атрибуты должны (после соответствующей обработки, если это нужно) передаваться другим узлам BGP.

Кроме хорошо известных атрибутов каждый путь может содержать один или несколько дополнительных атрибутов. Поддержка дополнительных атрибутов не является обязательной для каждой реализации BGP. Обработка нераспознанных дополнительных атрибутов определяется установкой бита Transitive в октете флагов атрибута. Пути с нераспознанными переходными дополнительными атрибутами должны приниматься. Если путь с нераспознанными дополнительными переходными атрибутами принят и передаётся другим узлам BGP, нераспознанные атрибуты этого пути должны передаваться другим узлам BGP с установленным (1) битом Partial в поле Attribute Flags. Если путь с

распознанным переходным атрибутом воспринят и передаётся другим узлам BGP, а бит Partial октета Attribute Flags имеет значение 1, установленное какой-либо из предыдущих AS, данная автономная система не должна сбрасывать этот бит в 0. Нераспознанные дополнительные непереходные атрибуты следует просто игнорировать, не передавая их другим узлам BGP.

Новые дополнительные переходные атрибуты могут добавляться в путь исходным отправителем (originator) или любой AS в пути. Если эти атрибуты не добавляются исходным отправителем, устанавливается значение Partial = 1 в октете Attribute Flags. Правила присоединения новых непереходных дополнительных атрибутов будут зависеть от природы конкретного атрибута. Предполагается, что документация к каждому новому дополнительному непереходному атрибуту будет включать такие правила (описание атрибута MULTI_EXIT_DISC может служить примером). Все дополнительные атрибуты (переходные и непереходные) могут обновляться (если это допустимо) автономными системами в пути.

Отправителю сообщения UPDATE следует размещать атрибуты пути в сообщениях UPDATE в порядке возрастания типа атрибутов. Получатель сообщения UPDATE должен быть готов к обработке атрибутов пути из сообщения с изменением их порядка.

Один атрибут не может появляться более одного раза в поле Path Attributes конкретного сообщения UPDATE.

5.1 Использование атрибута пути

Ниже описано использование каждого атрибута пути BGP.

5.1.1 ORIGIN

Атрибут ORIGIN является хорошо известным и обязательным. Этот атрибут должен генерироваться автономной системой, которая является исходным отправителем маршрутной информации. Этот атрибут должен включаться в сообщения UPDATE всех узлов BGP, которые выбрали распространение этой информации другим узлам BGP.

5.1.2 AS_PATH

AS_PATH относится к хорошо известным обязательным атрибутам и служит для идентификации автономных систем, через которые передаётся информация в данном сообщении UPDATE. Компонентами списка являются поля AS_SET или AS_SEQUENCE.

Когда узел BGP распространяет маршрут, который был получен в сообщении UPDATE от другого узла BGP, ему следует изменить атрибут AS_PATH с учётом размещения узла BGP, которому передаётся маршрут:

a) Когда данный узел BGP анонсирует маршрут другому узлу BGP, расположенному в той же автономной системе, анонсирующему узлу не следует изменять связанный с этим маршрутом атрибут AS_PATH.

b) Когда данный узел BGP анонсирует маршрут узлу BGP, расположенному в соседней автономной системе, анонсирующему узлу следует изменить связанный с этим маршрутом атрибут AS_PATH как показано ниже:

   1) если первый сегмент AS_PATH имеет тип AS_SEQUENCE, локальной системе следует поместить свой номер AS как последний элемент списка (в крайнюю левую позицию).

   2) если первый сегмент AS_PATH имеет тип AS_SET, локальной системе следует поместить новый сегмент типа AS_SEQUENCE в путь AS_PATH, включив свой номер AS в этот сегмент.

Когда узел BGP является источником маршрута:

a) исходному узлу следует включить свой номер AS в атрибут AS_PATH всех сообщений UPDATE, передаваемых узлам BGP в соседних автономных системах (в этом случае номер AS исходного отправителя будет единственным элементом атрибута AS_PATH).

b) исходному узлу следует включить пустой атрибут AS_PATH во все сообщения UPDATE, передаваемые узлам BGP в своей автономной системе (пустой атрибут AS_PATH содержит нулевое значение в поле размера).

5.1.3 NEXT_HOP

Атрибут пути NEXT_HOP определяет IP-адрес граничного маршрутизатора, который следует использовать в качестве следующего пункта (next hop) на пути к адресату, указанному в сообщении UPDATE. Если граничный маршрутизатор находится в той же AS - это внутренний граничный маршрутизатор, в остальных случаях - внешний. Узел BGP может анонсировать любой внутренний граничный маршрутизатор как next hop, если интерфейс, связанный с IP-адресом этого граничного маршрутизатора (указывается в атрибуте NEXT_HOP), входит в подсеть, разделяемую локальным и удаленным узлами BGP. Узел BGP может анонсировать любой внешний граничный маршрутизатор как next hop, если IP-адрес этого маршрутизатора, полученный от партнера BGP, и интерфейс, связанный с IP-адресом данного граничного маршрутизатора (указан в атрибуте NEXT_HOP), разделяют общую подсеть с локальным и удаленным узлами BGP. Узел BGP должен обеспечивать возможность запрета анонсирования внешних граничных маршрутизаторов.

Для узла BGP недопустимо анонсирование адреса партнера этому же партнёру в сообщении NEXT_HOP для маршрутов, по отношению к которым данный узел является исходным. Для узла BGP недопустимо устанавливать маршруты с собой в качестве next hop.

Когда узел BGP анонсирует маршрут другому узлу BGP, расположенному в той же автономной системе, анонсирующий узел не должен изменять атрибут NEXT_HOP, связанный с этим маршрутом. Когда узел BGP получает маршрут через внутреннее соединение, он может пересылать пакеты по адресу NEXT_HOP, если содержащийся в этом атрибуте адрес относится к той же подсети, в которой находятся локальный и удалённый узлы BGP.

5.1.4 MULTI_EXIT_DISC

Атрибут MULTI_EXIT_DISC может использоваться на внешних (между AS) соединениях для избавления от многочисленных точек выхода или входа в одну соседнюю AS. Значение атрибута MULTI_EXIT_DISC является четырехоктетным беззнаковым целым числом, которое называют метрикой. При прочих равных из множества точек

входа/выхода следует выбирать ту, которая имеет наименьшее значение метрики. Если сообщение получено через внешний канал, атрибут MULTI_EXIT_DISC может распространяться через внутренние соединения другим узлам BGP в той же AS. Атрибут MULTI_EXIT_DISC недопустимо распространять узлам BGP в соседних AS.

5.1.5 LOCAL_PREF
Атрибут LOCAL_PREF относится к числу хорошо известных необязательных. Это атрибут следует включать во все сообщения UPDATE, которые данный узел BGP передаёт другим узлам BGP, расположенным в той же автономной системе. Узлу BGP следует рассчитать уровень предпочтения для каждого внешнего маршрута и указывать этот уровень при анонсировании маршрута внутренним партнёрам. Предпочтение должно отдаваться маршрутам с более высоким уровнем. Узлу BGP следует использовать уровень предпочтения, определённый с помощью LOCAL_PREF, в процессе принятия решений о маршрутизации (см. параграф 9.1.1 Фаза 1 - расчёт уровня предпочтения).

Узлам BGP не следует включать этот атрибут в сообщения UPDATE, которые этот узел передаёт узлам BGP в соседних автономных системах. Если атрибут содержится в сообщении UPDATE, полученном от узла BGP, который находится в другой по отношению к принимающему узлу автономной системе, принимающему узлу следует игнорировать этот атрибут.

5.1.6 ATOMIC_AGGREGATE
Хорошо известный атрибут ATOMIC_AGGREGATE относится к числу необязательных. Если узел BGP при наличии набора перекрывающихся маршрутов от одного из его партнёров (см. параграф. 9.1.4 Перекрывающиеся маршруты) выбирает менее специфический маршрут (без выбора более специфического), тогда локальной системе следует присоединить атрибут ATOMIC_AGGREGATE к маршруту, распространяемому этой системой другим узлам BGP (если этот атрибут уже не был установлен в принятом менее специфичном маршруте). Узлу BGP, получившему маршрут с атрибутом ATOMIC_AGGREGATE, не следует удалять этот атрибут при распространении маршрута другим узлам. Узлу BGP, получившему маршрут с атрибутом ATOMIC_AGGREGATE, не следует делать каких-либо NLRI этого маршрута более специфическими (см. определение в параграфе 9.1.4 Перекрывающиеся маршруты) при анонсировании этого маршрута другим узлам BGP. Узел BGP, получивший маршрут с атрибутом ATOMIC_AGGREGATE, должен знать реальный путь к адресату, как указано в поле NLRI маршрута, хотя при отсутствии петель возможна передача через AS, которые не указаны в атрибуте AS_PATH.

5.1.7 AGGREGATOR
Атрибут AGGREGATOR является дополнительным переходным и может включаться в обновления, которые формируются при агрегировании (см. параграф 9.2.4.2 Агрегирование маршрутной информации). Узел BGP, объединяющий маршруты, может добавлять атрибут AGGREGATOR, содержащий номер AS и IP-адрес этого узла.

6. Обработка ошибок BGP.
В этой главе описаны операции обработки ошибок, которые обнаруживаются при обслуживании сообщений BGP.

При обнаружении любого из описанных здесь условий передаётся сообщение NOTIFICATION с соответствующими значениями полей Error Code, Error Subcode, Data и соединение BGP разрывается. Если субкод ошибки не задан, следует устанавливать Error Subcode = 0.

Фраза "соединение BGP разрывается" означает, что закрывается соединение транспортного уровня и освобождаются все ресурсы, выделенные для соединения BGP. Записи таблицы маршрутизации, связанные с удаленным партнёром, помечаются как некорректные, а информация об этом передаётся другим партнёрам. BGP до того, как маршруты будут удалены из системы.

Если явно не указано иное, поле Data (данные) сообщений NOTIFICATION, передаваемых для индикации ошибки, остаётся пустым.

6.1 Обработка ошибок в заголовке.
При обнаружении какой-либо ошибки в процессе обработки заголовка передаётся сообщение NOTIFICATION с Error Code = Message Header Error. Поле Error Subcode показывает конкретную причину ошибки.

Предполагается, что поле Marker в заголовке сообщений типа OPEN содержит только единицы. Ожидаемое значение поля Marker для остальных типов сообщений BGP определяется на основе присутствия дополнительного параметра Authentication Information в сообщении BGP OPEN и используемого механизма аутентификации (если поле Authentication Information присутствует в сообщении BGP OPEN). Если значение поля Marker в заголовке сообщения не совпадает с ожидаемым, происходит ошибка синхронизации и устанавливается значение Error Subcode = Connection Not Synchronized.

Если значение поля Length в заголовке сообщения меньше 19 или больше 4096, поле Length в заголовке сообщения OPEN, NOTIFICATION или UPDATE меньше минимального размера сообщения OPEN/NOTIFICATION/UPDATE или поле Length в заголовке сообщения KEEPALIVE не равно 19, устанавливается значение Error Subcode = Bad Message Length. Поле данных содержит ошибочное значение поля Length.

Если поле Type в заголовке сообщения не удаётся распознать, устанавливается Error Subcode = Bad Message Type и поле данных содержит ошибочное значение поля Type.

6.2 Обработка ошибок в сообщениях OPEN.
При обнаружении какой-либо ошибки в процессе обработки сообщений OPEN генерируется сообщение NOTIFICATION с Error Code = OPEN Message Error. Поле Error Subcode показывает конкретную причину ошибки.

Если версия, указанная в поле Version полученного сообщения OPEN не поддерживается, устанавливается значение Error Subcode = Unsupported Version Number. Поле данных содержит 2-октетное беззнаковое целое, которое показывает максимальный номер локально поддерживаемой версии BGP (который меньше номера версии, указанного в принятом сообщении OPEN).

Если поле Autonomous System в сообщении OPEN содержит неприемлемое значение, устанавливается Error Subcode = Bad Peer AS. Определение приемлемости номеров AS выходит за пределы рассмотрения этого документа.

Если неприемлемо значение поля Hold Time в сообщении OPEN, в поле Error Subcode **должно** быть установлено значение Unacceptable Hold Time. Реализации протокола **должны** отвергать значения Hold Time 1 или 2 секунды. Протокол **может** отвергать любые значения Hold Time. Реализация протокола, принимающая значение Hold Time, **должна** использовать согласованное значение Hold Time.

Если поле BGP Identifier в сообщении OPEN синтаксически некорректно, устанавливается значение Error Subcode = Bad BGP Identifier. Синтаксическая корректность означает, что поле BGP Identifier содержит корректный IP-адрес хоста.

Если один из дополнительных параметров (Optional Parameter) в сообщении OPEN не удалось распознать, устанавливается значение Error Subcode = Unsupported Optional Parameters.

Если сообщение OPEN содержит данные для аутентификации (как Optional Parameter), включается соответствующая процедура аутентификации. Если процедура аутентификации (на основе Authentication Code и Authentication Data) даёт отказ, устанавливается значение Error Subcode = Authentication Failure.

6.3 Обработка ошибок в сообщениях UPDATE.

При обнаружении какой-либо ошибки в процессе обработки сообщений UPDATE генерируется сообщение NOTIFICATION с Error Code = UPDATE Message Error. Поле Error Subcode показывает конкретную причину ошибки.

Контроль ошибок в сообщении UPDATE начинается с проверки атрибутов пути. Если значение Unfeasible Routes Length или Total Attribute Length слишком велико (т. е., Unfeasible Routes Length + Total Attribute Length + 23 превышает размер сообщения - Length), устанавливается значение Error Subcode = Malformed Attribute List.

Если любой из распознанных атрибутов имеет значение Attribute Flags, конфликтующее с Attribute Type Code, устанавливается значение Error Subcode = Attribute Flags Error. Поле данных содержит ошибочный атрибут (тип, размер и значение).

Если любой из распознанных атрибутов имеет значение Attribute Length, не соответствующее ожидаемой длине (на основе кода типа атрибута), устанавливается значение Error Subcode = Attribute Length Error. Поле данных содержит ошибочный атрибут (тип, размер и значение).

Если отсутствует какой-либо из обязательных хорошо известных атрибутов, устанавливается значение Error Subcode = Missing Well-known Attribute. Поле данных содержит значение Attribute Type Code для отсутствующего атрибута.

Если не распознан какой-либо из обязательных хорошо известных атрибутов, устанавливается значение Error Subcode = Unrecognized Well-known Attribute. Поле данных содержит нераспознанный атрибут (тип, размер и значение).

Если атрибут ORIGIN имеет неопределённое значение, устанавливается Error Subcode = Invalid Origin Attribute. Поле данных содержит нераспознанный атрибут (тип, размер и значение).

Если атрибут NEXT_HOP синтаксически некорректен, устанавливается Error Subcode = Invalid NEXT_HOP Attribute. Поле данных содержит некорректный атрибут (тип, размер и значение). Синтаксическая корректность означает, что атрибут NEXT_HOP представляет корректный IP-адрес хоста. Семантическая корректность применима только к внешним соединениям BGP. Это означает, что интерфейс, связанный с IP-адресом, который указан атрибутом NEXT_HOP, относится к той же подсети, в которой находится принимающий узел BGP, но его адрес не совпадает с IP-адресом принимающего узла BGP. Если атрибут NEXT_HOP семантически некорректен, ошибка должна протоколироваться и маршрут следует игнорировать (в этом случае сообщение NOTIFICATION не передаётся).

Проверяется семантическая корректность атрибута AS_PATH. Если путь семантически некорректен, устанавливается значение Error Subcode = Malformed AS_PATH.

Если дополнительный атрибут распознан, проверяется значение этого атрибута. При обнаружении ошибки атрибут отбрасывается и устанавливается значение Error Subcode = Optional Attribute Error. Поле данных содержит ошибочный атрибут (тип, размер и значение).

Если какой-либо из атрибутов появляется в сообщении UPDATE более одного раза, устанавливается значение Error Subcode = Malformed Attribute List.

Проверяется семантическая корректность поля NLRI в сообщении UPDATE. При обнаружении ошибки устанавливается значение Error Subcode = Invalid Network Field.

6.4 Обработка ошибок в сообщениях NOTIFICATION.

Если узел передаёт сообщение NOTIFICATION и при этом возникает ошибка, не существует способа уведомить об этой ошибке путём передачи последующего сообщения NOTIFICATION. Все ошибки такого типа (нераспознанное значение Error Code или Error Subcode) должны протоколироваться и передаваться администратору узла, отправившего ошибочное сообщение. Способы такого протоколирования и уведомления не рассматриваются в данном документе.

6.5 Обработка ошибок Hold Timer Expired.

Если система не получает сообщений KEEPALIVE и/или UPDATE и/или NOTIFICATION в течение периода, заданного полем Hold Time в сообщении OPEN, передаётся сообщение NOTIFICATION с кодом ошибки Hold Timer Expired и соединение BGP закрывается.

6.6 Обработка ошибок конечного автомата.

Любая ошибка, обнаруженная конечным автоматом (Finite State Machine - FSM) BGP (например, неожиданное событие), приводит к генерации сообщения NOTIFICATION с Error Code = Finite State Machine Error.

6.7 Разрыв соединения.

При отсутствии каких-либо фатальных ошибок из числа описанных выше узел BGP может в любой момент закрыть соединение BGP, передав партнёру сообщение NOTIFICATION с Error Code = Cease. Однако такие сообщения не должны использоваться при возникновении какой-либо из перечисленных выше фатальных ошибок.

6.8 Обнаружение конфликтов при соединении.

Если пара узлов BGP пытается одновременно организовать соединение TCP друг с другом, между узлами такой пары могут возникнуть два параллельных соединения. Будем называть такую ситуацию конфликтом при соединении. Очевидно, что при возникновении такого конфликта одно из соединений должно быть закрыто.

Выбор одного из пары соединений для закрытия базируется на соглашении об идентификаторах BGP. При возникновении конфликта сравниваются значения BGP Identifier вовлечённых в конфликт узлов и сохраняется только то соединение, которое было инициировано узлом BGP с большим значением BGP Identifier.

При получении сообщения OPEN локальная система **должна** проверить все свои соединения, находящиеся в состоянии OpenConfirm. Узел BGP может также проверить соединения, которые находятся в состоянии OpenSent, если он имеет информацию о значении BGP Identifier узла на противоположной стороне соединения (эта информация получается с помощью других протоколов). Если какое-либо из этих соединений относится к удалённому узлу BGP, идентификатор которого совпадает со значением BGP Identifier в сообщении OPEN, локальные система выполняет следующие процедуры разрешения конфликта:

1. Значение BGP Identifier локальной системы сравнивается с идентификатором удалённого узла BGP, указанным в сообщении OPEN.
2. Если локальное значение BGP Identifier меньше удалённого, локальная система закрывает существующее соединение BGP (оно находится в состоянии OpenConfirm) и принимает соединение, инициированное удаленным узлом BGP.
3. Если идентификатор BGP локальной системы больше удалённого идентификатора, закрывается новое соединение (связанное с принятым сообщением OPEN) и продолжается использование существующего (в состоянии OpenConfirm).

При сравнении идентификаторов BGP их значения трактуются как беззнаковые целые числа (4 октета).

Если одно из участвующих в конфликте соединений BGP находится в состоянии Established, такой конфликт разрешается безусловным закрытием нового соединения. Отметим, что конфликты с соединениями, находящимися в состоянии Idle, Connect или Active обнаружить невозможно.

Закрытие соединения BGP в результате процедуры разрешения конфликта сопровождается передачей сообщений NOTIFICATION с Error Code = Cease.

7. Согласование версий BGP.

Узлы BGP могут согласовать версию протокола путём повторных попыток организации соединения BGP, используя в первой попытке высший номер, поддерживаемый локальной стороной. Если при попытке организации соединения возникает ошибка с Error Code = OPEN Message Error и Error Subcode = Unsupported Version Number, узел BGP имеет информацию о номере версии, который был использован при неудачной попытке, номере версии, которую пытался использовать партнёр, номере версии, переданном партнёром в сообщении NOTIFICATION, и номере версии, которую он поддерживает. Если номера одной или более версий из числа поддерживаемых обоими партнёрами совпадают, имеющаяся информация позволяет быстро определить максимальный поддерживаемый номер версии. Для поддержки согласования версии BGP в будущих версиях протокола должен сохраняться формат сообщений OPEN и NOTIFICATION.

8. Конечный автомат BGP.

В этой главе рассматривается работа BGP в терминах конечного автомата. Ниже приведено краткое описание и обзор состояний BGP, определённых FSM. Краткое описание BGP FSM дано в Приложении 1.

Изначально BGP находится в состоянии Idle.

Состояние Idle

В этом состоянии BGP отвергает все входящие соединения BGP. Для партнера не выделяется никаких ресурсов. В ответ на событие Start (инициируется системой или оператором) локальная система инициализирует все ресурсы BGP, запускает таймер ConnectRetry, инициирует транспортное соединение с другим узлом BGP, начинает прослушивание входящих соединений от удалённых узлов BGP и переходит в состояние Connect. Значение таймера ConnectRetry определяется локальными условиями, но должно быть достаточно велико для того, чтобы обеспечить инициализацию TCP.

Если узел BGP обнаруживает ошибку, он закрывает соединение и переходит в состояние Idle. Выход из состояния Idle требует генерации нового события Start. Если эти события генерируются автоматически, возникновение ошибки BGP может привести к автогенерации попыток выхода из состояния Idle (flapping) на узле. Во избежание таких ситуаций рекомендуется генерировать события Start с некоторой задержкой после перехода узла в состояние Idle в результате ошибки. Для узла, перешедшего в состояние Idle в результате ошибки, интервал между генерацией последовательных событий Start (если они генерируются автоматически) должен возрастать экспоненциально. Начальная задержка должна составлять 60 секунд и после каждой попытки пауза должна удваиваться.

Все остальные события в состоянии Idle игнорируются.

Состояние Connect

В этом состоянии BGP ожидает завершения процесса организации транспортного соединения.

При успешном соединении на транспортном уровне локальная система сбрасывает таймер ConnectRetry, завершает инициализацию, передаёт партнёру сообщение OPEN и переходит в состояние OpenSent.

При отказе в соединении (например, тайм-аут повторной передачи), локальная система заново запускает таймер ConnectRetry, продолжает прослушивать попытки организации соединений от удалённых узлов BGP и переходит в состояние Active.

В случае завершения отсчёта таймера ConnectRetry локальная система запускает его заново, инициирует транспортное соединение с другим узлом BGP, продолжая прослушивать вызовы от удалённых узлов BGP, и остаётся в состоянии Connect.

События Start игнорируются в состоянии Connect[1].

В ответ на все прочие события (инициируемые системой или оператором) локальная система освобождает все ресурсы BGP, связанные с этим соединением и переходит в состояние Idle.

Состояние Active

В этом состоянии BGP пытается приобрести партнера, инициируя транспортное соединение.

При успешном соединении локальная система сбрасывает таймер ConnectRetry, завершает инициализацию, передаёт партнёру сообщение OPEN, устанавливает для Hold Timer достаточно большое значение и переходит в состояние OpenSent. Рекомендуется устанавливать для таймера удержания значение Hold Timer = 4 минуты.

При завершении отсчёта таймера ConnectRetry локальная система запускает этот таймер снова, инициирует транспортное соединение с другим узлом BGP, продолжая прослушивать входящие соединения от удалённых узлов BGP, и переходит в состояние Connect.

Если локальная система обнаруживает попытки удалённого узла BGP связаться с ней и IP-адрес удалённого узла не совпадает с ожидаемым, локальная система заново запускает таймер ConnectRetry, отвергает попытку соединения, продолжая прослушивать попытки соединений от удалённых узлов BGP и сохраняя состояние Active.

События Start игнорируются в состоянии Active.

В ответ на все прочие события (инициируемые системой или оператором) локальная система освобождает все ресурсы BGP, связанные с этим соединением и переходит в состояние Idle.

Состояние OpenSent

В этом состоянии BGP ждёт сообщения OPEN от своего партнёра. При получении сообщения OPEN проверяется корректность всех полей этого сообщения. Если при проверке заголовка или содержимого сообщения OPEN обнаруживаются ошибки (см. параграф 6.2 Обработка ошибок в сообщениях OPEN.) или возникает конфликт при организации соединения (см. параграф 6.8 Обнаружение конфликтов при соединении.), локальная система передаёт сообщение NOTIFICATION и переходит в состояние Idle.

Если в сообщении OPEN не обнаружено ошибок, узел BGP передаёт сообщение KEEPALIVE и устанавливает таймер KeepAlive. Для таймера Hold Timer, значение которого изначально устанавливается достаточно большим (см. выше), выбирается согласованное значение (см. параграф 4.2 Формат сообщений OPEN). Если согласовано нулевое значение Hold Time, таймеры Hold Time и KeepAlive не используются. Если значение поля Autonomous System совпадает с номером локальной AS, соединение является внутренним, в остальных случаях соединение является внешним (это будет влиять на обработку сообщений UPDATE, описанную ниже). После этого состояние меняется на OpenConfirm.

Если приходит уведомление о разрыве соединения от нижележащего протокола транспортного уровня, локальная система закрывает соединение BGP, заново запускает таймер ConnectRetry, продолжая слушать попытки соединений от удалённых узлов BGP, и переходит в состояние Active.

По истечении времени Hold Timer локальная система передаёт сообщение NOTIFICATION с кодом ошибки Hold Timer Expired и переходит в состояние Idle.

В ответ на событие Stop (инициируется системой или оператором) локальная система передаёт сообщение NOTIFICATION с Error Code = Cease и переходит в состояние Idle.

События Start игнорируются в состоянии OpenSent.

В ответ на все остальные события локальная система передаёт сообщение NOTIFICATION с Error Code = Finite State Machine Error и переходит в состояние Idle.

Всякий раз, когда BGP переходит из состояния OpenSent в состояние Idle, закрывается соединение BGP (и транспортное соединение) и освобождаются все ресурсы, связанные с этим соединением.

Состояние OpenConfirm

В этом состоянии BGP ожидает сообщений KEEPALIVE или NOTIFICATION.

Получив сообщение KEEPALIVE, система переходит в состояние Established.

По истечении времени Hold Timer до прихода сообщения KEEPALIVE, локальная система передаёт сообщение NOTIFICATION с кодом ошибки Hold Timer Expired и переходит в состояние Idle.

Получив сообщение NOTIFICATION, система переходит в состояние Idle.

По истечении времени KeepAlive локальная система передаёт сообщение KEEPALIVE и заново запускает таймер KeepAlive.

При получении от транспортного протокола уведомления о разрыве соединения локальная система переходит в состояние Idle.

В ответ на событие Stop (инициируется системой или оператором) локальная система передаёт сообщение NOTIFICATION с Error Code = Cease и переходит в состояние Idle.

События Start игнорируются в состоянии OpenConfirm.

В ответ на все остальные события локальная система передаёт сообщение NOTIFICATION с Error Code = Finite State Machine Error и переходит в состояние Idle.

Всякий раз, когда BGP переходит из состояния OpenConfirm в состояние Idle, закрывается соединение BGP (и транспортное соединение) и освобождаются все ресурсы, связанные с этим соединением.

Состояние Established

В состоянии Established узел BGP может обмениваться со своим партнёром сообщениями UPDATE, NOTIFICATION и KEEPALIVE.

Если локальная система получает сообщение UPDATE или KEEPALIVE, она заново запускает таймер Hold Timer (если согласованное значение Hold Time не равно нулю).

Получив сообщение NOTIFICATION, система переходит в состояние Idle.

Если локальная система получает сообщение UPDATE и процедура обработки ошибок в сообщениях UPDATE (см. параграф 6.3 Обработка ошибок в сообщениях UPDATE.) обнаруживает ошибку, локальная система передаёт сообщение NOTIFICATION и переходит в состояние Idle.

При получении от нижележащего транспортного протокола уведомления о разрыве соединения локальная система переходит в состояние Idle.

[1] В исходном документе ошибочно указано состояние Active. *Прим. перев.*

По истечении времени Hold Timer до прихода сообщения KEEPALIVE, локальная система передаёт сообщение NOTIFICATION с кодом ошибки Hold Timer Expired и переходит в состояние Idle.
По истечении времени KeepAlive локальная система передаёт сообщение KEEPALIVE и заново запускает таймер KeepAlive.
Передав сообщение KEEPALIVE или UPDATE, локальная система заново запускает таймер KeepAlive, если согласованное значение Hold Time не равно нулю.
В ответ на событие Stop (инициируется системой или оператором) локальная система передаёт сообщение NOTIFICATION с Error Code = Cease и переходит в состояние Idle.
События Start игнорируются в состоянии Established.
В ответ на все остальные события локальная система передаёт сообщение NOTIFICATION с Error Code = Finite State Machine Error и переходит в состояние Idle.
Всякий раз, когда BGP переходит из состояния Established в состояние Idle, закрывается соединение BGP (и транспортное соединение) и освобождаются все ресурсы, связанные с этим соединением.

9. Обработка сообщений UPDATE

Сообщение UPDATE может быть получено только в состоянии Established. При получении сообщения UPDATE проверяется корректность каждого поля в соответствии с параграфом 6.3 Обработка ошибок в сообщениях UPDATE..

Если не удаётся распознать дополнительные непереходные атрибуты, они просто игнорируются. Если не удаётся распознать дополнительные переходные атрибуты, устанавливается значение Partial=1 в поле флагов атрибута (третий по старшинству бит) и атрибут сохраняется для передачи другим узлам BGP.

Если все дополнительные атрибуты распознаны и имеют корректные значения, тогда (в зависимости от типа дополнительного атрибута) атрибуты обрабатываются локально, сохраняются и обновляются (при необходимости) для последующей передачи другим узлам BGP.

Если сообщение UPDATE содержит непустое поле WITHDRAWN ROUTES (отзываемые маршруты), ранее анонсированные маршруты, чьи адресаты указаны префиксами в данном поле, удаляются из таблицы Adj-RIB-In. Узлу BGP следует запустить Decision Process, поскольку анонсированные ранее маршруты больше не являются доступными.

Если сообщение UPDATE содержит доступный маршрут, это маршрут следует поместить в таблицу Adj-RIB-In и выполнить по отношению к нему перечисленные ниже операции:

i) Если поле NLRI идентично одному из маршрутов, хранящихся в Adj-RIB-In, новый маршрут должен использоваться взамен имеющегося в таблице Adj-RIB-In (т.е., старый маршрут отзывается). Узел BGP должен запустить Decision Process, поскольку старый маршруты больше не может использоваться.

ii) Если новый маршрут перекрывается с маршрутом, включённым ранее (см. 9.1.4 Перекрывающиеся маршруты) в таблицу Adj-RIB-In, узел BGP должен запустить Decision Process, поскольку более специфичный маршрут, явно указанный, как часть менее специфичного, недоступен для использования.

iii) Если атрибуты пути нового маршрута идентичны атрибутам пути маршрута, содержащегося в Adj-RIB-In, и новый маршрут более специфичен (см. 9.1.4 Перекрывающиеся маршруты), чем старый маршрут, выполнять какие-либо специальные действия не требуется.

iv) Если значение NLRI нового маршрута не присутствует ни в одном из имеющихся в таблице Adj-RIB-In маршрутов, этот маршрут просто добавляется в таблицу Adj-RIB-In и узел BGP должен запустить Decision Process.

v) Если новый маршрут перекрывается с более специфичным маршрутом (см. 9.1.4 Перекрывающиеся маршруты), указанным в таблице Adj-RIB-In, узел BGP должен запустить Decision Process для множества адресатов, связанных с менее специфичным маршрутом.

9.1 Процесс принятия решений

Процесс принятия решений (Decision Process) обеспечивает выбор маршрутов для последующего анонсирования путём применения правил, заданных в локальной базе PIB (Policy Information Base), к маршрутам из таблицы Adj-RIB-In. Результатом процесса является набор маршрутов, которые будут анонсироваться всем партнёрам. - эти маршруты хранятся в таблице Adj-RIB-Out.

Процесс выбора формализуется путём определения функций, принимающих атрибуты данного маршрута в качестве аргументов и возвращающих неотрицательное целое число, которое задаёт уровень предпочтения для данного маршрута. Функция, вычисляющая уровень предпочтения для данного маршрута, не должна использовать в качестве входных данных сведений о наличии или отсутствии других маршрутов и атрибутах пути других маршрутов. После использования этой функции для всех маршрутов выбор пути сводится к сравнению уровней предпочтения и выбору максимального значения.

Процесс выбора применяется ко всем маршрутам из таблицы Adj-RIB-In и отвечает за:

- Выбор маршрутов, анонсируемых узлам BGP в локальной AS;
- Выбор маршрутов, анонсируемых узлам BGP в соседних AS;
- Агрегирование маршрутов и снижение объёма маршрутных данных.

Decision Process делится на три фазы, каждая из которых включается определёнными событиями:

Фаза 1 отвечает за расчёт уровня предпочтения для каждого маршрута, полученного от узла BGP, расположенного в соседней AS, и анонсирование узлам BGP в локальной AS маршрутов с максимальным уровнем предпочтения для каждого из адресатов.

Фаза 2 начинается по завершении фазы 1 и отвечает за выбор лучшего маршрута из числа доступных для каждого адресата, а также включение выбранных маршрутов в подходящие Loc-RIB.

Фаза 3 начинается после обновления Loc-RIB и отвечает за распространение маршрутов из Loc-RIB всем партнёрам., расположенным в соседних AS, в соответствии с политикой, содержащейся в PIB. На этой фазе также может выполняться объединение маршрутов и снижение объёма маршрутной информации.

9.1.1 Фаза 1 - расчёт уровня предпочтения

Фазу 1 следует активизировать всякий раз, когда локальный узел BGP получает сообщение UPDATE от партнера, расположенного в соседней AS, которое анонсирует новый маршрут или замену/отзыв существующего маршрута.

Фаза 1 представляет собой автономный процесс, который завершается после выполнения всех требуемых операций.

Функция, используемая в фазе 1, должна заблокировать таблицу Adj-RIB-In прежде, чем начать работу с содержащимися в ней маршрутами и снять блокировку по завершении расчётов.

При получении нового маршрута или замене доступного локальный узел BGP должен определять уровень предпочтения для полученного маршрута. Если маршрут получен от узла BGP в локальной AS, в качестве уровня предпочтения следует брать значение атрибута LOCAL_PREF или рассчитывать этот уровень на основе конфигурационных параметров политики. Если маршрут получен от узла BGP в соседней AS, уровень предпочтения следует рассчитывать на основе конфигурационных параметров политики. Метод определения уровня предпочтения задаётся локальными условиями. Локальному узлу следует запустить внутренний процесс обновления (см. 9.2.1 Внутренние обновления) для выбора и анонсирования маршрута с высшим уровнем предпочтения.

9.1.2 Фаза 2 - выбор маршрута

Фаза 2 выполняется после завершения фазы 1 и представляет собой независимый процесс, который завершается после выполнения всех необходимых операций. В фазе 2 используются все маршруты из таблицы Adj-RIBs-In, включая маршруты, полученные от узлов BGP в локальной AS и соседних автономных системах.

Фаза 2 не должна начинаться, пока не будет завершено принятие решения для фазы 3. Фаза 2 должна блокировать таблицу Adj-RIBs-In перед началом работы и снимать блокировку по завершении работы.

Если атрибут NEXT_HOP маршрута BGP указывает адрес, к которому локальный узел BGP не имеет маршрута в таблице Loc-RIB, этот маршрут BGP **следует** исключать из обработки в фазе 2.

Для каждого набора адресатов, к которому существует доступный маршрут в таблице Adj-RIBs-In, локальный узел BGP должен убедиться, что выполняется хотя бы одно из перечисленных ниже условий:

a) маршрут имеет высший уровень предпочтения из всего набора путей к этому адресату;

b) маршрут является единственным для данного адресата;

c) маршрут выбран в результате применения в фазе 2 правил «разрыва связей» (tie breaking) описанных ниже (9.1.2.1 Разрыв связей (фаза 2).

После этого локальному узлу **следует** поместить маршрут в таблицу Loc-RIB, заменяя все маршруты к тому же адресату, которые будут обнаружены в Loc-RIB. Локальный узел **должен** определить следующий маршрутизатор (immediate next hop) для адреса, указанного атрибутом NEXT_HOP выбранного маршрута, запрашивая IGP и выбирая один из возможных путей в IGP. Адрес этого маршрутизатора **должен** использоваться при включении выбранного маршрута в таблицу Loc-RIB. Если маршрут к адресу, указанному атрибутом NEXT_HOP, меняется так, что изменяется адрес next hop, требуется повторить описанную выше процедуру выбора маршрута.

Недоступные маршруты должны удаляться из таблицы Loc-RIB и Adj-RIBs-In.

9.1.2.1 Разрыв связей (фаза 2)

В таблице Adj-RIBs-In узла BGP может храниться несколько маршрутов к одному адресату, имеющих одинаковый уровень предпочтения. Локальный узел может выбрать только один из таких маршрутов для включения в таблицу Loc-RIB. В рассмотрение принимаются все маршруты с одинаковым уровнем предпочтения - как полученные от узлов BGP в соседних AS, так и принятые от узлов той же автономной системы.

В описанной ниже процедуре разрыва связей предполагается, что для каждого маршрута-кандидата все узлы BGP в автономной системе могут определить стоимость пути (внутренняя дистанция) до адреса, указанного атрибутом NEXT_HOP в данном маршруте. Разрыв связей выполняется по следующему алгоритму:

a) Если локальная система принимает во внимание атрибут MULTI_EXIT_DISC и маршруты-кандидаты отличаются значениями этого атрибута, следует выбирать маршрут с минимальным значением MULTI_EXIT_DISC.

b) В противном случае выбирается маршрут с минимальной стоимостью (внутренней дистанцией) до элемента, указанного атрибутом NEXT_HOP. Если стоимость нескольких маршрутов совпадает, процедура продолжается следующим образом:

   - если хотя бы один из таких маршрутов анонсирован узлом BGP в соседней AS, среди таких маршрутов выбирается путь, анонсированный узлом с наименьшим значением BGP Identifier из числа находящихся в соседней AS;

   - в противном случае выбирается маршрут, анонсированный узлом с минимальным значением BGP Identifier.

9.1.3 Фаза 3 - распространение информации о маршруте

Фаза 3 выполняется после завершения фазы 2 или при возникновении одного из следующих событий:

a) изменение маршрутов к локальным адресатам в таблице Loc-RIB;

b) информация об изменении локально сгенерированных маршрутов получена с помощью внешнего BGP;

c) организовано новое соединение между парой узлов BGP.

Фаза 3 представляет собой самостоятельный процесс, завершающийся после выполнения требуемых операций. Функция Routing Decision фазы 3 должна быть заблокирована, пока не завершится принятие решения в фазе 2.

Все маршруты из таблицы Loc-RIB должны быть обработаны и включены в соответствующие записи связанной таблицы Adj-RIBs-Out. Возможно использование методов агрегирования маршрутов и снижения объёма маршрутных данных (см. 9.2.4.1 Снижение объёма информации).

Для поддержки будущих вариантов использования групповой адресации между AS узлам BGP, принимающим участие в multicast-маршрутизации между AS, следует анонсировать маршруты, полученные от одного из своих внешних партнёров и, если такой маршрут включён в таблицу Loc-RIB данного узла, анонсировать его также партнёру, от которого маршрут получен. Для узлов BGP, не принимающих участия в multicast-маршрутизации между AS, такое анонсирование является необязательным. При передаче таких анонсов атрибут NEXT_HOP должен содержать адрес партнера. Реализация протокола может также оптимизировать рассылку таких анонсов, отсекая часть информации в атрибуте AS_PATH и оставляя только номер своей AS и партнера, которому анонсируется маршрут (при этом следует устанавливать для атрибута ORIGIN значение INCOMPLETE). Кроме того, не требуется передавать дополнительные и необязательные атрибуты пути в таких анонсах.

По завершении обновлений таблиц Adj-RIBs-Out и FIB (Forwarding Information Base - база рассылки информации) локальному узлу BGP следует активизировать процесс внешнего обновления 9.2.2.

9.1.4 Перекрывающиеся маршруты

Узел BGP может передавать маршруты с перекрывающимися NLRI другому узлу BGP. Перекрытие NLRI происходит в тех случаях, когда множество адресатов отображается в несоответствующее множество маршрутов. Поскольку BGP представляет NLRI с использованием префиксов IP, перекрытия всегда могут быть выражены как подмножества. Маршрут, описывающий более узкое множество адресатов (более длинный префикс) будем называть более специфичным по сравнению с маршрутом, описывающим более широкое множество адресатов (префикс короче), - такие маршруты будем называть менее специфичными.

Отношения предпочтительности позволяют разделить менее специфичный маршрут на 2 части:

- множество адресатов, описываемое менее специфичным маршрутом, и
- множество адресатов, описываемое перекрытием менее специфичного и более специфичного маршрутов.

Когда перекрывающиеся маршруты присутствуют в одной таблице Adj-RIB-In, более специфичные маршруты должны следовать перед менее специфичными.

Набор адресатов, описываемый перекрытием, представляет часть менее специфичного маршрута, которая доступна, но не используется. Если более специфичный маршрут перестаёт работать, описываемые перекрытием адресаты остаются доступными через менее специфичный маршрут.

Если узел BGP получает перекрывающиеся маршруты, процессу выбора маршрутов (Decision Process) следует принимать во внимание семантику перекрывающихся маршрутов. В частности, если узел BGP воспринимает менее специфичный маршрут и отклоняет более специфичный от того же узла, адресаты, представленные перекрытием, могут не пересылаться в AS, указанные атрибутом AS_PATH такого маршрута. Следовательно, узел BGP может выбирать следующие варианты:

a) установить оба маршрута;

b) установить только более специфичный маршрут;

c) установить только неперекрывающуюся часть менее специфичного маршрута (т. е., деагрегировать его)

d) объединить два маршрута и установить агрегированный маршрут;

e) установить только менее специфичный маршрут;

f) не устанавливать ни один из маршрутов.

Если узел BGP выбирает вариант e), он должен добавить в маршрут атрибут ATOMIC_AGGREGATE (маршрут с таким атрибутом не может быть деагрегирован, т. е., значение NLRI такого маршрута не может быть более специфичным). Пересылка по такому маршруту не гарантирует, что пакеты IP будут на самом деле проходить только через те AS, которые указаны в атрибуте AS_PATH этого маршрута. Если узел BGP выбирает вариант a), он не должен анонсировать более общий маршрут без более специфичного.

9.2 Процесс передачи обновлений

Процесс передачи обновлений (Update-Send) отвечает за анонсирование сообщений UPDATE всем партнёрам. Например, он распространяет маршруты, выбранные Decision Process, другим узлам BGP, которые могут располагаться в той же или соседних AS. Правила для обмена информацией между узлами BGP, расположенными в различных AS, приводятся в параграфе 9.2.2, а правила обмена информацией внутри автономной системы - в параграфе 9.2.1.

Распространение маршрутных данных между узлами BGP в одной AS будем называть внутренним распределением.

9.2.1 Внутренние обновления

Процесс внутреннего обновления обеспечивает распространение маршрутной информации между узлами BGP, расположенными в локальной AS.

Когда узел BGP получает сообщение UPDATE от другого узла BGP, расположенного в той же AS, принимающий узел не должен перераспределять информацию, содержащуюся в сообщении UPDATE другим узлам BGP той же AS.

Когда узел BGP получает новый маршрут от узла BGP из соседней AS, он должен анонсировать этот маршрут всем другим узлам BGP в своей AS путём рассылки сообщений UPDATE, если справедливо любое из условий:

1) уровень предпочтения полученного локальным узлом BGP нового маршрута выше уровня, присвоенного другому маршруту, который был получен от узла BGP в соседней AS

2) нет других маршрутов, полученных от узлов BGP в соседних AS

3) полученный недавно маршрут выбран в результате разрыва связи между несколькими маршрутами, которые имели высший уровень предпочтения и вели к тому же адресату (см. параграф 9.2.1.1).

Когда узел BGP получает сообщение UPDATE с непустым полем WITHDRAWN ROUTES, он должен удалить из таблицы Adj-RIB-In все маршруты к адресатам, указанным этим полем (как префикс IP). Кроме того, узел должен выполнить следующие операции:

1) если соответствующий доступный маршрут не был ранее анонсирован, дополнительных действий не требуется;

2) если соответствующий доступный маршрут был анонсирован, нужно выполнить следующее:

   i. если выбранный для анонсирования новый маршрут имеет такое же значение NLRI, как недоступные маршруты, локальный узел BGP должен анонсировать замену маршрута;

   ii. если взятый на замену маршрут невозможно анонсировать, узел BGP должен включить адресатов недоступного маршрута (префикс IP) в поле WITHDRAWN ROUTES сообщения UPDATE и разослать это сообщение каждому партнёру, которому ранее была анонсирована доступность соответствующего маршрута.

Все возможные маршруты, которые анонсируются, должны быть помещены в таблицу Adj-RIBs-Out, а все недоступные маршруты следует удалить из Adj-RIBs-Out.

9.2.1.1 Разорванные связи (внутренние обновления)

Если локальный узел BGP имеет соединения с несколькими узлами BGP в соседних AS, это ведёт к существованию множества таблиц Adj-RIBs-In, связанных с этими партнёрами. Таблицы Adj-RIBs-In могут содержать несколько маршрутов к одному адресату и с одинаковым уровнем предпочтения, которые были анонсированы узлами BGP из соседних AS. Локальный узел BGP должен выбирать один из таких маршрутов, пользуясь приведёнными правилами:

a) Если маршруты-кандидаты отличаются только атрибутами NEXT_HOP и MULTI_EXIT_DISC, а локальная система принимает во внимание атрибут MULTI_EXIT_DISC, следует выбирать маршрут с минимальным значением атрибута MULTI_EXIT_DISC.

b) Если локальная система может установить стоимость пути к объекту, указанному атрибутом NEXT_HOP маршрута-кандидата, следует выбирать маршрут с минимальной стоимостью.

c) Во всех остальных случаях следует выбирать маршрут, анонсированный узлом BGP с наименьшим значением BGP Identifier.

9.2.2 Внешние обновления

Процесс внешнего обновления включает рассылку маршрутной информации узлам BGP, расположенным в соседних AS. В фазе 3 процесса выбора маршрутов узел BGP обновляет Adj-RIBs-Out и свою таблицу рассылки (Forwarding Table). Все новые маршруты и маршруты, недавно ставшие недоступными, для которых нет замены, должны анонсироваться узлам BGP, расположенным в соседних AS, с помощью сообщений UPDATE.

Все маршруты из таблицы Loc-RIB, помеченные как недоступные, должны быть удалены. Изменения в доступных адресатах своей AS также должны анонсироваться с помощью сообщений UPDATE.

9.2.3 Управление объёмом служебного трафика

Протокол BGP вынужден ограничивать объем служебного трафика (сообщения UPDATE) в целях снижения расхода полосы каналов, требуемой для анонсирования, и ресурсов системы, требуемых на этапе принятия решения (Decision Process) для обработки информации, содержащейся в сообщениях UPDATE.

9.2.3.1 Частота анонсирования маршрутов

Параметр MinRouteAdvertisementInterval определяет минимальное время, которое должно пройти между анонсированием маршрутов для конкретного адресата от одного узла BGP. Процедура ограничения скорости рассылки применяется независимо для каждого адресата, хотя значение MinRouteAdvertisementInterval устанавливается для узла BGP в целом.

Два сообщения UPDATE, передаваемые одним узлом BGP, когда он анонсирует возможные маршруты к некоторому общему набору адресатов, полученные от узлов BGP в соседних AS, должны быть разделены промежутком времени не менее MinRouteAdvertisementInterval. Очевидно, что для достижения в точности такого поведения требуется использовать отдельный таймер для каждого общего набора адресатов. Такой подход будет порождать недопустимую нагрузку (overhead). На практике подходит любой метод, обеспечивающий между двумя последовательными сообщениями UPDATE с анонсом доступного маршрута к некому множеству адресатов в соседние AS от одного узла BGP интервал не менее MinRouteAdvertisementInterval и способный гарантировать постоянное значение верхней границы для такого интервала.

Поскольку внутри AS требуется быстрое схождение маршрутов, эта процедура не применяется для маршрутов, полученных от внутренних узлов BGP. Чтобы избавиться от длительных «чёрных дыр», процедура не применяется при явном отзыве недоступных маршрутов (для которых адресат, указанный префиксом IP, включён в поле WITHDRAWN ROUTES сообщения UPDATE).

Эта процедура не ограничивает скорость выбора маршрута, внося лишь ограничение на частоту анонсирования. Если новый маршрут выбран несколько раз в течение ожидания MinRouteAdvertisementInterval, по завершении периода MinRouteAdvertisementInterval должен анонсироваться последний выбранный маршрут.

9.2.3.2 Частота обновления из исходной AS

Параметр MinASOriginationInterval определяет минимальный интервал времени между последовательными сообщениями UPDATE, которые содержат информацию об изменениях внутри AS анонсирующего узла BGP.

9.2.3.3 Флуктуации

Для снижения вероятности пиковой нагрузки при распределении сообщений BGP данным узлом BGP таймеры, связанные с MinASOriginationInterval, Keepalive и MinRouteAdvertisementInterval должны задаваться с использованием флуктуаций (jitter). Каждый узел BGP должен использовать одинаковое отклонение для всех перечисленных таймеров, независимо от адресатов, для которых будут передаваться сообщения (т. е., флуктуации задаются на уровне узла).

Величина флуктуации должна определяться умножением базовой величины на случайное значение от 0,75 до 1,0 (с равномерным распределением).

9.2.4 Эффективная организация маршрутных данных

После выбора маршрутных данных для анонсирования узел BGP может использовать несколько методов эффективной организации этих данных.

9.2.4.1 Снижение объёма информации

Снижение объёма информации означает снижение детализации контроля над политикой маршрутизации - после сжатия информации одни и те же правила будут применяться ко всем адресатам и путям одного класса.

Перечисленные способы позволяют снизить объем данных, помещаемых в Adj-RIBs-Out на этапе Decision Process:

a) **NLRI**. IP-адреса получателей могут быть представлены префиксами IP. В тех случаях, когда имеется соответствие между структурой адреса и системами, находящимися под управлением администратора AS, можно уменьшить размер NLRI, передаваемых в сообщениях UPDATE.

b) **AS_PATH**. Информация о пути AS может быть представлена как в упорядоченной (AS_SEQUENCE), так и в неупорядоченной (AS_SET) форме. AS_SET используется в алгоритме агрегирования маршрутов, описанном в параграфе 9.2.4.2. Это позволяет снизить объем данных AS_PATH за счёт однократного указания номера каждой AS (независимо от числа её упоминаний во множестве объединяемых AS_PATH).

   AS_SET означает, что адресаты, указанные в NLRI, могут быть достигнуты по пути, проходящему по крайней мере через некоторые из составляющих AS. AS_SET обеспечивают достаточную информацию для предотвращения петель в передаче маршрутных данных, однако могут теряться потенциально возможные пути, поскольку они больше не указываются явно в форме AS_SEQUENCE. На практике это не вызывает проблем, поскольку по прибытии одного пакета IP на край группы AS узел BGP в этой точке явно будет иметь более детальную информацию о пути и сможет различать отдельные маршруты к адресатам.

9.2.4.2 Агрегирование маршрутной информации

Агрегирование представляет собой процесс объединения характеристик нескольких разных маршрутов таким образом, чтобы их можно было анонсировать как единый маршрут. Агрегирование может выполняться как часть процесса принятия решения для снижения объёма маршрутных данных, помещаемых в Adj-RIBs-Out.

Агрегирование снижает объем информации, которую узел BGP должен сохранять и рассылать другим узлам BGP. Маршруты можно агрегировать путём применения описанной ниже процедуры раздельно к однотипным атрибутам пути и NLRI.

Маршруты с атрибутами MULTI_EXIT_DISC и NEXT_HOP не могут агрегироваться, если значения этих атрибутов не совпадают для всех маршрутов.

Атрибуты пути с различными кодами типа не могут быть агрегированы. Однотипные атрибуты пути могут агрегироваться в соответствии с приведёнными ниже правилами:

Атрибут ORIGIN. Если хотя бы один из агрегируемых маршрутов имеет ORIGIN = INCOMPLETE, для объединённого маршрута также должно устанавливаться ORIGIN = INCOMPLETE. Если хотя бы один из объединяемых маршрутов имеет значение ORIGIN = EGP, агрегированный маршрут также должен иметь значение EGP для этого атрибута. В остальных случаях для агрегированного маршрута устанавливается ORIGIN = INTERNAL.

Атрибут AS_PATH. Если у агрегируемых маршрутов совпадают значения AS_PATH, объединённый маршрут имеет такое же значение AS_PATH.

В целях объединения атрибутов AS_PATH будем моделировать каждую AS в атрибуте AS_PATH как пару <type, value>, где type определяет тип сегмента пути, к которому относится AS (например, AS_SEQUENCE, AS_SET), а value указывает номер AS. Если агрегируемые маршруты имеют разные атрибуты AS_PATH, агрегированный атрибут AS_PATH должен удовлетворять каждому из перечисленных ниже требований:

- Все пары типа AS_SEQUENCE объединённого атрибута AS_PATH должны присутствовать в каждом атрибуте AS_PATH исходного набора агрегируемых маршрутов.
- Все пары типа AS_SET объединённого атрибута AS_PATH должны присутствовать хотя бы в одном атрибуте AS_PATH исходного набора (как AS_SET или AS_SEQUENCE).
- Для любой пары X типа AS_SEQUENCE в агрегированном AS_PATH, которая предшествует паре Y агрегированного AS_PATH, X предшествует Y в каждом атрибуте AS_PATH исходного набора, который содержит Y (независимо от типа Y).

- Ни одна пара (независимо от её типа) не должна появляться в агрегированном AS_PATH более одного раза.

Разработчики могут выбирать любой алгоритм, который обеспечивает соответствие приведённым правилам. Соответствующая требованиям этого документа реализация должна выполнять по крайней мере описанный ниже алгоритм для выполнения приведённых требований:

- Определить наиболее длинную последовательность лидирующих пар (как описано выше), присутствующую в атрибутах AS_PATH всех объединяемых маршрутов и сделать её лидирующей в объединённом атрибуте AS_PATH.
- Установить для оставшихся пар из атрибутов AS_PATH объединяемых маршрутов тип AS_SET и присоединить их к агрегированному атрибуту AS_PATH.
- Если объединённый атрибут AS_PATH содержит несколько одинаковых пар (независимо от типа), лишние (все, кроме одной) пары типа AS_SET следует удалить из объединённого атрибута AS_PATH.

В Приложении 6 (параграф 6.8) рассмотрен другой алгоритм, удовлетворяющий приведённым здесь требованиям и обеспечивающим поддержку более сложных вариантов политики.

Атрибут ATOMIC_AGGREGATE. Если хотя бы один из объединяемых маршрутов имеет атрибут ATOMIC_AGGREGATE, объединённый маршрут также должен иметь этот атрибут.

Атрибут AGGREGATOR. Все атрибуты AGGREGATOR в объединяемых маршрутах следует игнорировать.

9.3 Критерии выбора маршрута

В общем случае рассмотрение дополнительных правил сравнения альтернативных маршрутов выходит за пределы данного документа. Однако имеются два исключения:

- Если локальная AS присутствует в пути AS нового маршрута, этот маршрут не может считаться лучше какого-либо из имеющихся путей. Нарушение этого правила ведёт к возникновению маршрутных петель.
- Для обеспечения эффективной распределенной обработки следует выбирать только маршруты, представляющиеся стабильными. Таким образом, AS должна избегать применения нестабильных маршрутов и не должна вносить скороспелых спонтанных изменений при выборе путей. Трактовка слов «нестабильный» и «скороспелый» в предыдущем предложении требует некоторого опыта, но, в принципе, достаточно понятна.

9.4 Порождение маршрутов BGP

Узел BGP может порождать (originate) маршруты BGP, помещая информацию, полученную из других источников (например, IGP), в BGP. Порождающий маршруты узел BGP должен указывать для таких маршрутов уровень предпочтения, используя для этого Decision Process (см. 9.1 Процесс принятия решений). Эти маршруты могут также рассылаться другим узлам BGP в локальной AS, как часть процесса внутреннего обновления (см. 9.2.1 Внутренние обновления). Решение о целесообразности рассылки полученной от других источников информации внутри AS с использованием BGP зависит от используемой в AS среды (например, типа IGP) и должно задаваться на уровне конфигурации.

Приложение 1. Переходы и действия BGP FSM.

В этом приложении рассматриваются переходы между состояниями BGP FSM, вызванные событиями BGP. Ниже приведён список таких состояний и событий для тех случаев, когда согласованное значение Hold Time отличается от нуля.

Состояния BGP

1 - Idle - бездействие
2 - Connect - соединение
3 - Active - активно
4 - OpenSent - передано сообщение OPEN
5 - OpenConfirm - подтверждено сообщение OPEN
6 - Established - соединение организовано

События BGP

1 - BGP Start - начало работы
2 - BGP Stop - остановка
3 - BGP Transport connection open - транспортное соединение открыто
4 - BGP Transport connection closed - транспортное соединение закрыто
5 - BGP Transport connection open failed - неудачная попытка открыть транспортное соединение
6 - BGP Transport fatal error - неисправимая ошибка на транспортном уровне
7 - ConnectRetry timer expired - завершён отсчёт таймера ConnectRetry
8 - Hold Timer expired - завершён отсчёт таймера Hold Timer
9 - KeepAlive timer expired - завершён отсчёт таймера KeepAlive
10 - Receive OPEN message - получено сообщение OPEN
11 - Receive KEEPALIVE message - получено сообщение KEEPALIVE
12 - Receive UPDATE messages - получено сообщение UPDATE
13 - Receive NOTIFICATION message - получено сообщение NOTIFICATION

В приведённой ниже таблице перечислены состояния и переходы BGP FSM, а также указаны действия в результате таких переходов.

Событие	Действия	Сообщение	Следующий этап
Idle (1)			
1	Инициализация ресурсов Запуск таймера ConnectRetry Инициирование транспортного соединения	Нет	2

прочие	Нет	Нет	1
Connect(2)			
1	Нет	Нет	2
3	Полная инициализация Сброс таймера ConnectRetry	OPEN	4
5	Перезапуск таймера ConnectRetry	Нет	3
7	Перезапуск таймера ConnectRetry	Нет	2
прочие	Освобождение ресурсов	Нет	1
Active (3)			
1	Нет	Нет	2
2	Полная инициализация Сброс таймера ConnectRetry	OPEN	4
5	Разрыв соединения Перезапуск таймера ConnectRetry		3
7	Перезапуск таймера ConnectRetry	Нет	2
прочие	Освобождение ресурсов	Нет	1
OpenSent(4)			
1	Нет	Нет	4
4	Разрыв транспортного соединения Перезапуск таймера ConnectRetry	Нет	3
6	Освобождение ресурсов	Нет	1
10	Обработка сообщения OPEN завершена успешно Неудача при обработке сообщения OPEN	KEEPALIVE NOTIFICATION	5 1
прочие	Разрыв транспортного соединения Освобождение ресурсов	NOTIFICATION	1
OpenConfirm (5)			
1	Нет	Нет	5
4	Освобождение ресурсов	Нет	1
6	Освобождение ресурсов	Нет	1
9	Перезапуск таймера KeepAlive	KEEPALIVE	5
11	Полная инициализация Перезапуск таймера Hold Timer	Нет	6
13	Разрыв транспортного соединения Освобождение ресурсов		1
прочие	Разрыв транспортного соединения Освобождение ресурсов	NOTIFICATION	1
Established (6)			
1	Нет	Нет	6
4	Освобождение ресурсов	Нет	1
6	Освобождение ресурсов	Нет	1
9	Перезапуск таймера KeepAlive	KEEPALIVE	6
11	Перезапуск таймера Hold Timer	KEEPALIVE	6
12	Обработка сообщения UPDATE завершена успешно Неудача при обработке сообщения UPDATE	UPDATE NOTIFICATION	1 1
13	Разрыв транспортного соединения Освобождение ресурсов		1
прочие	Разрыв транспортного соединения Освобождение ресурсов	NOTIFICATION	1

Ниже приведён сжатый вариант таблицы состояний и переходов.

События	Idle (1)	Connect (2)	Active (3)	OpenSent (4)	OpenConfirm (5)	Estab (6)
1	2	2	3	4	5	6
2	1	1	1	1	1	1
3	1	4	4	1	1	1
4	1	1	1	3	1	1
5	1	3	3	1	1	1
6	1	1	1	1	1	1
7	1	2	2	1	1	1
8	1	1	1	1	1	1
9	1	1	1	1	5	5
10	1	1	1	1 или 5	1	1
11	1	1	1	1	6	6
12	1	1	1	1	1	1 или 6
13	1	1	1	1	1	1

Приложение 2. Сравнение с RFC 1267

BGP-4 может работать в среде, где множество доступных адресатов может указываться с помощью одного префикса IP. Концепция классов сетей или подсетей чужеродна для BGP-4. Для поддержки работы с префиксами в BGP-4 изменена семантика и кодирование, связанное с атрибутом AS_PATH. В спецификацию добавлено определение

семантики, связанное с префиксами IP. Такое расширение позволяет BGP-4 поддерживать предложенную схему supernet [9].

Для упрощения настройки вводится новый атрибут LOCAL_PREF, упрощающий процедуру выбора маршрута.

Атрибут INTER_AS_METRIC переименован в MULTI_EXIT_DISC. Добавлен новый атрибут ATOMIC_AGGREGATE для управления возможностью деагрегирования маршрутов. Другой новый атрибут - AGGREGATOR - может добавляться в агрегированные маршруты, чтобы указать, какая AS и какой узел BGP в этой AS вызвали агрегирование.

Для обеспечения симметрии значение Hold Time согласуется на уровне соединений. Поддерживаются нулевые значения Hold Time.

Приложение 3. Сравнение с RFC 1163

Все изменения, перечисленные в Приложении 2, и изменения, указанные ниже.

Для обнаружения конфликтов при соединениях BGP и восстановления работы протокола добавлено новое поле BGP Identifier в сообщения OPEN. Для описания процедур детектирования и разрешения конфликтов при соединениях в документ добавлен новый параграф (6.8 Обнаружение конфликтов при соединении.).

Снято ограничение, требовавшее чтобы граничный маршрутизатор, указанный атрибутом пути NEXT_HOP, относился к той же AS, в которой находится узел BGP.

В новом документе оптимизировано и упрощено описание процедур обмена информацией о ранее доступных маршрутах.

Приложение 4. Сравнение с RFC 1105

Все изменения, перечисленные в Приложениях 2 и 3, а также перечисленные ниже отличия.

Потребовалось внесение незначительных изменений в конечный автомат RFC1105 для согласования с пользовательским интерфейсом TCP в системах 4.3 BSD.

Понятия и соотношения Up/Down/Horizontal, присутствующие в RFC1105, были исключены из протокола.

Внесён ряд изменений в формат сообщений RFC1105:

- Поле Hold Time было удалено из заголовка BGP и включено в сообщение OPEN.
- Поле номера версии было удалено из заголовка BGP и включено в сообщение OPEN.
- Из сообщений OPEN было удалено поле Link Type.
- Вместо подтверждений OPEN CONFIRM используются сообщения KEEPALIVE.
- Существенно изменён формат сообщений UPDATE, добавлены новые поля для поддержки множества атрибутов пути.
- Поле Marker было расширено и стало использоваться также для аутентификации.

Отметим, что достаточно часто протокол BGP, соответствующий RFC 1105, называют BGP-1, соответствующий RFC 1163 - BGP-2, а соответствующий RFC 1267 - BGP-3. Вариант BGP, описанный в этом документе, называют BGP-4.

Приложение 5. Опции TCP, которые могут использоваться с BGP

Если пользовательский интерфейс TCP в локальной системе поддерживает функцию TCP PUSH, каждое сообщение BGP следует передавать с установленным флагом PUSH. Установка флага приводит к ускорению передачи сообщений BGP.

Если пользовательский интерфейс TCP в локальной системе поддерживает предпочтения для соединений TCP, транспортные соединения BGP следует открывать с уровнем предпочтения Internetwork Control (110) [6].

Приложение 6. Рекомендации разработчикам

В этом приложении даются некоторые рекомендации разработчикам.

6.1 Множество префиксов сетей в одном сообщении

Протокол BGP позволяет указывать в одном сообщении множество адресных префиксов с одинаковым путём AS и адресом следующего маршрутизатора (next-hop). Настоятельно рекомендуется использовать эту возможность при реализации протокола. Передача сообщений с единственным префиксом существенно повышает уровень служебного трафика и увеличивает нагрузку при сканировании таблиц маршрутизации в случаях обновления для узлов BGP и других протоколов маршрутизации (увеличивается и объем передаваемых обновлений). Одним из способов создания сообщений с множеством префиксов для каждого пути AS на основе таблицы, не организованной по путям AS, является создание множества сообщений при сканировании таблицы. При обработке префиксов сообщение для связанных с префиксом пути AS и шлюза создаётся только в том случае, когда сообщения для такой пары путь-маршрутизатор не существует - в противном случае префикс просто добавляется в конец существующего сообщения. Если в сообщение уже нельзя добавить новый префикс по соображениям размера, имеющееся сообщение передаётся, а для префикса создаётся новое сообщение. После завершения сканирования всей таблицы маршрутов созданные сообщения передаются и выделенные для них ресурсы освобождаются. Максимальное сжатие при таком методе обеспечивается в тех случаях, когда все адресаты перекрываются адресными префиксами с одним шлюзом и атрибутом пути. В этом случае сообщение может содержать столько префиксов, сколько позволяет ограничение на размер сообщений (4096 октетов).

При работе с реализациями BGP, которые не поддерживают множества префиксов в одном сообщении, может потребоваться выполнение ряда операций для снижения нагрузки в результате лавинной рассылки данных, полученных при обретении нового партнера или существенном изменении сетевой топологии. Одним из способов такого снижения является ограничение частоты передачи обновлений. Это позволяет избавиться от избыточного сканирования таблиц для «мгновенного» обновления узлов BGP и других протоколов. Недостатком этого способа является увеличение задержек при распространении маршрутной информации. Выбор минимального интервала обновлений, который незначительно превышает время обработки множества сообщений, позволяет минимизировать задержку при распространении маршрутных данных. Наилучшим решением будет просмотр всех полученных сообщений до передачи обновлений.

6.2 Обработка сообщений при использовании потокового протокола

BGP использует протокол TCP в качестве транспортного механизма. Из-за потоковой природы TCP данные для каждого принятого сообщения не обязаны прибывать одновременно. Это может приводить к возникновению трудностей при обработке данных как сообщений, особенно в системах типа BSD Unix, где невозможно определить количество принятых, но ещё не обработанных данных.

Одним из вариантов решения проблемы является просмотр заголовка до обработки сообщения. Для типа KEEPALIVE сообщение состоит только из заголовка, остальные типы сообщений требуют предварительной проверки заголовка (в частности, общего размера сообщения). Если проверка даёт положительный результат, указанная в заголовке длина сообщения за вычетом размера заголовка даёт размер данных в сообщении, которые осталось прочитать. Реализации протокола, сталкивающиеся с проблемами при попытке чтения данных от партнера, могут установить для каждого партнера буфер сообщений (4096 байтов) и заполнять его данными, по мере их поступления от партнера.

6.3 Отказ от избыточных переключений маршрутов

Во избежание ненужных переключений маршрутов (route flapping) узлу BGP, которому нужно отозвать адресата и передать обновление с более (или менее) специфичным маршрутом, следует объединять анонсы в одно сообщение UPDATE.

6.4 Таймеры BGP

BGP использует 5 таймеров: ConnectRetry, Hold Time, KeepAlive, MinASOriginationInterval и MinRouteAdvertisementInterval. Для ConnectRetry предлагается устанавливать значение 120 секунд, для Hold Time - 90, для KeepAlive и MinRouteAdvertisementInterval - 30, а для MinASOriginationInterval - 15 секунд.

Реализация BGP **должна** обеспечивать возможность установки значений для всех таймеров.

6.5 Порядок атрибутов пути

Реализации, комбинирующие обновления, как описано выше (6.1 Множество префиксов сетей в одном сообщении), могут предпочесть просмотр всех атрибутов пути, представленных в определённом порядке. Такой подход позволяет быстро идентифицировать наборы атрибутов из разных обновлений, которые идентичны семантически. Для реализации такого подхода полезно упорядочивать атрибуты в соответствии с кодом типа (оптимизация не является обязательной).

6.6 Сортировка AS_SET

Другим полезным способом оптимизации является упорядочивание по номерам AS, найденным в атрибуте AS_SET. Такая оптимизация также не является обязательной.

6.7 Контроль за согласованием версий

Поскольку протокол BGP-4 может передавать агрегированные маршруты, которые не могут быть корректно представлены в BGP-3, реализации, поддерживающие BGP-4 и иные версии BGP, должны обеспечивать возможность работы только с BGP-4 независимо для каждого партнера.

6.8 Комплексное агрегирование AS_PATH

Реализация, обеспечивающая механизм агрегирования маршрутов с сохранением значительного количества данных о пути, может использовать описанную ниже процедуру.

Для объединения атрибутов AS_PATH двух маршрутов будем представлять каждую AS как пару <type, value>, где type указывает тип сегмента пути, к которому принадлежит AS (например, AS_SEQUENCE, AS_SET), а value задаёт номер AS. Если две пары <type, value> совпадают, они относятся к одной AS.

Алгоритм объединения двух атрибутов AS_PATH работает следующим образом:

a) Идентифицируется совпадение AS (как описано выше) в каждом атрибуте AS_PATH, которые находятся в том же относительном порядке для каждого атрибута AS_PATH. Две AS (X и Y) следуют в одинаковом порядке, если:

   - X предшествует Y в обоих атрибутах AS_PATH или Y предшествует X в обоих атрибутах AS_PATH.

b) Агрегированный атрибут AS_PATH состоит из AS, найденных на этапе (a) и представленных в том же порядке, который был обнаружен в объединяемых атрибутах AS_PATH. Если две последовательные AS, найденные на этапе (a), не следуют одна за другой непосредственно в каждом из объединяемых атрибутов AS_PATH, мешающие AS (AS, расположенные между двумя последовательно совпадающими AS) в обоих атрибутах объединяются в сегмент пути AS_SET, который содержит мешающие AS из обоих атрибутов AS_PATH. Этот сегмент пути помещается в комбинированном атрибуте между двумя последовательными AS, идентифицированными в пункте (a).

Если две последовательные AS, идентифицированные на этапе (a), непосредственно следуют одна за другой в одном атрибуте, но этого не наблюдается в другом, то мешающие AS второго атрибута комбинируются в сегмент пути AS_SET, который помещается между двумя последовательными AS в агрегированном атрибуте.

Если в результате применения описанной выше процедуры данный номер AS появляется в агрегированном атрибуте AS_PATH более одного раза, все вхождения этого номера, кроме последнего (самый правый) следует удалить из агрегированного атрибута PATH.

Литература

[1] Mills, D., "Exterior Gateway Protocol Formal Specification", RFC 904, BBN, April 1984.

[2] Rekhter, Y., "EGP and Policy Based Routing in the New NSFNET Backbone", RFC 1092, T.J. Watson Research Center, February 1989.

[3] Braun, H-W., "The NSFNET Routing Architecture", RFC 1093, MERIT/NSFNET Project, February 1989.

[4] Postel, J., "Transmission Control Protocol - DARPA Internet Program Protocol Specification", STD 7, RFC 793, DARPA, September 1981.

[5] Rekhter, Y., and P. Gross, "Application of the Border Gateway Protocol in the Internet", RFC 1772, T.J. Watson Research Center, IBM Corp., MCI, March 1995.

[6] Postel, J., "Internet Protocol - DARPA Internet Program Protocol Specification", STD 5, RFC 791, DARPA, September 1981.

[7] "Information Processing Systems - Telecommunications and Information Exchange between Systems - Protocol for Exchange of Inter-domain Routing Information among Intermediate Systems to Support Forwarding of ISO 8473 PDUs", ISO/IEC IS10747, 1993

[8] Fuller, V., Li, T., Yu, J., and K. Varadhan, "Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy", RFC 1519, BARRNet, cisco, MERIT, OARnet, September 1993

[9] Rekhter, Y., Li, T., "An Architecture for IP Address Allocation with CIDR", RFC 1518, T.J. Watson Research Center, Cisco, September 1993

Вопросы безопасности

Вопросы безопасности не рассматриваются в этом документе.

Адреса редакторов

Yakov Rekhter

T.J. Watson Research Center IBM Corporation

P.O. Box 704, Office H3-D40

Yorktown Heights, NY 10598

Phone: +1 914 784 7361

EMail: yakov@watson.ibm.com

Tony Li

Cisco Systems, Inc.

170 W. Tasman Dr.

San Jose, CA 95134

EMail: tli@cisco.com

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru