Network Working Group — Y. Rekhter, Ed.
Request for Comments: 4271 — T. Li, Ed.
Obsoletes: 1771 — S. Hares, Ed.
Category: Standards Track — January 2006

Протокол BGP-4

A Border Gateway Protocol 4 (BGP-4)

Статус документа

В этом документе содержится спецификация протокола, предложенного сообществу Internet. Документ служит приглашением к дискуссии в целях развития и совершенствования протокола. Текущее состояние стандартизации протокола вы можете узнать из документа "Internet Official Protocol Standards" (STD 1). Документ может распространяться без ограничений.

Тезисы

Этот документ посвящен обсуждению протокола BGP[1], который является протоколом маршрутизации между автономными системами (inter-Autonomous System routing protocol).

Основной функцией поддерживающей протокол 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 для принятия некоторых решений на уровне политики AS.

BGP-4 обеспечивает новые механизмы поддержки бесклассовой междоменной маршрутизации (CIDR[2]). Эти механизмы включают поддержку анонсирования группы адресатов с помощью префикса IP и позволяют обойтись без концепции «класса» сетей в рамках протокола BGP. BGP-4 также добавляет механизм объединения маршрутов, включающий объединение путей AS.

Этот документ прекращает действие RFC 1771.

Оглавление

[1]Border Gateway Protocol – протокол граничного шлюза.
[2]Classless Interdomain Routing — бесклассовая междоменная маршрутизация. *Прим. перев.*

1. Введение

Протокол граничного шлюза (BGP) является протоколом маршрутизации между автономными системами (AS).

Основной функцией поддерживающей протокол 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 для принятия некоторых решений на уровне политики AS.

BGP-4 обеспечивает новые механизмы поддержки бесклассовой междоменной маршрутизации (CIDR) [RFC1518, RFC1519]. Эти механизмы включают поддержку анонсирования группы адресатов с помощью префикса IP и позволяют обойтись без концепции «класса» сетей в рамках протокола BGP. BGP-4 также добавляет механизм объединения маршрутов, включающий объединение путей AS.

Маршрутная информация, передаваемая с использованием BGP поддерживает только парадигму пересылки на основе адреса получателя (destination-based forwarding paradigm), которая предполагает, что маршрутизатор пересылает пакеты, опираясь лишь на адрес получателя, содержащийся в заголовке IP-пакета. Это, в свою очередь, отражает набор правил политики, которые могут быть применены (или не применены) с использованием BGP. Протокол BGP может поддерживать только правила, соответствующие парадигме пересылки по адресу получателя.

1.1. Определения основных терминов

В этом разделе приводятся определения основных терминов, имеющих специфическое толкование в контексте протокола BGP и встречающихся в данном документе.

Adj-RIB-In

Необработанная маршрутная информация, которая анонсируется локальному узлу BGP его партнерами.

Adj-RIB-Out

Adj-RIBs-Out содержит маршруты, анонсируемые указанным партнерам BGP с помощью сообщений UPDATE, передаваемых локальным узлом BGP.

Autonomous System (AS) – автономная система

В соответствии с классическим определением автономная система представляет собой набор маршрутизаторов, находящихся под единым административным управлением, использующих один протокол внутридоменной

маршрутизации (interior gateway protocol или IGP) и общую метрику для определения маршрутизации пакетов внутри AS, а также использующих протокол междоменной маршрутизации для определения маршрутов пересылки пакетов в другие AS. С момента создания этого определения для AS стало общепринятым использование нескольких протоколов IGP, а в некоторых случаях и нескольких наборов метрик. Использование термина AS в таких случаях подчеркивает, что даже при наличии нескольких IGP и метрик администрирование AS с точки зрения других автономных систем представляет собой единый согласованный план маршрутизации и согласованную картину адресатов, доступных через данную AS.

BGP Identifier – идентификатор BGP

4-октетное целое число без знака, идентифицирующее узел BGP, отправляющий сообщение BGP. Данный узел BGP устанавливает в качестве значения BGP Identifier адрес IP, присвоенный узлу. Значение идентификатора BGP задается при старте системы и совпадает для всех локальных интерфейсов и самого узла BGP.

BGP speaker – узел BGP

Маршрутизатор, поддерживающий протокол BGP.

EBGP

External BGP (BGP-соединение с внешним партнером).

External peer – внешний партнер

Партнер BGP, относящийся к другой (внешней по отношению к локальной системе) AS.

Feasible route – подходящий маршрут

Анонсируемый маршрут, который пригоден для использования получателем.

IBGP

Internal BGP (BGP-соединение с внутренним партнером).

Internal peer – внутренний партнер

Партнер BGP, относящийся к той же AS, что и локальная система.

IGP – протокол внутреннего шлюза

Interior Gateway Protocol – протокол маршрутизации, используемый для обмена маршрутной информацией между маршрутизаторами одной AS.

Loc-RIB

Loc-RIB содержит маршруты, выбранные процессом принятия решений (Decision Process) локального узла BGP.

NLRI

Network Layer Reachability Information – информация о доступности на сетевом уровне.

Route - маршрут

Единица информации, связывающая набор адресатов с атрибутами пути к этим адресатам. Набор адресатов представляет собой системы, чьи адреса IP содержатся в одном префиксе IP, передаваемом в поле NLRI сообщения UPDATE. Путь представляет собой информацию, содержащуюся в поле атрибутов пути того же сообщения UPDATE.

RIB

Routing Information Base – база маршрутной информации.

Unfeasible route – неподходящий маршрут

Анонсированный ранее подходящий маршрут, который утратил доступность.

1.2. Уровни требований

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не следует** (SHALL NOT), **следует** (SHOULD), **не нужно** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с RFC 2119 [RFC2119].

2. Благодарности

Первый вариант этого документа был опубликован в RFC 1267 (октябрь 1991), который подготовили 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 (директор по маршрутизации в IESG[1]) 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.

Некоторые фрагменты этого документа были заимствованы из протокола IDRP [IS10747], являющегося аналогом BGP в OSI. В связи с этим следует отметить работу группы ANSI X3S3.3 под руководством Lyman Chapin и Charles Kunzinger, который также был редактором IDRP в этой группе.

Авторы также выражают свою признательность Benjamin Abarbanel, Enke Chen, Edward Crabbe, Mike Craren, Vincent Gillet, Eric Gray, Jeffrey Haas, Dimitry Haskin, Stephen Kent, John Krawczyk, David LeRoy, Dan Massey, Jonathan Natale, Dan Pei, Mathew Richardson, John Scudder, John Stewart III, Dave Thaler, Paul Traina, Russ White, Curtis Villamizar, Alex Zinin за их комментарии.

Отдельной благодарности заслуживает Andrew Lange за его помощь в подготовке окончательного варианта этого документа.

И, наконец, благодарим всех членов группы IDR за их идеи и поддержку в процессе создания этого документа.

[1] Internet Engineering Steering Group

3. Основы работы протокола

Протокол граничного шлюза BGP является протоколом маршрутизации между автономными системами[1]. Он создан на основе опыта, полученного при разработке протокола EGP (определен в [RFC904]) и его использовании в магистралях NSFNET ([RFC1092] и [RFC1093]). Дополнительную информацию о протоколе BGP можно найти в документах [RFC1772], [RFC1930], [RFC1997] и [RFC2858].

Основной функцией поддерживающей протокол 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 для принятия некоторых решений на уровне политики AS.

В контексте этого документа предполагается, что узел BGP анонсирует своим партнерам только те маршруты, которые он сам использует (т. е., узел BGP говорит, что он “использует” маршрут BGP, если тот является наиболее предпочтительным из BGP-маршрутов и применяется для пересылки пакетов). Рассмотрение прочих случаев не входит в задачи данного документа.

В контексте этого документа термин "IP-адрес" относится к адресам IP версии 4 [RFC791].

Маршрутная информация, передаваемая с использованием BGP, поддерживает только парадигму пересылки на основе адреса получателя, которая предполагает, что маршрутизатор пересылает пакет исключительно на основе адреса получателя, содержащегося в заголовке IP-пакета. Это, в свою очередь, отражает набор правил политики, которые могут быть применены (или не применены) при использовании BGP. Протокол BGP может поддерживать только правила, соответствующие парадигме пересылки по адресу получателя. Отметим, что некоторые правила не могут поддерживаться в рамках парадигмы пересылки на основе адреса получателя и требуют использования иных методов типа маршрутизации, задаваемой отправителем (source routing или explicit routing). Такие правила не могут быть реализованы в рамках BGP. Например, BGP не позволяет одной AS, передающей трафик в соседнюю AS для пересылки тому или иному адресату (достижимому через эту AS, но не относящемуся к ней), предполагать, что этот трафик будет доставляться адресату иным путем, нежели трафик, исходящий из соседней AS (для того же адресата). С другой стороны, BGP может поддерживать любую политику, согласующуюся с парадигмой пересылки на основе адреса получателя.

BGP-4 обеспечивает новые механизмы поддержки бесклассовой междоменной маршрутизации (CIDR) [RFC1518, RFC1519]. Эти механизмы включают поддержку анонсирования набора адресатов в форме префикса IP и позволяют обойтись без концепции «класса» сетей в рамках BGP. BGP-4 также включает механизм объединения маршрутов, включающий объединение путей AS.

В этом документе используется 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 AS 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 через данную AS.

BGP использует в качестве транспортного протокол TCP [RFC793]. Это избавляет от необходимости реализации явного фрагментирования уведомлений, повторной передачи и порядковых номеров. BGP слушает протокол TCP через порт 179. Механизм уведомлений об ошибках, используемый в BGP, предполагает, что TCP поддерживает аккуратное завершение соединений (т. е., все остающиеся данные будут доставлены прежде, чем соединение будет закрыто).

Между парой систем организуется соединение TCP. После этого системы обмениваются между собой стандартными сообщениями для согласования и подтверждения параметров соединения.

Первоначальный поток данных является частью таблицы маршрутизации BGP, которая разрешена политикой экспорта, и называется Adj-Ribs-Out (см. параграф 3.2). В дальнейшем при при изменении таблицы маршрутов передаются нарастающие обновления. BGP не требует периодического обновления таблицы маршрутизации. Чтобы локальные изменения политики могли вступать в силу без сброса соединений, узлу BGP **следует** (a) сохранять текущую версию маршрутов, анонсированных ему всеми партнерами в течение работы соединения или (b) использовать расширение Route Refresh[2] [RFC2918].

Для обеспечения сохранности соединения периодически передаются сообщения KEEPALIVE. Сообщения NOTIFICATION передаются в ответ на ошибки или при возникновении особых условий. При возникновении ошибок в соединении передается сообщение NOTIFICATION и соединение закрывается.

Партнер в другой AS называется внешним партнером (external peer), а партнер из той же AS называется внутренним (internal peer). Для внутренних и внешних соединений BGP обычно используются аббревиатуры IBGP и EBGP, соответственно.

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) данной AS. В целях данного документа принимается допущение, что согласованная картина путей за пределы AS обеспечивается за счет организации каждым узлом BGP внутри данной AS соединений IBGP всеми остальными узлами BGP в этой AS.

Данный документ задает поведение протокола BGP. Это поведение может быть изменено (и изменяется) дополнительными спецификациями. При расширении протокола новое поведение полностью документируется в спецификациях расширения.

3.1. Маршруты – анонсирование и хранение

В целях данного протокола маршрут определяется как единица информации, связывающая набор адресатов с путем к этим адресатам. Набором адресатов являются системы, чьи адреса IP содержатся в одном префиксе IP, указываемом

[1] inter-Autonomous System routing protocol
[2] Route Refresh – обновление маршрута.

полем NLRI[1] сообщения UPDATE, а путь представляет собой информацию, задаваемую в поле атрибутов пути того же сообщения UPDATE.

Маршруты анонсируются между узлами BGP в сообщениях UPDATE. Множество маршрутов с одинаковыми атрибутами пути может быть объединено в одном сообщении UPDATE путем включения множества префиксов в поле NLRI сообщения UPDATE.

Маршруты хранятся в базах маршрутных данных (RIB) Adj-RIBs-In, Loc-RIB и Adj-RIBs-Out, описанных в параграфе 3.2.

Если узел BGP решает анонсировать полученный ранее маршрут, он **может** добавить или изменить атрибуты пути перед анонсированием маршрута партнерам.

Протокол BGP обеспечивает механизмы, позволяющие узлам BGP информировать своих партнеров о том, что анонсированный ранее маршрут перестал быть доступным. Существует три метода, которые данный узел BGP может использовать для указания отзываемых маршрутов.

a) Префикс IP, указывающий адресата ранее анонсированного маршрута, может быть анонсирован в поле WITHDRAWN ROUTES сообщения UPDATE и соответствующий маршрут будет помечен как недоступный для дальнейшего использования.

b) Замена маршрута с сохранением ранее анонсированного значения NLRI.

c) Узел BGP может закрыть соединение, что приведет к удалению всех маршрутов, которые данная пара узлов анонсировала друг другу.

Изменение одного или нескольких атрибутов маршрута сопровождается анонсированием замены. Анонсируемый измененный маршрут содержит иные атрибуты, но имеет такой же префикс, как и ранее анонсированный маршрут.

3.2. База маршрутной информации RIB

База маршрутной информации (RIB[2]) узла BGP состоит из трех отдельных частей.

a) **Adj-RIBs-In** – маршрутные данные, полученные из входящих сообщений UPDATE, которые были приняты от других узлов BGP. Эта база представляет маршруты, которые могут использоваться как входные данные для процесса принятия решения (Decision Process).

b) **Loc-RIB** – локальная маршрутная информация узла BGP, выбранная путем применения локальной политики к маршрутам, содержащимся в Adj-RIBs-In. Эти маршруты будут использоваться локальным узлом BGP. Значения next hop для каждого из этих маршрутов **должны** быть преобразуемыми с помощью таблицы маршрутизации (Routing Table) локального узла BGP.

c) **Adj-RIBs-Out** – информация локального узла BGP, выбранная им для анонсирования своим партнерам. Маршрутные данные из Adj-RIBs-Out будут передаваться от локального узла BGP в сообщениях UPDATE для анонсирования партнерам.

Таким образом, Adj-RIBs-In содержит необработанные маршрутные данные, которые были анонсированы локальному узлу BGP его партнерами, Loc-RIB содержит маршруты, которые выбраны в процессе принятия решения локальным узлом BGP, Adj-RIBs-Out содержит маршруты для анонсирования заданным партнерам (в передаваемых локальным узлом сообщениях UPDATE).

Хотя концептуальная модель различает базы Adj-RIBs-In, Loc-RIB и Adj-RIBs-Out это не требует от реализации протокола поддержки трех отдельных копий маршрутной информации. Выбор способа хранения маршрутных данных (например в 3 копиях или 1 копии с указателями) не задается протоколом.

Маршрутная информация, которую узел BGP использует для пересылки пакетов (или для создания таблицы, используемой для пересылки пакетов) сохраняется в таблице маршрутизации. Таблица маршрутизации включает маршруты в непосредственно подключенные сети, статические маршруты, маршруты, полученные от протоколов IGP, и маршруты, полученные от BGP. Включение того или иного маршрута BGP в таблицу маршрутизации или замена маршрута, полученного из других источников, маршрутом BGP определяются локальной политикой, а не спецификациями данного документа. В дополнение к пересылке пакетов таблица маршрутизации используется для преобразования адресов next-hop, заданных в обновлениях BGP (см. параграф 5.1.3).

4. Формат сообщений

В этой главе описан формат сообщений BGP.

Сообщения BGP передаются через соединения TCP. Обработка сообщений производится только после того, как сообщение будет принято полностью. Максимальный размер сообщения составляет 4096 октетов. Все реализации протокола должны поддерживать сообщения максимального размера. Наименьшее сообщение, которое может быть передано, содержит заголовок BGP (19 октетов) без поля данных.

Многооктетные поля передаются с использованием сетевого порядка байтов.

4.1. Формат заголовка

Каждое сообщение BGP имеет заголовок фиксированного размера, за которым может (но не обязано) следовать поле данных, зависящих от типа сообщения. Схема заголовка показана на рисунке.

Marker - маркер

Это 16-октетное поле включено для обеспечения совместимости[3] и должно содержать 1 в каждом бите.

Length - размер

Это 2-октетное целое число без знака показывает общий размер сообщения, (включая заголовок) в октетах. По этому значению можно найти следующее сообщение (начало поля Marker) в потоке TCP. Значение поля Length **должно** быть не менее 19 и не более 4096. В зависимости от типа сообщения на значение поля размера **могут** накладываться дополнительные ограничения. Заполнение сообщения дополнительными октетами ("padding") после данных не допускается. Следовательно, значение поля Length **должно** быть наименьшим числом, которое позволит включить оставшуюся часть сообщения.

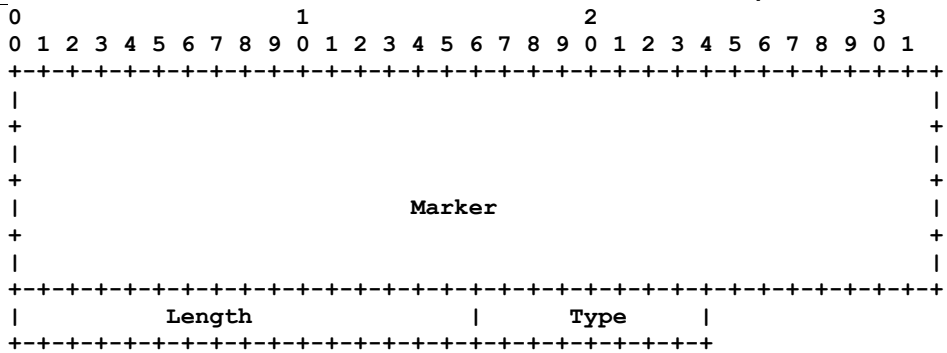
[1]Network Layer Reachability Information – информация о доступности на сетевом уровне.
[2]Routing Information Base
[3]С предыдущими версиями спецификации протокола BGP. *Прим. перев.*

Type - тип

Это 1-октетное целое число без знака содержит код типа сообщения. В данном документе определяются следующие коды типов сообщений:

1 - OPEN
2 - UPDATE
3 - NOTIFICATION
4 - KEEPALIVE

В [RFC2918] определен один дополнительный код типа[1].



4.2. Формат сообщения OPEN

После организации соединения TCP первое сообщение от каждой из сторон соединения имеет тип OPEN. После восприятия сообщения OPEN узел BGP возвращает подтверждающее сообщение KEEPALIVE.

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

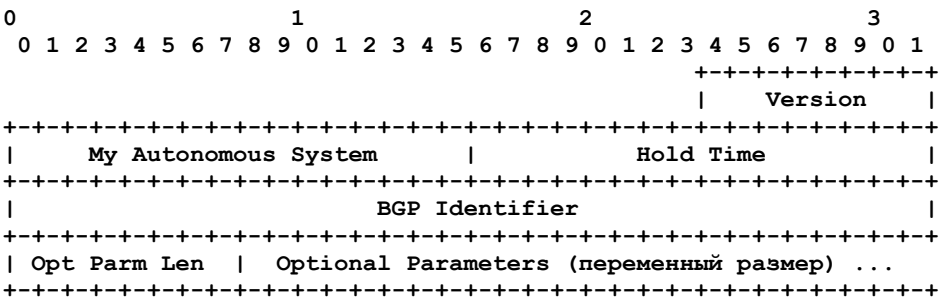
Version - версия

1-октетное целое число без знака, показывающую номер версии протокола. Текущая версия BGP имеет номер 4.

My Autonomous System – моя АС

Это 2-октетное целое число без знака показывает номер AS отправителя сообщения[2].

Hold Time – время удержания



Это 2-октетное целое число без знака показывает число секунд, которое отправитель предлагает установить для таймера удержания (Hold Timer). При получении сообщения OPEN узел BGP **должен** рассчитать значение Hold Timer, используя меньшее из значений Hold Time в локальной конфигурации и принятом сообщении OPEN. Значение Hold Time **должно** быть нулевым или не менее 3 секунд. Реализация **может** отвергать соединения по значению поля Hold Time. Рассчитанное значение показывает максимальное время (в секундах), которое может проходить между получением от партнера сообщений KEEPALIVE и/или UPDATE.

BGP Identifier – идентификатор BGP

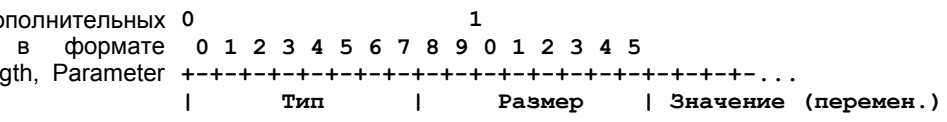
Это 4-октетное целое число без знака показывает идентификатор BGP отправителя сообщения. Узел BGP устанавливает в качестве идентификатора BGP адрес IP, присвоенный этому узлу BGP. Значение идентификатора BGP определяется при старте узла и совпадает для всех локальных интерфейсов и самого узла BGP.

Optional Parameters Length – размер дополнительных параметров

Это 1-октетное целое число без знака показывает общий размер поля Optional Parameters в октетах. Нулевое значение этого поля говорит об отсутствии дополнительных параметров.

Optional Parameters – дополнительные параметры

Это поле содержит список дополнительных параметров, представленных в формате <Parameter Type, Parameter Length, Parameter Value>.



Однооктетное поле типа (Parameter Type) обеспечивает однозначную идентификацию параметра. Однооктетное поле размера (Parameter Length) показывает размер поля Parameter Value в октетах. Значение параметра (Parameter Value) имеет переменный размер и интерпретируется в соответствии с типом параметра (поле Parameter Type).

В [RFC3392] определен дополнительный параметр Capabilities (возможности).

Минимальный размер сообщения OPEN составляет 29 октетов (с учетом заголовка).

4.3. Формат сообщения UPDATE

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

Сообщение UPDATE служит для анонсирования доступных маршрутов с общими атрибутами пути узлу-партнеру 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, а также другие поля, показанные на рисунке (отметим, что некоторые из этих полей являются необязательными).

Withdrawn Routes Length – размер аннулируемых маршрутов

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

[1] Значение поля Type=5 используется для сообщений Route-Refresh, добавленных в RFC 2918. *Прим. перев.*

[2] Если в AS используется 4-октетный номер, данное поле содержит зарезервированное значение 23456, определенное в RFC 4893. *Прим. перев.*

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

```
+-----------------------------------------------------+
|   Withdrawn Routes Length (2 октета)                |
+-----------------------------------------------------+
|   Withdrawn Routes (variable)                       |
+-----------------------------------------------------+
|   Total Path Attribute Length (2 октета)            |
+-----------------------------------------------------+
|   Path Attributes (перемен.)                        |
+-----------------------------------------------------+
|   Network Layer Reachability Information (перемен.) |
+-----------------------------------------------------+
```

Withdrawn Routes – отзываемые маршруты

Это поле переменной длины содержит список префиксов IP-адресов для маршрутов, которые отзываются. Каждый префикс представляется парой <length, prefix> в формате, показанном на рисунке справа.

Ниже описано назначение полей.

a) Length - размер

Поле Length показывает размер адресного префикса IP в битах. Нулевое значение размера указывает на префикс, которому соответствуют все адреса IP (сам префикс содержит 0 октетов).

b) Prefix - префикс

Поле Prefix содержит префикс адресов IP, за которым следует минимально возможное количество битов заполнения, служащих для выравнивания по границе октета. Отметим, что значения битов заполнения не принимаются во внимание.

Total Path Attribute Length – общий размер атрибутов пути.

Это 2-октетное целое число без знака показывает общий размер поля Path Attributes в октетах. Данное значение позволяет определить размер поля Network Layer Reachability Information, как описано ниже.

Нулевое значение поля говорит об отсутствии полей Network Layer Reachability Information и Path Attribute в данном сообщении UPDATE.

Path Attributes – атрибуты пути

Последовательность переменной длины с атрибутами пути присутствует в каждом сообщении UPDATE за исключением тех сообщений, которые служат только для отзыва маршрутов. Каждый атрибут пути представляется триплетом <attribute type, attribute length, attribute value> переменной длины.

Поле типа (Attribute Type) является двухоктетным и состоит из октета флагов (Attribute Flags), за которым следует октет кода 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 (бит 0) октета Attribute Flags является флагом Optional и показывает относится данный атрибут к числу дополнительных (1) или общеизвестных (0).

Следующий по старшинству бит (бит 1) октета Attribute Flags является флагом транзитивности (Transitive), который определяет является атрибут транзитивным[1] (1) или нетранзитивным (0).

Для общеизвестных (well-known) атрибутов флаг Transitive **должен** устанавливаться в 1 (см. обсуждение транзитивных атрибутов в разделе 5).

Следующий бит (бит 2) октета Attribute Flags является флагом Partial и показывает, является информация, содержащаяся в дополнительном транзитивном атрибуте частичной (1) или полной (0). Для общеизвестных атрибутов и дополнительных непереходных атрибутов флаг Partial **должен** иметь значение 0.

Четвертый по старшинству бит (бит 3) октета Attribute Flags является флагом расширенного размера (Extended Length) и определяет размер поля Attribute Length - 1 октет (0) или 2 октета (1).

Четыре младших бита октета Attribute Flags не используются. Отправитель **должен** устанавливать для них нулевые значения, а получатель **должен** игнорировать эти биты.

Октет Attribute Type Code содержит код типа атрибута. Определенные в настоящий момент коды типов перечислены в разделе 5.

Если бит Extended Length октета Attribute Flags имеет значение 0, третий октет Path Attribute содержит значение размера данных атрибута в октетах. При значении бита Extended Length = 1 третий и четвертый октеты атрибута пути содержат размер данных атрибута в октетах.

Остальные октеты поля Path Attribute представляют собой значение атрибута и интерпретируются в соответствии со значениями октетов Attribute Flags и Attribute Type Code. Коды поддерживаемых типов (Attribute Type Code) и значения их атрибутов описаны ниже.

a) **ORIGIN** (тип 1):

Атрибут ORIGIN относится к числу общеизвестных и обязательных, определяя источник маршрутной информации. Октет данных может содержать значения, приведенные в таблице:

Значение	***Описание***
0	IGP – данные NLRI являются внутренними для исходной AS
1	EGP – данные NLRI получены от протокола EGP [RFC904]
2	INCOMPLETE – данные NLRI получены из иных источников.

Использование этого атрибута рассматривается в параграфе 5.1.1.

b) **AS_PATH** (тип 2):

Общеизвестный обязательный атрибут AS_PATH состоит из последовательности сегментов AS path. Каждый сегмент представляется триплетом <path segment type, path segment length, path segment value>.

Тип сегмента пути (path segment type) представляет собой 1-октетное поле, для которого определены следующие значения:

[1] Переходным. *Прим. перев.*

Значение	*Тип сегмента*
1	AS_SET – неупорядоченный набор AS, через которые проходит маршрут из сообщения UPDATE.
2	AS_SEQUENCE - упорядоченный набор AS (последовательность), через которые проходит маршрут из сообщения UPDATE.

Размер сегмента пути (path segment length) представляет собой 1-октетное поле, в котором указывается число номеров AS (не число октетов) в поле path segment value. Поле сегмента пути (path segment value) содержит один или множество номеров AS, каждый из которых представляется 2-октетным[1] полем. Использование этого атрибута рассматривается в параграфе 5.1.2.

c) **NEXT_HOP** (тип 3):

Этот общеизвестный обязательный атрибут определяет (индивидуальный[2]) IP-адрес маршрутизатора, который **следует** использовать в качестве следующего этапа на пути к адресатам, указанным в поле NLRI сообщения UPDATE.

Использование этого атрибута рассматривается в параграфе 5.1.3.

d) **MULTI_EXIT_DISC** (тип 4):

Этот необязательный, непереходный атрибут представляет собой 4-октетное целое число без знака. Значение атрибута **может** использоваться узлом BGP в процессе выбора маршрутов (Decision Process) для разделения множества точек входа в соседнюю АС.

Использование этого атрибута рассматривается в параграфе 5.1.4.

e) **LOCAL_PREF** (тип 5):

Общеизвестный атрибут LOCAL_PREF представляет собой 4-октетное целое число без знака. Узел BGP использует этот атрибут для информирования своих внутренних партнеров, показывая свой уровень предпочтения для анонсируемого маршрута.

Использование этого атрибута рассматривается в параграфе 5.1.5.

f) **ATOMIC_AGGREGATE** (тип 6)

ATOMIC_AGGREGATE является общеизвестным необязательным атрибутом нулевой длины.

Использование этого атрибута рассматривается в параграфе 5.1.6.

g) **AGGREGATOR** (тип 7)

Необязательный транзитивный атрибут AGGREGATOR имеет размер 6 октетов. Этот атрибут содержит номер последней AS, формирующей агрегированный маршрут (2 октета), после которого указан IP-адрес узла BGP, создавшего агрегированный маршрут (4 октета). Для этого поля **следует** устанавливать тот же адрес, который используется для поля BGP Identifier узла, создавшего агрегированный маршрут.

Использование этого атрибута рассматривается в параграфе 5.1.7.

Network Layer Reachability Information

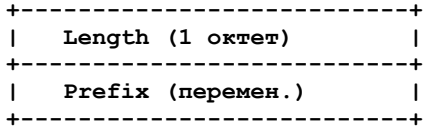
Это поле переменной длины содержит список адресных префиксов IP. Число октетов в поле Network Layer Reachability Information не задается явно, но может быть вычислено по формуле:

Поле Length сообщения UPDATE - 23 - Total Path Attributes Length - Withdrawn Routes Length

Значение поля Length для сообщения UPDATE указано в постоянной части заголовке BGP, Значения полей Total Path Attribute Length и Withdrawn Routes Length указываются в переменной части сообщения UPDATE, а 23 представляет собой суммарный размер постоянного заголовка BGP и полей Total Path Attribute Length, Withdrawn Routes Length.

Информация о доступности представляется в форме одной или множества пар <length, prefix>.

Назначение полей пары описано ниже:



a) Length - размер

Поле Length показывает размер адресного префикса IP в битах. Нулевое значение размера указывает на префикс, которому соответствуют все адреса IP (сам префикс содержит 0 октетов).

b) Prefix - префикс

Поле Prefix содержит префикс адресов IP, за которым следует минимально возможное количество битов заполнения, служащих для выравнивания по границе октета. Отметим, что значения битов заполнения не принимаются во внимание.

Минимальный размер сообщения UPDATE составляет 23 октета; 19 занимает постоянный заголовок BGP, 2 октета – поле Withdrawn Routes Length и 2 октета - Total Path Attribute Length (поля Withdrawn Routes Length и Total Path Attribute Length в этом случае содержат значение 0).

Сообщение UPDATE может анонсировать не более одного набора атрибутов пути, но этому пути может соответствовать множество адресатов, путь к которым описывается общим набором атрибутов. Все атрибуты пути, содержащиеся в данном сообщении UPDATE, применимы к каждому из адресатов, соответствующих значению поля NLRI в сообщении UPDATE.

Сообщение UPDATE может содержать множество аннулируемых маршрутов. Каждый из таких маршрутов идентифицируется своим адресатом (указывается префиксом IP), однозначно определяющим маршрут в контексте соединения между парой узлов BGP, для которого ранее этот маршрут был анонсирован.

[1] В настоящее время могут использоваться также 4-октетные значения номеров AS. Поддержка таких значений определена в RFC 4893. *Прим. перев.*

[2] unicast

Сообщение UPDATE может анонсировать только отзываемые маршруты – в таких случаях сообщение не будет включать атрибутов пути и поля NLRI. Если же сообщение анонсирует только доступные маршруты, в него не требуется включать поле WITHDRAWN ROUTES.

В сообщениях UPDATE **не следует** указывать один и тот же префикс в полях WITHDRAWN ROUTES и NLRI. Однако узел BGP **должен** быть способен к обработке сообщений такого типа. Узлу BGP **следует** трактовать такие сообщения UPDATE, как будто они не содержат адресного префикса в поле WITHDRAWN ROUTES.

4.4. Формат сообщения KEEPALIVE

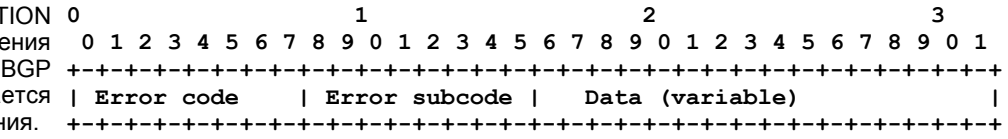
BGP не использует на уровне TCP каких-либо механизмов для проверки доступности других узлов. Вместо этого используются сообщения KEEPALIVE, которыми партнеры обмениваются достаточно часто, чтобы между двумя сообщениями не истекло время, заданное таймером удержания (Hold Timer). Разумным значением максимального 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 постоянному заголовку BGP сообщения NOTIFICATION содержат описанные ниже поля.

Error Code – код ошибки

Это 1-октетное целое число без знака показывает тип сообщения NOTIFICATION. Коды типов перечислены в таблице.

Код	*Символьное имя*	*Описание*
1	Message Header Error – ошибка в заголовке сообщения	параграф 6.1
2	OPEN Message Error – ошибка в сообщении OPEN	параграф 6.2
3	UPDATE Message Error – ошибка в сообщении UPDATE	параграф 6.3
4	Hold Timer Expired – истекло время удержания	параграф 6.5
5	Finite State Machine Error – ошибка машины конечных состояний	параграф 6.6
6	Cease – разрыв соединения	параграф 6.7

Error subcode – субкод ошибки

Это 1-октетное целое число без знака содержит более конкретную информацию о природе ошибки. С каждым кодом ошибки (Error Code) может быть связан один или несколько субкодов (Error Subcode). При отсутствии субкода для ошибки в поле Error Subcode помещается нулевое значение.

Субкоды для 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 для OPEN Message Error

1 – Unsupported Version Number - неподдерживаемый номер версии.

2 - Bad Peer AS – некорректный номер AS у партнера.

3 - Bad BGP Identifier – некорректный идентификатор BGP.

4 - Unsupported Optional Parameter – неподдерживаемый дополнительный параметр.

5 – [Не используется, см. Приложение А].

6 - Unacceptable Hold Time – недопустимое значение времени удержания.

Субкоды для UPDATE Message Error

1 - Malformed Attribute List – некорректно сформированный список атрибутов.

2 - Unrecognized Well-known Attribute – нераспознанный общеизвестный атрибут.

3 - Missing Well-known Attribute – отсутствует обязательный атрибут.

4 - Attribute Flags Error некорректные флаги атрибута.

5 - Attribute Length Error – некорректный размер атрибута.

6 - Invalid ORIGIN Attribute – некорректный атрибут ORIGIN.

7 - [Не используется, см. Приложение А].

8 - Invalid NEXT_HOP Attribute – некорректный атрибут NEXT_HOP.

9 - Optional Attribute Error – ошибка в дополнительном атрибуте.

10 - Invalid Network Field некорректное указание сети.

11 - Malformed AS_PATH – некорректный формат AS_PATH.

Data - данные

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

Отметим, что размер поля Data можно определить на основании значения поля Length в заголовке сообщения по формуле:

```
Length в заголовке сообщения = 21 + размер поля Data
```

Минимальный размер сообщений NOTIFICATION составляет 21 октет (с учетом 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, транзитивные (переходные).
4. Optional non-transitive – дополнительные, непереходные.

Реализации BGP **должны** распознавать все общеизвестные атрибуты. Некоторые из этих атрибутов являются обязательными и **должны** включаться в каждое сообщение UPDATE, содержащее поле NLRI. Остальные атрибуты являются необязательными и **могут** включаться или не включаться в сообщения UPDATE.

После того, как узел BGP обновит значения любого из общеизвестных атрибутов, он **должен** сообщить измененные атрибуты своим партнерам в передаваемых обновлениях.

Кроме общеизвестных 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 значением бита Transitive в октете флагов атрибута. Пути с нераспознанными переходными дополнительными атрибутами **следует** принимать. Если путь с нераспознанными дополнительными переходными атрибутами принят и передается 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 передается 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. Если путь с распознанными транзитивными атрибутами передается другим партнерам BGP и значение бита Partial в поле Attribute Flags уже установлено в 1 какой-либо из предшествующих AS, для текущей AS **недопустимо** сбрасывать этот бит в 0. Нераспознанные нетранзитивные дополнительные атрибуты **должны** игнорироваться без каких-либо действий и передачи другим партнерам BGP.

Новые дополнительные переходные атрибуты **могут** добавляться в конце пути исходным отправителем (originator) или любым узлом BGP на пути. Если эти атрибуты не добавлены исходным отправителем, для бита Partial в октете Attribute Flags устанавливается значение 1. Правила присоединения новых непереходных дополнительных атрибутов зависят 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) узлами BGP 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 неупорядоченных атрибутов пути из сообщения UPDATE.

Один и тот же атрибут (несколько экземпляров одного типа) не может включаться несколько раз в поле Path Attributes сообщения UPDATE.

Обязательные атрибуты **должны** присутствовать в сообщениях, передаваемых в IBGP и EBGP, если сообщение UPDATE включает поле NLRI. Использование дополнительных атрибутов может определяться по собственному усмотрению, требоваться или запрещаться в зависимости от контекста.

Атрибут	***EBGP***	***IBGP***
ORIGIN	обязательно	обязательно
AS_PATH	обязательно	обязательно
NEXT_HOP	обязательно	обязательно
MULTI_EXIT_DISC	по своему усмотрению	по своему усмотрению
LOCAL_PREF	см. параграф 5.1.5	требуется
ATOMIC_AGGREGATE	см. параграфы 5.1.6 и 9.1.4	
AGGREGATOR	по своему усмотрению	по своему усмотрению

5.1. Использование атрибутов пути

Ниже описывается использование всех атрибутов пути BGP.

5.1.1. ORIGIN

Атрибут ORIGIN является общеизвестным и обязательным. Этот атрибут генерируется автономной системой, которая является исходным отправителем маршрутной информации. Другим узлам BGP **не следует** изменять значение этого атрибута.

5.1.2. AS_PATH

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

Когда узел BGP распространяет маршрут, который был получен из сообщения UPDATE от другого узла BGP в сообщении UPDATE, он должен изменить в маршруте атрибут AS_PATH с учетом размещения узла BGP, которому передается маршрут:

a) Когда узел BGP анонсирует маршрут внутреннему партнеру, анонсирующему узлу **не следует** изменять связанный с этим маршрутом атрибут AS_PATH.

b) Когда узел BGP анонсирует маршрут внешнему партнеру, анонсирующий узел обновляет атрибут AS_PATH, как показано ниже:

   1) Если первый сегмент пути в AS_PATH имеет тип AS_SEQUENCE, локальному узлу следует поместить свой номер AS в качестве последнего элемента списка (в крайнюю левую позицию со смещением вправо остальных октетов протокольного сообщения). Если такое включение будет приводить к переполнению сегмента AS_PATH (т. е., число AS превысит 255), **следует** добавить впереди (prepend) новый сегмент, указав в нем свой номер AS.

   2) Если первый сегмент пути в AS_PATH имеет тип AS_SET, локальная система добавляет впереди (prepend) новый сегмент типа AS_SEQUENCE, включая в него свой номер AS.

   3) При пустом AS_PATH локальная система создает сегмент пути типа AS_SEQUENCE, помещает в него свой номер AS и включает этот сегмент в AS_PATH.

Когда узел BGP является источником маршрута:

a) этот узел включает свой номер AS в сегмент пути типа AS_SEQUENCE атрибута AS_PATH всех сообщений UPDATE, передаваемых внешним партнерам. В таких случаях номер AS являющегося источником маршрута узла будет единственным элементом сегмента пути, а данный сегмент будет единственным в атрибуте AS_PATH.

b) этот узел включает пустой атрибут AS_PATH во все сообщения UPDATE, передаваемые внутренним партнерам (пустой атрибут AS_PATH имеет нулевое значение в поле размера).

Всякий раз, когда изменение атрибута AS_PATH связано с включением или добавлением впереди номера AS локальной системы, эта система **может** включать/добавлять впереди более одного экземпляра своего номера AS в атрибут AS_PATH. Этот процесс определяется параметрами локальной конфигурации.

5.1.3. NEXT_HOP

Общеизвестный обязательный атрибут NEXT_HOP определяет IP-адрес маршрутизатора, который **следует** использовать в качестве следующего интервала на пути к адресатам, указанным в сообщении UPDATE. Атрибут NEXT_HOP определяется следующим образом:

1) При передаче сообщения внутреннему партнеру, если маршрут имеет нелокальное происхождение, узлу BGP **не следует** изменять значение NEXT_HOP за исключением тех случаев, когда он явно настроен на анонсирование своего адреса IP в качестве NEXT_HOP. При анонсировании внутренним партнерам маршрутов локального происхождения, узлу BGP **следует** использовать в качестве NEXT_HOP адрес внутреннего интерфейса, через который анонсируемая сеть доступна для принимающего анонс узла. Если маршрут непосредственно соединен с анонсирующим узлом или адрес интерфейса, через который узлу доступна анонсируемая сеть, является адресом внутреннего партнера, узлу BGP **следует** использовать свой адрес IP (адрес интерфейса, через который доступен партнер) в качестве значения атрибута NEXT_HOP.

2) При передаче сообщения внешнему партнеру X, когда тот находится на расстоянии одного интервала IP от данного узла:

   - Если анонсируемый маршрут получен от внутреннего партнера или имеет локальное происхождение, узел BGP может использовать в качестве атрибута NEXT_HOP адрес интерфейса внутреннего партнера (или внутреннего маршрутизатора), через который анонсируемая сеть доступна для данного узла. В этом случае партнер X будет иметь общую подсеть с указанным адресом. Этот случай является вариантом NEXT_HOP из “третьих рук” (third party).
   - В остальных случаях, если анонсируемый маршрут получен от внешнего партнера, узел BGP может использовать в атрибуте NEXT_HOP адрес IP любого смежного маршрутизатора (известный из принятого атрибута NEXT_HOP), который данный узел использует для локального определения маршрута, В таких случаях X имеет с указанным адресом общую подсеть. Этот случай является вариантом NEXT_HOP из “третьих рук”.
   - В противном случае, если внешний партнер, для которого анонсируется маршрут, имеет общую подсеть с одним из интерфейсов анонсирующего узла, последний **может** использовать связанный с таким интерфейсом адрес IP в качестве значения атрибута NEXT_HOP. Этот случай является вариантом NEXT_HOP из “первых рук” (first party).
   - По умолчанию (если не выполняется ни одно из перечисленных выше условий), узлу BGP **следует** использовать в качестве атрибута NEXT_HOP IP-адрес интерфейса, который служит данному узлу для организации соединения BGP с партнером X.

3) При передаче сообщения внешнему партнеру X, находящемуся на расстоянии нескольких интервалов IP от данного узла (multihop EBGP):

   - Узел **может** быть настроен на распространение атрибута NEXT_HOP. В таких случаях при анонсировании полученного от одного из партнеров маршрута узел должен указывать в качестве атрибута NEXT_HOP в анонсируемом маршруте значение NEXT_HOP, полученное в анонсе маршрута от партнера (т. е, узел не изменяет значение NEXT_HOP).
   - По умолчанию узлу BGP **следует** использовать IP-адрес интерфейса, который узел указывает в атрибуте NEXT_HOP для организации соединения BGP с узлом X.

Обычно значение атрибута NEXT_HOP выбирается так, чтобы принимался кратчайший из возможных путей. Узел BGP **должен** обеспечивать возможность запрета анонсирования атрибутов NEXT_HOP, полученных “из третьих рук” для работы в сетях с несовершенными мостами.

Маршрут, порожденный узлом BGP, **не следует** анонсировать партнеру с использованием в качестве атрибута NEXT_HOP адреса этого партнера. Узлу BGP **не следует** устанавливать маршруты со своим адресом в качестве NEXT_HOP.

Атрибут NEXT_HOP используется узлами BGP для определения реального выходного интерфейса и адреса ближайшего маршрутизатора (immediate next-hop address), по которому **следует** пересылать транзитные пакеты для связанных с маршрутом адресатов.

Адрес ближайшего маршрутизатора определяется путем рекурсивного просмотра маршрутов для IP-адреса из атрибута NEXT_HOP, использования содержимого таблицы маршрутизации (Routing Table) и выбора одной записи, если существует множество равноценных путей. Запись таблицы маршрутизации, которая соответствует IP-адресу из атрибута NEXT_HOP, всегда будет задавать выходной интерфейс. Если запись таблицы маршрутизации указывает подключенную подсеть, но не задает адрес next-hop, тогда адрес из атрибута NEXT_HOP **следует** использовать в качестве адреса ближайшего маршрутизатора. Если запись в таблице также содержит адрес next-hop, этот адрес **следует** использовать в качестве адреса ближайшего маршрутизатора для пересылки пакетов.

5.1.4. MULTI_EXIT_DISC

Дополнительный непереходный атрибут MULTI_EXIT_DISC[1] предназначен для использования на внешних (между AS) соединениях при выборе из множества путей в одну смежную AS. Значение атрибута MULTI_EXIT_DISC представляет собой 4-октетное целое число без знака, которое называют метрикой. При прочих равных из нескольких маршрутов **следует** выбирать тот, у которого меньше значение метрики. При получении через EBGP атрибут MULTI_EXIT_DISC **можно** распространять через IBGP другим узлам BGP в данной AS (см. также параграф 9.1.2.2). Атрибут MULTI_EXIT_DISC, полученный из соседней AS, **недопустимо** распространять в другие соседние AS.

Узел BGP **должен** обеспечивать механизм, позволяющий в соответствии с локальной конфигурацией удалять из маршрутов атрибут MULTI_EXIT_DISC. Если узел BGP настроен на удаление атрибута MULTI_EXIT_DISC из маршрутов, такое удаление **должно** выполняться до того, как будет определяться предпочтительный маршрут или происходить выбор маршрута (фазы 1 и 2 Decision Process).

Реализация **может** также в соответствии с локальной конфигурацией изменять значение атрибутов MULTI_EXIT_DISC, полученных через EBGP. Если узел BGP настроен на изменение значений атрибута MULTI_EXIT_DISC, принятых через EBGP, такое изменение **должно** выполняться до того, как будет определяться предпочтительный маршрут или происходить выбор маршрута (фазы 1 и 2 Decision Process). Некоторые ограничения описаны в параграфе 9.1.2.2.

5.1.5. LOCAL_PREF

Атрибут LOCAL_PREF относится к числу общеизвестных необязательных. Это атрибут **следует** включать во все сообщения UPDATE, которые данный узел BGP передает внутренним партнерам (узлам BGP, расположенным в той же автономной системе). Узлу BGP **следует** рассчитать уровень предпочтения для каждого внешнего маршрута на основе локальной политики и включать этот уровень в анонсы для внутренних партнеров. Предпочтение **должно** отдаваться маршрутам с более высоким уровнем. Узел BGP использует уровень предпочтения из LOCAL_PREF, в процессе выбора маршрутов (см. параграф 9.1.1).

Для узлов BGP **недопустимо** включение этого атрибута в сообщения UPDATE, передаваемые внешним партнерам, за исключением случаев использования конфедераций BGP [RFC3065]. Если атрибут содержится в сообщении UPDATE, полученном от внешнего партнера, принимающий узел **должен** игнорировать этот атрибут, за исключением случаев использования конфедераций BGP [RFC3065].

5.1.6. ATOMIC_AGGREGATE

Атрибут ATOMIC_AGGREGATE относится к числу общеизвестных, но не обязательных.

Когда узел BGP объединяет (агрегирует) несколько маршрутов с целью анонсирования отдельному партнеру, значение AS_PATH агрегированного маршрута обычно включает сегмент AS_SET из набора AS, для которых выполняется объединение маршрутов. Во многих случаях администратор сети может определить возможность агрегирования маршрутов без анонсирования AS_SET, чтобы при этом не возникало маршрутных петель.

Если агрегирование не включает по крайней мере некоторые AS из атрибутов AS_PATH объединяемых маршрутов, в создаваемый агрегированный маршрут при анонсировании его партнеру **следует** включать атрибут ATOMIC_AGGREGATE.

Узлу BGP, получившему маршрут с атрибутом ATOMIC_AGGREGATE, **не следует** удалять этот атрибут при распространении маршрута другим узлам.

Для узла BGP, получившего маршрут с атрибутом ATOMIC_AGGREGATE, **недопустимо** указание каких-либо NLRI из этого маршрута как более специфичных (в соответствии с определением параграфа 9.1.4) при анонсировании данного маршрута другим узлам BGP.

Узлу BGP, получившему маршрут с атрибутом ATOMIC_AGGREGATE, следует отдавать себе отчет в том, что актуальный путь к адресату, указанному в NLRI этого маршрута, хотя и не содержит петель, может не совпадать с путем, заданным в атрибуте AS_PATH этого маршрута.

5.1.7. AGGREGATOR

Дополнительный транзитивный атрибут AGGREGATOR **может** включаться в обновления, формируемые при объединении маршрутов (см. параграф 9.2.2.2). Узел BGP, выполняющий агрегирование, **может** добавлять атрибут AGGREGATOR, в который при этом **следует** включать свой номер AS и адрес IP. Адрес IP **следует** указывать тот же, который используется для поля BGP Identifier данного узла.

6. Обработка ошибок BGP

В этой главе рассматриваются действия, предпринимаемые при обнаружении ошибок в процессе обработки сообщений BGP.

При выполнении любого из описанных здесь условий передается сообщение NOTIFICATION с соответствующими значениями кода (Error Code) и субкода (Error Subcode) ошибки, а также полем Data и соединение BGP закрывается

[1]Этот атрибут рассматривается в RFC 4451, перевод которого имеется на сайте www.protocols.ru. *Прим. перев.*

(если явно не указано, что передается сообщение NOTIFICATION, но соединение BGP не закрывается). Если субкод ошибки не указан, **должно** использоваться нулевое значение.

Фраза «соединение BGP разрывается[1]» означает, что закрывается соединение TCP, очищается связанная с соединением BGP база Adj-RIB-In и удаляются все ресурсы, выделенные данному соединению BGP. Записи в базе Loc-RIB, связанные с удаленным партнером, помечаются как некорректные. Локальная система заново рассчитывает наилучшие маршруты для адресатов, маршруты к которым помечены как некорректные. До удаления некорректных маршрутов из таблицы они анонсируются партнерам путем отзыва ставших некорректными маршрутов или задания новых маршрутов взамен некорректных.

Если явно не указано иное, поле Data сообщений NOTIFICATION, передаваемых для индикации ошибок, остается пустым.

6.1. Отработка ошибок в заголовках сообщений

Все ошибки, обнаруживаемые при обработке заголовка сообщения, должны указываться путем передачи сообщений NOTIFICATION с кодом ошибки Message Header Error (ошибка в заголовке сообщения). Поле Error Subcode указывает природу ошибки более точно.

Ожидаемое в заголовке сообщения значение поля Marker состоит только из единиц. Если поле Marker в заголовке сообщения содержит неожиданное значение, возникает ошибка синхронизации и в поле Error Subcode **должно** указываться значение Connection Not Synchronized (соединение не синхронизировано).

Если выполняется хотя бы одно из перечисленных здесь условий:

- поле Length в заголовке сообщения содержит значение меньше 19 или больше 4096;
- значение поля Length в заголовке сообщения OPEN меньше минимального размера сообщения OPEN;
- значение поля Length в заголовке сообщения UPDATE меньше минимального размера сообщения UPDATE;
- значение поля Length в заголовке сообщения KEEPALIVE не равно 19;
- значение поля Length в заголовке сообщения NOTIFICATION меньше минимального размера сообщения NOTIFICATION,

для поля Error Subcode **должно** устанавливаться значение Bad Message Length (некорректный размер сообщения). Поле Data в таких случаях **должно** содержать ошибочное значение поля Length.

Если не распознано поле Type в заголовке сообщения, в поле Error Subcode **должно** помещаться значение Bad Message Type (некорректный тип сообщения). Поле Data в таких случаях **должно** содержать ошибочное значение поля Type.

6.2. Отработка ошибок в сообщениях OPEN

Все ошибки, детектируемые в процессе обработки сообщений OPEN, **должны** указываться сообщениями NOTIFICATION с Error Code = OPEN Message Error. Значение Error Subcode уточняет природу ошибки.

Если версия протокола, указанная в поле Version полученного сообщения OPEN, не поддерживается, **должно** устанавливаться значение Error Subcode = Unsupported Version Number. Поле Data в таких случаях представляет собой 2-октетное целое число без знака, которое показывает (в старшем октете) насколько наибольший номер версии, поддерживаемой локально, меньше номера версии, предложенного удаленным партнером BGP (показан в принятом сообщении OPEN) или (в младшем октете) насколько наименьший локально поддерживаемый номер версии больше предложенного удаленным партнером BGP.

Если поле My Autonomous System в сообщении OPEN содержит неприемлемое значение, в поле Error Subcode **должно** помещаться значение Bad Peer AS. Определение допустимости номеров AS выходит за пределы спецификации данного протокола.

Если значение поля Hold Time в принятом сообщении OPEN неприемлемо, **должно** устанавливаться значение Error Subcode = Unacceptable Hold Time. Реализация **должна** отвергать значения Hold Time в одну или две секунды. Реализация **может** отвергнуть любое предложенное значение Hold Time. Реализация, принявшая значение Hold Time, **должна** использовать согласованное значение параметра Hold Time .

Если поле BGP Identifier в принятом сообщении OPEN синтаксически некорректно, **должно** устанавливаться значение Error Subcode = Bad BGP Identifier. Синтаксическая корректность означает, что поле BGP Identifier содержит допустимый индивидуальный (unicast) IP-адрес хоста.

Если не распознано поле Optional Parameters в принятом сообщении OPEN, **должно** устанавливаться Error Subcode = Unsupported Optional Parameters.

Если один из дополнительных параметров принятого сообщения OPEN распознан, но имеет некорректный формат, **должно** устанавливаться значение Error Subcode = 0.

6.3. Отработка ошибок в сообщениях UPDATE

Все ошибки, детектируемые при обработке сообщений UPDATE, **должны** приводить к генерации сообщения NOTIFICATION с Error Code = UPDATE Message Error. Субкод ошибки уточняет ее природу.

Проверка ошибок в сообщении UPDATE начинается с атрибутов пути. Если значение поля Withdrawn Routes Length или Total Attribute Length слишком велико (т. е., Withdrawn Routes Length + Total Attribute Length + 23 превосходит значение поля Length в заголовке сообщения), в поле Error Subcode **должно** устанавливаться значение Malformed Attribute List.

Если в любом распознанном атрибуте возникает конфликт флагов (Attribute Flags) и типа атрибута (Attribute Type Code), **должно** устанавливаться значение Error Subcode = Attribute Flags Error. В поле Data **должен** включаться связанный с ошибкой атрибут (тип, размер и значение).

Если в любом распознанном атрибуте размер (Attribute Length) конфликтует с ожидаемым (на основе кода типа) значением, **должно** устанавливаться значение Error Subcode = Attribute Length Error. В поле Data **должен** включаться связанный с ошибкой атрибут (тип, размер и значение).

[1] "the BGP connection is closed"

При отсутствии любого из общеизвестных обязательных атрибутов, **должен** устанавливаться субкод Missing Well-known Attribute, а в поле Data **должен** включаться код типа (Attribute Type Code) пропущенного атрибута.

Если не распознан любой из общеизвестных обязательных атрибутов, **должно** устанавливаться значение Error Subcode = Unrecognized Well-known Attribute, а в поле Data **должен** включаться нераспознанный атрибут (тип, размер и значение).

Если атрибут ORIGIN имеет неопределенный тип, в поле Error Subcode **должно** указываться значение Invalid Origin Attribute, а в поле Data **должен** включаться нераспознанный атрибут (тип, размер и значение).

Если поле атрибута NEXT_HOP синтаксически некорректно, для поля Error Subcode **должно** устанавливаться значение Invalid NEXT_HOP Attribute. Поле Data **должно** содержать некорректный атрибут (тип, размер и значение). Синтаксическая корректность означает, что атрибут NEXT_HOP содержит допустимый IP-адрес хоста.

Семантически корректный адрес IP в поле NEXT_HOP **должен** соответствовать двум критериям:

a) **недопустимо** включать в это поле IP-адрес принимающего узла;

b) в случае EBGP, когда отправитель и получатель расположены на расстоянии одного интервала (one IP hop), IP-адрес в поле NEXT_HOP **должен** быть IP-адресом отправителя, использованным для организации соединения BGP, или интерфейс, связанный с адресом из поля NEXT_HOP, **должен** находиться в одной подсети с принимающим узлом BGP.

Если атрибут NEXT_HOP семантически некорректен, **следует** записать информацию об этом в журнальный файл системы, маршрут **следует** игнорировать. В таких случаях передавать сообщение NOTIFICATION и разрывать соединение **не следует.**

Проверяется синтаксическая корректность атрибута AS_PATH. При наличии синтаксических ошибок в пути **должно** устанавливаться значение Error Subcode = Malformed AS_PATH.

Если сообщение UPDATE получено от внешнего партнера, локальная система **может** проверить совпадение расположенного слева (по порядку октетов протокольного сообщения) номера AS в атрибуте AS_PATH с номером автономной системы партнера, передавшего сообщение. Если номера не совпадают, **должно** устанавливаться значение Error Subcode = Malformed AS_PATH.

Если дополнительный атрибут распознан, его значение **должно** быть проверено. При обнаружении ошибки атрибут **должен** быть отброшен и **требуется** установить Error Subcode = Optional Attribute Error. Поле Data в таком случае **должно** содержать связанный с ошибкой атрибут (тип, размер и значение).

Если тот или иной атрибут неоднократно встречается в сообщении UPDATE, в поле Error Subcode **должно** устанавливаться значение Malformed Attribute List.

Проверяется синтаксическая корректность поля NLRI в сообщении UPDATE. При обнаружении ошибки **должно** устанавливаться значение Error Subcode = Invalid Network Field.

Если префикс в поле NLRI семантически некорректен (например, содержит неожиданный групповой адрес IP), информацию об ошибке **следует** записать в локальный журнальный файл, а префикс **следует** игнорировать.

Сообщения UPDATE с корректными атрибутами пути, но без NLRI **следует** трактовать как корректные.

6.4. Отработка ошибок в сообщениях NOTIFICATION

Если узел передает сообщение NOTIFICATION и получатель этого сообщения детектирует в нем ошибку, получатель не может использовать сообщение NOTIFICATION для уведомления своего партнера об ошибке. Все ошибки этого типа (например, нераспознанное значение Error Code или Error Subcode) должны локально протоколироваться с передачей уведомления администратору узла, отправившего ошибочное сообщение. Способы такого протоколирования и уведомления не рассматриваются в данном документе.

6.5. Отработка значений Hold Timer

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

6.6. Обработка ошибок машины конечных состояний

Любая ошибка, обнаруженная машиной конечных состояний (FSM[1]) BGP (например, неожиданное событие), указывается путем передачи сообщения NOTIFICATION с Error Code = Finite State Machine Error.

6.7. Обработка Cease

При отсутствии каких-либо критических ошибок (из числа описанных выше) узел BGP **может** в любой момент закрыть соединение BGP, передав партнеру сообщение NOTIFICATION с Error Code = Cease. Однако такие сообщения **недопустимо** использовать при возникновении какой-либо из перечисленных выше критических ошибок.

Узел BGP **может** обеспечивать возможность вносить задаваемый параметрами локальной конфигурации верхний предел для числа адресных префиксов, принимаемых от соседа. В случае превышения порога, заданного параметрами локальной конфигурации (a) новые префиксы от этого соседа отбрасываются (с сохранением соединения с данным соседом) или (b) закрывается соединение BGP с этим соседом. Если узел BGP принимает решение о разрыве соединения BGP со своим соседом в результате получения от того избыточного числа префиксов, этот узел **должен** передать соседу сообщение NOTIFICATION с Error Code = Cease. Узел **может** также записать информацию об этом в журнальный файл.

6.8. Детектирование конфликтов в соединениях BGP

Если пара узлов BGP пытается одновременно организовать соединение TCP друг с другом, между узлами такой пары могут возникнуть два параллельных соединения. Если IP-адрес отправителя в одном из таких соединений совпадает с IP-адресом получателя в другом соединении и наоборот, возникает конфликт при соединении. При возникновении такого конфликта одно из соединений **должно** быть закрыто.

[1] Finite State Machine – машина конечных состояний.

Выбор одного из пары соединений для закрытия базируется на соглашении об идентификаторах BGP. При возникновении конфликта сравниваются значения BGP Identifier вовлеченных 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 удаленному 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 удаленного узла BGP, указанным в сообщении OPEN. Сравнение производится с преобразованием значений BGP Identifier к принятому для хостов порядку байтов (host byte order) и трактовкой полученных значений как 4-октетных целых чисел без знака.
2) Если значение BGP Identifier для локального узла меньше соответствующего значения для удаленного узла, локальная система закрывает существующее соединение BGP с этим узлом (это соединение находится в состоянии OpenConfirm) и принимает соединение BGP от удаленного партнера.
3) В противном случае локальная система закрывает недавно созданное соединение BGP (связанное с недавно полученным сообщением OPEN) и продолжает использовать существующее соединение с этим партнером (то, которое уже находится в состоянии OpenConfirm).

Если конфигурационные параметры не задают иного, конфликт с существующим соединением BGP, которое находится в состоянии Established, приводит к разрыву недавно созданного соединения.

Отметим, что конфликт соединений не может быть детектирован для состояний Idle, Connect и Active.

Разрыв соединения BGP (в результате процедуры разрешения конфликта) осуществляется путем передачи сообщения NOTIFICATION с Error Code = Cease.

7. Согласование версий BGP

Узлы BGP могут согласовать версию протокола путем 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 партнер, номере версии, переданном партнером в сообщении NOTIFICATION, и номере версии, которую тот поддерживает. Если номера одной или более версий из числа поддерживаемых обоими партнерами 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. Машина конечных состояний BGP

Структуры данных и FSM, описанные в данном документе, являются концептуальными моделями и не реализуются в точном соответствии с приведенными описаниями. Если реализация поддерживает описанную функциональность, она будет демонстрировать соответствующее описанному здесь поведение.

В этой главе описывается работа BGP в терминах машины конечных состояний (FSM). Глава разбита на две части:

1) Описание событий для машины состояний (параграф 8.1)
2) Описание FSM (параграф 8.2)

Обязательными атрибутами каждого соединения являются:

1) State – состояние;
2) ConnectRetryCounter – счетчик числа попыток организации соединения;
3) ConnectRetryTimer – таймер повторов для соединения;
4) ConnectRetryTime – время ожидания для повтора;
5) HoldTimer – таймер удержания;
6) HoldTime – время удержания;
7) KeepaliveTimer – таймер сохранения;
8) KeepaliveTime – время сохранения.

Атрибуты состояния сессии показывают текущее состояние BGP FSM. Счетчик ConnectRetryCounter показывает число попыток узла BGP организовать соединение с партнером.

Обязательные атрибуты, связанные с таймерами, описаны в главе 10. Для каждого таймера существуют значения "timer" и "time" (начальное значение).

Ниже перечислены дополнительные атрибуты сессий. Эти атрибуты могут поддерживаться для соединений или для локальной системы в целом:

1) AcceptConnectionsUnconfiguredPeers
2) AllowAutomaticStart
3) AllowAutomaticStop
4) CollisionDetectEstablishedState
5) DampPeerOscillations
6) DelayOpen
7) DelayOpenTime
8) DelayOpenTimer
9) IdleHoldTime
10) IdleHoldTimer
11) PassiveTcpEstablishment

12) SendNOTIFICATIONwithoutOPEN

13) TrackTcpState

Дополнительные атрибуты сессий определяют различные параметры BGP, оказывающие влияние на смену состояний BGP FSM. Две группы атрибутов, связанных с таймерами, включают:

Группа 1: DelayOpen, DelayOpenTime, DelayOpenTimer

Группа 2: DampPeerOscillations, IdleHoldTime, IdleHoldTimer

Первый параметр (DelayOpen, DampPeerOscillations) является дополнительным атрибутом, который показывает, что функция Timer активна. Значение "Time" указывает начальное состояние таймера (DelayOpenTime, IdleHoldTime). "Timer" задает реальный таймер.

Описание взаимодействия между дополнительными атрибутами и состояниями, передаваемыми FSM, приведено в параграфе 8.1.1. Параграф 8.2.1.3 содержит краткий обзор двух различных типов дополнительных атрибутов (флаги и таймеры).

8.1. События BGP FSM

8.1.1. Дополнительные события, связанные с дополнительными атрибутами сессии

Входной информацией для BGP FSM являются события, которые могут относиться к числу обязательных (mandatory) и необязательных (optional). Некоторые из дополнительных событий связаны с дополнительными атрибутами сессии. Дополнительные атрибуты сессий включают несколько групп функций FSM.

Связи между функциями FSM, событиями и дополнительными атрибутами сессий описаны ниже.

Группа 1: Автоматические административные события (старт/стоп)

Дополнительные атрибуты сессии: AllowAutomaticStart, AllowAutomaticStop, DampPeerOscillations, IdleHoldTime, IdleHoldTimer

Опция 1: AllowAutomaticStart

Описание. Соединение с партнером BGP может быть инициировано или разорвано административными средствами. Такая операция может выполняться вручную с участием оператора или автоматически под управлением встроенной логики реализации BGP. Термин «автоматически» говорит о том, что соединение с партнером BGP организуется тогда, когда логика определяет, что соединение BGP следует перезапустить.

Атрибут AllowAutomaticStart указывает, что данное соединение BGP поддерживает автоматический запуск соединения BGP.

Если реализация BGP поддерживает AllowAutomaticStart, рестарт соединения с партнером может повторяться. Опции DampPeerOscillations, IdleHoldTime, IdleHoldTimer управляют скоростью организации повторных соединений.

Опция DampPeerOscillations определяет использование дополнительной логики для подавления осцилляций BGP в форме последовательности автоматически повторяющихся процедур старта и остановки. Параметр IdleHoldTime задает продолжительность периода сохранения партнером BGP состояния Idle перед тем, как будет возможен новый автоматический рестарт. Таймер IdleHoldTimer управляет сохранением состояния Idle.

Примером логики DampPeerOscillations является рост значения IdleHoldTime в тех случаях, когда BGP-партнер порождает периодические осцилляции (организация/разрыв соединения) в течение некоторого интервала времени. Для включения этой логики партеру достаточно организовать 10 соединений и их разрывов в течение 5 минут. Значение IdleHoldTime будет сменено с нуля на 120 секунд.

Значения: TRUE или FALSE

Опция 2: AllowAutomaticStop

Описание. Этот дополнительный атрибут сессии BGP показывает, что для соединения BGP разрешено «автоматическое» прерывание. Автоматическим называется прерывание сессии под управлением поддерживаемой реализацией логики. Рассмотрение такой логики не входит в задачи данной спецификации.

Значения: TRUE или FALSE

Опция 3: DampPeerOscillations

Описание. Дополнительный атрибут сессии DampPeerOscillations показывает, что соединение BGP использует логику подавления осцилляций в состоянии Idle.

Значения: TRUE или FALSE

Опция 4: IdleHoldTime

Описание. IdleHoldTime принимает значение, установленное для IdleHoldTimer.

Значение: Время в секундах.

Опция 5: IdleHoldTimer

Описание. Таймер IdleHoldTimer служит для контроля осцилляций BGP за счет сохранения партнера BGP в состоянии Idle в течение заданного интервала времени. Событие IdleHoldTimer_Expires описано в параграфе 8.1.3.

Значение: Время в секундах.

Группа 2: Не указанные в конфигурации партнеры

Дополнительные атрибуты сессии: AcceptConnectionsUnconfiguredPeers

Опция 1: AcceptConnectionsUnconfiguredPeers

Описание. Машина состояний BGP FSM может позволять восприятие соединений BGP от неуказанных в конфигурации соседей. Дополнительный атрибут сессии AcceptConnectionsUnconfiguredPeers позволяет FSM поддерживать переходы состояний, которые позволяют реализации принимать или отвергать соединения от таких партнеров.

Атрибут AcceptConnectionsUnconfiguredPeers влияет на безопасность. Детальное описание можно найти в документе “Уязвимости BGP” [RFC4272].

Значения: TRUE или FALSE

Группа 3: Обработка TCP

Дополнительные атрибуты сессии: PassiveTcpEstablishment, TrackTcpState

Опция 1: PassiveTcpEstablishment

Описание. Эта опция показывает, что BGP FSM будет пассивно ожидать вызова от удаленного партнера BGP для организации соединения TCP.

Значения: TRUE или FALSE

Опция 2: TrackTcpState

Описание. BGP FSM обычно отслеживает конечный результат попытки организации соединения TCP, а не отдельные сообщения TCP. Опционально BGP FSM может поддерживать дополнительное взаимодействие с системой согласования параметров соединений TCP. Учет событий TCP может увеличить объем записей в журнальных файлах и число смен состояний BGP FSM.

Значения: TRUE или FALSE

Группа 4: Обработка сообщений BGP

Дополнительные атрибуты сессии: DelayOpen, DelayOpenTime, DelayOpenTimer, SendNOTIFICATIONwithoutOPEN, CollisionDetectEstablishedState

Опция 1: DelayOpen

Описание. Дополнительный атрибут сессии DelayOpen позволяет реализации настроить задержку передачи сообщения OPEN на заданное время. Такая задержка позволяет удаленному партнеру BGP первым отправить сообщение OPEN.

Значения: TRUE или FALSE

Опция 2: DelayOpenTime

Описание. DelayOpenTime – начальное значение таймера DelayOpenTimer.

Значение: Время в секундах.

Опция 3: DelayOpenTimer

Описание. Дополнительный атрибут сессии DelayOpenTimer используется для задержки передачи сообщения OPEN в данном соединении. Событие DelayOpenTimer_Expires (Событие 12) описано в параграфе 8.1.3.

Значение: Время в секундах.

Опция 4: SendNOTIFICATIONwithoutOPEN

Описание. SendNOTIFICATIONwithoutOPEN позволяет партнеру передать сообщение NOTIFICATION без отправки сначала сообщения OPEN. Без этого дополнительного атрибута соединение BGP предполагает, что сообщение OPEN должно быть отправлено партнером до того, как ему будет передаваться сообщение NOTIFICATION.

Значения: TRUE или FALSE

Опция 5: CollisionDetectEstablishedState

Описание. Обычно Detect Collision (см. параграф 6.8) игнорируется для состояния Established. Этот дополнительный атрибут сессии показывает, что данное соединение BGP обрабатывает конфликты и в состоянии Established.

Значения: TRUE или FALSE

Примечание: Дополнительные сеансовые атрибуты проясняют описание BGP FSM для имеющихся возможностей реализации BGP. Эти атрибуты могут быть предопределенными для реализации и недоступными через интерфейс управления, поддерживаемый реализацией. Если поддерживается новая (2 и выше) версия BGP MIB, эти поля будут доступны через интерфейс управления.

8.1.2. События административного плана

К числу административных относятся те события, при которых операторский интерфейс машины политики BGP[1] сигнализирует машине конечных состояний BGP о необходимости запуска или остановки машины состояний BGP. Базовые средства индикации запуска и остановки дополняются необязательными атрибутами соединения, которые передают сигналы о некоторых типах запуска и остановки BGP FSM. Примером такой комбинации может служить Событие 5 - AutomaticStart_with_PassiveTcpEstablishment. С помощью такого события реализация BGP сигнализирует BGP FSM об использовании Automatic Start с опцией для применения процедуры Passive TCP Establishment. В свою очередь Passive TCP establishment сигнализирует, что BGP FSM будет ждать вызова удаленной стороны для организации соединения TCP.

Отметим, что только Событие 1 (ManualStart) и Событие 2 (ManualStop) относятся к числу обязательных административных событий. Все остальные события административного типа (События 3–8) являются дополнительными. Каждое из описанных ниже событий имеет номер, определение, статус (обязательное или дополнительное), а также дополнительные атрибуты сессии, которые **следует** устанавливать на каждой стадии. При генерации Событий 1 - 8 для BGP FSM проверяются условия, заданные в поле "**Статус дополнительных атрибутов**". Если любое из этих условий не выполняется, локальной системе следует записать в журнальный файл сведения об ошибке FSM.

[1]BGP Policy engine

В некоторых реализациях установка дополнительных атрибутов сессии может быть неявной и, следовательно, эти атрибуты не могут явно устанавливаться оператором. В параграфе 8.2.1.5 описаны такие неявные установки дополнительных сеансовых атрибутов. Описанные ниже административные события также могут быть в некоторых реализациях неявными и недоступными для оператора.

Событие 1: ManualStart

Определение: администратор локальной системы вручную инициирует соединение с партнером.

Статус: обязательный

Статус дополнительных атрибутов: для атрибута PassiveTcpEstablishment **следует** установить значение FALSE.

Событие 2: ManualStop

Определение: администратор локальной системы вручную останавливает соединение с партнером.

Статус: обязательный

Статус дополнительных атрибутов: взаимодействие с дополнительными атрибутами отсутствует.

Событие 3: AutomaticStart

Определение: локальная система автоматически организует соединение BGP.

Статус: дополнительный, зависит от локальной системы.

Статус дополнительных атрибутов:

1) для атрибута AllowAutomaticStart **следует** установить значение TRUE, если происходит данное событие;
2) если поддерживается дополнительный атрибут сессии PassiveTcpEstablishment, для него **следует** установить значение FALSE;
3) если поддерживается DampPeerOscillations, **следует** установить значение FALSE, когда произойдет данное событие.

Событие 4: ManualStart_with_PassiveTcpEstablishment

Определение: локальный администратор вручную инициирует соединение с партнером при включенном режиме PassiveTcpEstablishment; дополнительный сеансовый атрибут PassiveTcpEstablishment показывает, что будут прослушиваться вызовы партнера прежде, чем соединение будет организовано.

Статус: дополнительный, зависит от локальной системы.

Статус дополнительных атрибутов:

1) для атрибута PassiveTcpEstablishment **следует** установить значение TRUE, если это событие происходит;
2) после завершения события для атрибута DampPeerOscillations **следует** установить значение FALSE.

Событие 5: AutomaticStart_with_PassiveTcpEstablishment

Определение: локальная система автоматически инициирует соединение BGP при включенном режиме PassiveTcpEstablishment; дополнительный сеансовый атрибут PassiveTcpEstablishment показывает, что будут прослушиваться вызовы партнера прежде, чем соединение будет организовано.

Статус: дополнительный, зависит от локальной системы.

Статус дополнительных атрибутов:

1) для атрибута AllowAutomaticStart **следует** установить значение TRUE;
2) для атрибута PassiveTcpEstablishment **следует** установить значение TRUE;
3) если поддерживается атрибут DampPeerOscillations, для него **следует** установить значение FALSE.

Событие 6: AutomaticStart_with_DampPeerOscillations

Определение:локальная система автоматически инициирует соединение BGP при включенном режиме подавления осцилляций; конкретный метод подавления осцилляций определяется реализацией и его рассмотрение выходит за пределы данной спецификации.

Статус: дополнительный, зависит от локальной системы.

Статус дополнительных атрибутов:

1) для атрибута AllowAutomaticStart **следует** установить значение TRUE;
2) для атрибута DampPeerOscillations **следует** установить значение TRUE;
3) для атрибута PassiveTcpEstablishment **следует** установить значение TRUE.

Событие 7: AutomaticStart_with_DampPeerOscillations_and_PassiveTcpEstablishment

Определение: локальная система автоматически инициирует соединение BGP при включенном режиме подавления осцилляций и PassiveTcpEstablishment; конкретный метод подавления осцилляций определяется реализацией и его рассмотрение выходит за пределы данной спецификации.

Статус: дополнительный, зависит от локальной системы.

Статус дополнительных атрибутов:

1) для атрибута AllowAutomaticStart **следует** установить значение TRUE;
2) для атрибута DampPeerOscillations **следует** установить значение TRUE;
3) для атрибута PassiveTcpEstablishment **следует** установить значение TRUE.

Событие 8: AutomaticStop

Определение: локальная система автоматически останавливает соединение BGP; примером автоматической остановки может служить избыточное число префиксов от данного партнера и автоматический разрыв локальной системой соединения с этим партнером.

Статус: дополнительный, зависит от локальной системы.

Статус дополнительных атрибутов: для атрибута AllowAutomaticStop **следует** установить значение TRUE.

8.1.3. События, связанные с таймерами

Событие 9: ConnectRetryTimer_Expires

Определение: событие генерируется при завершении отсчета таймера ConnectRetryTimer.

Статус: обязательное.

Событие 10: HoldTimer_Expires

Определение: событие генерируется при завершении отсчета таймера HoldTimer.

Статус: обязательное.

Событие 11: KeepaliveTimer_Expires

Определение: событие генерируется при завершении отсчета таймера KeepaliveTimer.

Статус: обязательное.

Событие 12: DelayOpenTimer_Expires

Определение: событие генерируется при завершении отсчета таймера DelayOpenTimer.

Статус: дополнительное.

Статус дополнительных атрибутов: если произошло данное событие

1) для атрибута DelayOpen **следует** установить значение TRUE;
2) **следует** поддерживать атрибут DelayOpenTime;
3) **следует** поддерживать атрибут DelayOpenTimer.

Событие 13: IdleHoldTimer_Expires

Определение: это событие генерируется при завершении отсчета таймера IdleHoldTimer, которое показывает, что для соединения BGP завершился период ожидания (back-off), служащий для предотвращения осцилляции BGP.

Таймер IdleHoldTimer используется только в тех случаях, когда разрешено постоянное использование функции подавления осцилляций путем установки DampPeerOscillations = TRUE.

Реализации, не поддерживающие функцию постоянного подавления осцилляций, могут не поддерживать таймер IdleHoldTimer.

Статус: дополнительное.

Статус дополнительных атрибутов: если происходит данное событие:

1) для атрибута DampPeerOscillations **слебует** установить значение TRUE;
2) **следует** дождаться завершения отсчета таймера IdleHoldTimer.

8.1.4. События, связанные с соединениями TCP

Событие 14: TcpConnection_Valid

Определение: событие, показывающее прием локальной системой запроса на соединение TCP с корректным адресом и номером порта TCP для отправителя и получателя; принятие решение о корректности IP-адресов отправителя и получателя является прерогативой реализации.

В качестве порта получателя BGP **следует** использовать значение 179, заданное IANA.

Запросы на организацию соединений TCP фиксируются локальной системой при получении пакетов TCP SYN.

Статус: дополнительное.

Статус дополнительных атрибутов: для атрибута TrackTcpState **следует** установить значение TRUE, если происходит данное событие.

Событие 15: Tcp_CR_Invalid

Определение: событие, показывающее получение локальной системой запроса на организацию соединения TCP с некорректным значением адреса или номера порта для отправителя или получателя.

В качестве порта получателя BGP **следует** использовать значение 179, заданное IANA.

Запросы на организацию соединений TCP фиксируются локальной системой при получении пакетов TCP SYN.

Статус: дополнительное.

Статус дополнительных атрибутов: для атрибута TrackTcpState **следует** установить значение TRUE, если происходит данное событие.

Событие 16: Tcp_CR_Acked

Определение: событие, показывающее, что локальная система запросила организацию соединения TCP с удаленным партнером.

Локальная система передала пакет TCP SYN, приняла отклик TCP SYN/ACK и передала подтверждение TCP ACK.

Статус: обязательное.

Событие 17: TcpConnectionConfirmed

Определение: событие, показывающее, что локальная система получила от удаленного узла подтверждение организации соединения TCP.

Модуль TCP удаленного партнера передал пакет TCP SYN; локальный модуль передал в ответ SYN, ACK и получил завершающее подтверждение ACK.

Статус: обязательное.

Событие 18: TcpConnectionFails

Определение: событие, показывающее, что локальная система получила информацию об отказе при попытке организации соединения TCP.

Модуль TCP удаленного партнера BGP мог передать пакет FIN, на который локальный узел ответил пакетом FIN-ACK. Другим вариантом является детектирование тайм-аута для соединения TCP и прекращения попытки организации соединения.

Статус: обязательное.

8.1.5. События, связанные с сообщениями BGP

Событие 19: BGPOpen

Определение: это событие генерируется при получении корректного сообщения OPEN.

Статус: обязательное.

Статус дополнительных атрибутов:

1) для атрибута DelayOpen **следует** установить значение FALSE;
2) таймер DelayOpenTimer **следует** выключить.

Событие 20: BGPOpen with DelayOpenTimer running

Определение: это событие генерируется при получении корректного сообщения OPEN для партнера, который уже имеет организованное транспортное соединение и в настоящее время задерживает передачу сообщения BGP OPEN.

Статус: дополнительное.

Статус дополнительных атрибутов:

3) для атрибута DelayOpen **следует** установить значение FALSE;
4) таймер DelayOpenTimer **следует** включить.

Событие 21: BGPHeaderErr

Определение: это событие генерируется при получении сообщения BGP с некорректным заголовком.

Статус: обязательное.

Событие 22: BGPOpenMsgErr

Определение: это событие генерируется при получении сообщения OPEN, содержащего ошибки.

Статус: обязательное.

Событие 23: OpenCollisionDump

Определение: это событие генерируется административным путем при детектировании конфликта соединений в процесс обработки входящего сообщения OPEN, если данное соединение планируется разорвать; описание детектирования конфликтов приведено в параграфе 6.8.

Событие 23 является административным действием, генерируемым логикой реализации, принимающей решение о сбросе соединения в соответствии с правилами параграфа 6.8. Это событие может происходить, если FSM реализована как две связанных машины состояний.

Статус: дополнительное.

Статус дополнительных атрибутов: если машина состояний обрабатывает это событие из состояния Established, для дополнительного атрибута CollisionDetectEstablishedState **следует** установить значение TRUE.

Примечание: событие OpenCollisionDump может происходить в состояниях Idle, Connect, Active, OpenSent и OpenConfirm без установки каких-либо дополнительных атрибутов.

Событие 24: NotifMsgVerErr

Определение: это событие генерируется при получении сообщения NOTIFICATION с кодом ошибки несоответствия версий.

Статус: обязательное.

Событие 25: NotifMsg

Определение: это событие генерируется при получении сообщения NOTIFICATION с кодом ошибки, отличным от несовпадения версий.

Статус: обязательное.

Событие 26: KeepAliveMsg

Определение: это событие генерируется при получении сообщения KEEPALIVE.

Статус: обязательное.

Событие 27: UpdateMsg

Определение: это событие генерируется при получении корректного сообщения UPDATE.

Статус: обязательное.

Событие 28: UpdateMsgErr

Определение: это событие генерируется при получении некорректного сообщения UPDATE

Статус: обязательное.

8.2. Описание FSM

8.2.1. Определение FSM

Реализация BGP **должна** поддерживать отдельную FSM для каждого включенного в конфигурацию партнера. Каждый узел BGP, включенный в потенциальное соединение, будет пытаться связаться с партнером, если для данного узла не

задано сохранение состояния Idle или пассивный режим. В последующем обсуждении активная или подключающаяся сторона соединения TCP (та сторона, с которой был передан первый пакет TCP SYN) называется исходящей (outgoing). Пассивная или ожидающая сторона (отправитель первого пакета SYN/ACK) будет называться входящей. Дополнительные разъяснения терминов «активный» и «пассивный» приведены ниже в параграфе 8.2.1.1.

Реализация BGP **должна** подключиться к порту TCP с номером 179 и прослушивать его с целью приема входящих вызовов в дополнение к своим попыткам организовать соединение с партнером. Для каждого входящего соединения **должен** создаваться экземпляр машины состояний. Существует период, в течение которого соединение с партнером на другой стороне уже организовано, но его идентификатор BGP еще не известен. В течение этого периода могут одновременно существовать входящее и исходящее соединение для одной пары партнеров. Такая ситуация называется конфликтом при соединении (см. параграф 6.8).

Реализация BGP будет иметь не более одной машины FSM для каждого указанного в конфигурации партнера и одну FSM для каждого входящего соединения TCP, в котором партнер еще не идентифицирован. Каждый экземпляр FSM соответствует одному соединению TCP.

Между парой партнеров может существовать несколько соединений, если в них используются различные пары адресов IP. Такая ситуация называется «множественным партнерством» (multiple "configured peerings").

8.2.1.1. Термины "активный" и "пассивный"

Термины "активный" и "пассивный" присутствуют в сленге операторов Internet уже почти десятилетие и оказались весьма полезными. Эти термины в контексте соединений TCP или соединений с партнерами имеют специфическое толкование. В любом соединении TCP может быть только одна активная и одна пассивная сторона (в соответствии с приведенным выше определением и описанными ниже состояниями FSM). Когда узел BGP настроен как активный, он может располагаться как на активной, так и на пассивной стороне соединения, которое будет организовано в результате. После завершения процесса организации соединения TCP уже не имеет значения, какая из сторон была активной, а какая пассивной на этапе организации соединения. Единственное различие заключается в том, какая из сторон будет использовать порт TCP с номером 179.

8.2.1.2. FSM и детектирование конфликтов

Существует одна машина FSM на каждое соединение BGP. При возникновении конфликта соединений до того, как партнер будет полностью идентифицирован, может существовать два соединения с одним партнером. После разрешения конфликта (см. параграф 6.8) FSM разорванного соединения **следует** освободить (отключить).

8.2.1.3. FSM и дополнительные атрибуты сессий

Дополнительные атрибуты сессий действуют как флаги (TRUE или FALSE) или дополнительные таймеры. Для атрибутов, являющихся флагами, должно поддерживаться соответствующее действие BGP FSM, если для флага может быть установлено значение TRUE. Например, если в реализации BGP могут быть установлены опции AutoStart и PassiveTcpEstablishment, должны поддерживаться События 3, 4 и 5. Если дополнительный атрибут сессии не может иметь значения TRUE, соответствующие события не поддерживаются.

Каждый из дополнительный таймеров (DelayOpenTimer и IdleHoldTimer) имеет группу атрибутов, включающую:

- флаг индикации поддержки;
- значение времени (Time) для таймера;
- таймер.

Формат дополнительный таймеров показан ниже:

DelayOpenTimer: DelayOpen, DelayOpenTime, DelayOpenTimer

IdleHoldTimer: DampPeerOscillations, IdleHoldTime, IdleHoldTimer

Если флаг индикации поддержки для дополнительного таймера (DelayOpen или DampPeerOscillations) не может иметь значение TRUE, таймеры и события, поддерживающие данную опцию, не поддерживаются.

8.2.1.4. Номера событий FSM

Номера событий (1-28) используются при описании машины состояний. Реализации могут использовать эти номера для систем сетевого управления. Точная форма FSM или событий FSM зависит от реализации.

8.2.1.5. Действия FSM, зависящие от реализации

В некоторых случаях BGP FSM указывает, что будет выполняться инициализация BGP или удаление ресурсов BGP. Инициализация BGP FSM и связанных с машиной ресурсов зависит от связанной с политикой части реализации BGP. Детальное рассмотрение этих действий выходит за пределы описания FSM.

8.2.2. Машина конечных состояний

Состояние Idle

Изначально FSM узла BGP находится в состоянии Idle (далее машина конечных состояний узла BGP будет обозначаться для краткости BGP FSM).

В этом состоянии BGP FSM отвергает все входящие соединения BGP для данного узла. Никаких ресурсов не выделено. В ответ на событие ManualStart (1) или AutomaticStart (3) локальная система будет:

- инициализировать все ресурсы BGP для соединения с партнером;
- устанавливать ConnectRetryCounter = 0;
- запускать таймер ConnectRetryTimer с начальным значением;
- инициировать соединение TCP с другим узлом BGP;
- прослушивать соединения, инициированные удаленными узлами BGP;
- переходить в состояние Connect.

События ManualStop (Событие 2) и AutomaticStop (Событие 8) игнорируются в состоянии Idle.

В ответ на событие ManualStart_with_PassiveTcpEstablishment (4) или AutomaticStart_with_PassiveTcpEstablishment (5) локальная система будет:

- инициализировать все ресурсы BGP;
- устанавливать ConnectRetryCounter = 0;
- запускать таймер ConnectRetryTimer с начальным значением;
- прослушивать соединения, инициированные удаленными узлами BGP;
- переходить в состояние Active.

Точное значение ConnectRetryTimer определяется локально, но его **следует** делать достаточно большим для того, чтобы прошла инициализация TCP.

Если атрибут DampPeerOscillations имеет значение TRUE, в состоянии Idle возможны три события:

- AutomaticStart_with_DampPeerOscillations (Событие 6),
- AutomaticStart_with_DampPeerOscillations_and_PassiveTcpEstablishment (Событие 7),
- IdleHoldTimer_Expires (Событие 13).

Эти события будут использоваться локальной системой для предотвращения осцилляций. Метод предотвращения постоянных осцилляций выходит за пределы данного документа.

Любое другое событие (9-12, 15-28) в состоянии Idle не приводит к смене состояния локальной системы.

Состояние Connect

В этом состоянии BGP FSM ожидает завершения процесса организации соединения TCP. Стартовые события (1, 3-7) игнорируются в состоянии Connect. В ответ на событие ManualStop (Событие 2) локальная система будет:

- сбрасывать соединение TCP;
- освобождать все ресурсы BGP;
- устанавливать ConnectRetryCounter = 0;
- останавливать таймер ConnectRetryTimer и устанавливать для него нулевое значение;
- переходить в состояние Idle.

В ответ на событие ConnectRetryTimer_Expires (9) локальная система будет:

- сбрасывать соединение TCP;
- заново запускать таймер ConnectRetryTimer;
- останавливать таймер DelayOpenTimer и сбрасывать его значение в 0;
- инициировать соединение TCP с другим узлом BGP;
- продолжать прослушивание порта для определения входящих вызовов от других узлов BGP;
- сохранять состояние Connect.

Если происходит событие DelayOpenTimer_Expires (12) в состоянии Connect, локальная система будет:

- передавать партнеру сообщение OPEN;
- устанавливать большое значение для таймера удержания HoldTimer;
- переходить в состояние OpenSent.

Если BGP FSM получает информацию о событии TcpConnection_Valid (14), обрабатывается соединение TCP и сохраняется состояние Connect.

При получении BGP FSM информации о событии Tcp_CR_Invalid (15) локальная система отвергнет соединение TCP и сохранит состояние Connect.

При успешной организации соединения TCP (Событие 16 или 17) локальная система будет сначала проверять атрибут DelayOpen. Если этот атрибут имеет значение TRUE, локальная система будет:

- останавливать таймер ConnectRetryTimer (если тот включен) и сбрасывать его в 0;
- устанавливать начальное значение для таймера DelayOpenTimer;
- сохранять состояние Connect.

Если атрибут DelayOpen имеет значение FALSE, локальная система будет:

- останавливать таймер ConnectRetryTimer (если тот включен) и сбрасывать его в 0;
- завершать инициализацию BGP;
- передавать партнеру сообщение OPEN;
- устанавливать большое значение для таймера удержания HoldTimer;
- переходить в состояние OpenSent.

Предлагается использовать для HoldTimer значение 4 минуты.

При отказе в организации соединения TCP (Событие 18) локальная система проверяет DelayOpenTimer. Если этот таймер запущен, локальная система будет:

- заново запускать таймер ConnectRetryTimer;
- останавливать таймер DelayOpenTimer и сбрасывать его значение в 0;
- продолжать прослушивание порта для приема входящих вызовов от других узлов BGP;
- переходить в состояние Active.

Если таймер DelayOpenTimer не запущен, локальная система будет:

- заново запускать таймер ConnectRetryTimer;
- сбрасывать соединение TCP;
- освобождать все ресурсы BGP;
- переходить в состояние Idle.

Если получено сообщение OPEN при запущенном таймере DelayOpenTimer (Событие 20), локальная система будет:

- останавливать таймер ConnectRetryTimer (если тот включен) и сбрасывать его значение в 0;
- завершать инициализацию BGP;

- останавливать и сбрасывать в 0 таймер DelayOpenTimer;
- передавать сообщение OPEN;
- передавать сообщение KEEPALIVE;
- если начальное значение таймера HoldTimer отлично от 0:
  - запускается таймер KeepaliveTimer с начальным значением;
  - таймер HoldTimer сбрасывается в согласованное значение,

  в противном случае (начальное значение HoldTimer равно 0)
  - сбрасывается таймер KeepaliveTimer;
  - таймер HoldTimer сбрасывается в 0,
- система переходит в состояние OpenConfirm.

Если значение поля AS совпадает с номером локальной автономной системы, для соединения устанавливается статус внутреннего, в противном случае соединение считается внешним.

Если обнаружены ошибки при проверке заголовка BGP (Событие 21) или сообщения OPEN (Событие 22) (см. параграф 6.2), локальная система будет:

- (*необязательно*) если атрибут SendNOTIFICATIONwithoutOPEN имеет значение TRUE, локальная система сначала будет передавать сообщение NOTIFICATION с соответствующим кодом ошибки;
- останавливать таймер ConnectRetryTimer (если тот включен) и сбрасывать его значение в 0;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- увеличивать значение ConnectRetryCounter на 1;
- (*необязательно*) выполнять процедуру подавления осцилляций, если DampPeerOscillations = TRUE;
- переходить в состояние Idle.

При получении сообщения NOTIFICATION об ошибке верификации (Событие 24), локальная система проверяет таймер DelayOpenTimer. Если этот таймер запущен, локальная система будет:

- останавливать таймер ConnectRetryTimer (если тот включен) и сбрасывать его значение в 0;
- останавливать и сбрасывать в 0 таймер DelayOpenTimer;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- переходить в состояние Idle.

Если таймер DelayOpenTimer не запущен, локальная система будет:

- останавливать таймер ConnectRetryTimer (если тот включен) и сбрасывать его значение в 0;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- увеличивать значение ConnectRetryCounter на 1;
- выполнять процедуру подавления осцилляций, если DampPeerOscillations = TRUE;
- переходить в состояние Idle.

В ответ на все остальные события (8, 10-11, 13, 19, 23, 25-28) локальная система будет:

- если таймер ConnectRetryTimer запущен, – останавливать и сбрасывать его в 0;
- если таймер DelayOpenTimer, – останавливать и сбрасывать его в 0;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- увеличивать значение ConnectRetryCounter на 1;
- выполнять процедуру подавления осцилляций, если DampPeerOscillations = TRUE;
- переходить в состояние Idle.

Состояние Active

В этом состоянии BGP FSM пытается приобрести партнеров путем прослушивания и восприятия соединений TCP.

Стартовые события (1, 3-7) игнорируются в состоянии Active. В ответ на событие ManualStop (2) локальная система будет:

- при запущенном таймере DelayOpenTimer и установленном атрибуте SendNOTIFICATIONwithoutOPEN передавать сообщение NOTIFICATION с кодом ошибки Cease;
- освобождать все ресурсы BGP и останавливать таймер DelayOpenTimer;
- сбрасывать соединение TCP;
- устанавливать ConnectRetryCounter = 0;
- останавливать таймер ConnectRetryTimer и сбрасывать его значение в 0;
- переходить в состояние Idle.

В ответ на событие ConnectRetryTimer_Expires (9) локальная система будет:

- заново запускать таймер ConnectRetryTimer (с начальным значением);
- инициировать соединение TCP с другим узлом BGP;
- продолжать прослушивание входящих вызовов TCP, которые могут приходить от удаленных узлов BGP;
- переходить в состояние Connect.

В ответ на событие DelayOpenTimer_Expires (12) локальная система будет:

- устанавливать ConnectRetryCounter = 0;

- останавливать и сбрасывать в 0 таймер DelayOpenTimer;
- завершать инициализацию BGP;
- передавать удаленному узлу сообщение OPEN;
- устанавливать большое значение для таймера удержания;
- переходить в состояние OpenSent.

Для этого перехода предлагается устанавливать значение HoldTimer равным 4 минутам.

При получении сведений о событии TcpConnection_Valid (14) локальная система обрабатывает флаги соединения TCP и остается в состоянии Active.

При получении информации о событии Tcp_CR_Invalid (15) локальная система отвергнет соединение TCP и сохранит состояние Active.

При успешной организации соединения TCP (Событие 16 или 17) локальная система будет проверять сначала дополнительный атрибут DelayOpen.

Если DelayOpen = TRUE, локальная система будет:

- останавливать таймер ConnectRetryTimer и сбрасывать его значение в 0;
- устанавливать для таймера DelayOpenTimer начальное значение (DelayOpenTime);
- сохранять состояние Active.

Если DelayOpen = FALSE, локальная система будет:

- устанавливать ConnectRetryTimer = 0;
- завершать инициализацию BGP;
- передавать удаленному узлу сообщение OPEN;
- устанавливать большое значение для таймера удержания;
- переходить в состояние OpenSent.

Для этого перехода предлагается устанавливать значение HoldTimer равным 4 минутам.

В ответ на событие TcpConnectionFails (18) локальная система будет:

- заново запускать таймер ConnectRetryTimer (с начальным значением);
- останавливать и сбрасывать в 0 таймер DelayOpenTimer;
- освобождать все ресурсы BGP;
- увеличивать значение ConnectRetryCounter на 1;
- (*необязательно*) выполнять процедуру подавления осцилляций, если DampPeerOscillations = TRUE;
- переходить в состояние Idle.

Если получено сообщение OPEN и запущен таймер DelayOpenTimer (Событие 20), локальная система будет:

- останавливать таймер ConnectRetryTimer (если тот запущен) и сбрасывать его значение в 0;
- останавливать и сбрасывать в 0 таймер DelayOpenTimer;
- завершать инициализацию BGP;
- передавать сообщение OPEN;
- передавать сообщение KEEPALIVE;
- если значение HoldTimer отлично от 0:
  - запускать таймер KeepaliveTimer с начальным значением;
  - сбрасывать таймер HoldTimer в согласованное значение,

  если HoldTimer = 0
  - сбрасывать таймер KeepaliveTimer (0);
  - сбрасывать в 0 таймер HoldTimer;
- переходить в состояние OpenConfirm.

Если в поле AS содержится номер локальной автономной системы, соединение относится к числу внутренних, в противном случае считается внешним.

Если обнаружены ошибки при проверке заголовка BGP (Событие 21) или сообщения OPEN (Событие 22) (см. параграф 6.2), локальная система будет:

- (*необязательно*) если атрибут SendNOTIFICATIONwithoutOPEN имеет значение TRUE, локальная система сначала будет передавать сообщение NOTIFICATION с соответствующим кодом ошибки;
- останавливать таймер ConnectRetryTimer (если тот включен) и сбрасывать его значение в 0;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- увеличивать значение ConnectRetryCounter на 1;
- (*необязательно*) выполнять процедуру подавления осцилляций, если DampPeerOscillations = TRUE;
- переходить в состояние Idle.

При получении сообщения NOTIFICATION об ошибке верификации (Событие 24), локальная система проверяет таймер DelayOpenTimer. Если этот таймер запущен, локальная система будет:

- останавливать таймер ConnectRetryTimer (если тот включен) и сбрасывать его значение в 0;
- останавливать и сбрасывать в 0 таймер DelayOpenTimer;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- переходить в состояние Idle.

Если таймер DelayOpenTimer не запущен, локальная система будет:

- устанавливать ConnectRetryTimer = 0;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- увеличивать значение ConnectRetryCounter на 1;
- (*необязательно*) выполнять процедуру подавления осцилляций, если DampPeerOscillations = TRUE;
- переходить в состояние Idle.

В ответ на любое другое событие (8, 10-11, 13, 19, 23, 25-28) локальная система будет:

- устанавливать ConnectRetryTimer = 0;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- увеличивать значение ConnectRetryCounter на 1;
- (*необязательно*) выполнять процедуру подавления осцилляций, если DampPeerOscillations = TRUE;
- переходить в состояние Idle.

Состояние OpenSent

В этом состоянии BGP FSM ожидает сообщения OPEN от партнера.

Стартовые события (1, 3-7) игнорируются в состоянии OpenSent.

Если в состоянии OpenSent происходит событие ManualStop (2), локальная система будет:

- передавать сообщение NOTIFICATION с кодом Cease;
- устанавливать ConnectRetryTimer = 0;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- устанавливать ConnectRetryCounter = 0;
- переходить в состояние Idle.

Если в состоянии OpenSent происходит событие AutomaticStop (8), локальная система будет:

- передавать сообщение NOTIFICATION с кодом Cease;
- устанавливать ConnectRetryTimer = 0;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- увеличивать значение ConnectRetryCounter на 1;
- (*необязательно*) выполнять процедуру подавления осцилляций, если DampPeerOscillations = TRUE;
- переходить в состояние Idle.

В ответ на событие HoldTimer_Expires (10) локальная система будет:

- передавать сообщение NOTIFICATION с кодом ошибки Hold Timer Expired;
- устанавливать ConnectRetryTimer = 0;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- увеличивать значение ConnectRetryCounter на 1;
- (*необязательно*) выполнять процедуру подавления осцилляций, если DampPeerOscillations = TRUE;
- переходить в состояние Idle.

События TcpConnection_Valid (14), Tcp_CR_Acked (16) или TcpConnectionConfirmed (17) говорят о том, что может иметь место попытка организации второго соединения TCP. Это второе соединение находится под контролем системы обработки конфликтов при соединениях (параграф 6.8), пока не будет принято сообщение OPEN.

Запросы соединений TCP через некорректный порт (Tcp_CR_Invalid - Событие 15)) игнорируются.

При получении информации о событии TcpConnectionFails (18) локальная система будет:

- закрывать соединение BGP;
- заново запускать таймер ConnectRetryTimer;
- продолжать прослушивание порта на предмет вызовов от удаленных узлов BGP;
- переходить в состояние Active.

При получении сообщения OPEN проверяется корректность всех полей этого сообщения. Если сообщение OPEN не содержит ошибок (Событие 19), локальная система будет:

- сбрасывать в 0 таймер DelayOpenTimer;
- устанавливать для таймера ConnectRetryTimer значение 0;
- передавать сообщение KEEPALIVE;
- устанавливать значение таймера KeepaliveTimer (см. ниже);
- устанавливать для таймера HoldTimer согласованное значение (см. параграф 4.2);
- переходить в состояние OpenConfirm.

Если согласованное время удержания равно 0, таймеры HoldTimer и KeepaliveTimer не запускаются. Если значение поля My Autonomous System совпадает с номером локальной AS, соединение трактуется как внутреннее, в противном случае относится к числу внешних (это будет оказывать влияние на описанную ниже обработку сообщений UPDATE).

Если обнаружены ошибки при проверке заголовка BGP (Событие 21) или сообщения OPEN (Событие 22) (см. параграф 6.2), локальная система будет:

- передавать сообщение NOTIFICATION с соответствующим кодом ошибки;

- устанавливать ConnectRetryTimer = 0;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- увеличивать значение ConnectRetryCounter на 1;
- (*необязательно*) выполнять процедуру подавления осцилляций, если DampPeerOscillations = TRUE;
- переходить в состояние Idle.

При получении корректного сообщения BGP OPEN (Событие 19 или 20) требуется применять механизм детектирования конфликтов (параграф 6.8).

Событие CollisionDetectDump происходит, когда реализация BGP определяет наличие конфликта при соединении (рассмотрение этих механизмов выходит за пределы данного документа).

Если в состоянии OpenSent возникает необходимость закрыть соединение, машине состояний передается сигнал OpenCollisionDump (Событие 23). При получении такого события в состоянии OpenSent локальная система будет:

- передавать сообщение NOTIFICATION с кодом Cease;
- устанавливать ConnectRetryTimer = 0;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- увеличивать значение ConnectRetryCounter на 1;
- (*необязательно*) выполнять процедуру подавления осцилляций, если DampPeerOscillations = TRUE;
- переходить в состояние Idle.

Если получено сообщение NOTIFICATION с кодом ошибки несоответствия версий (Событие 24), локальная система будет:

- устанавливать ConnectRetryTimer = 0;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- переходить в состояние Idle.

В ответ на любое другой событие (9, 11-13, 20, 25-28) локальная система будет:

- передавать сообщение NOTIFICATION с кодом Finite State Machine Error;
- устанавливать ConnectRetryTimer = 0;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- увеличивать значение ConnectRetryCounter на 1;
- (*необязательно*) выполнять процедуру подавления осцилляций, если DampPeerOscillations = TRUE;
- переходить в состояние Idle.

Состояние OpenConfirm

В этом состоянии BGP FSM ожидает приема сообщения KEEPALIVE или NOTIFICATION.

Любый стартовые события (1, 3-7) игнорируются в состоянии OpenConfirm.

В ответ на событие ManualStop (2), инициированное оператором, локальная система будет:

- передавать сообщение NOTIFICATION с кодом Cease;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- устанавливать ConnectRetryCounter = 0;
- устанавливать ConnectRetryTimer = 0;
- переходить в состояние Idle.

В ответ на событие AutomaticStop (8), инициированное системой, локальная система будет:

- передавать сообщение NOTIFICATION с кодом Cease;
- устанавливать ConnectRetryTimer = 0;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- увеличивать значение ConnectRetryCounter на 1;
- (*необязательно*) выполнять процедуру подавления осцилляций, если DampPeerOscillations = TRUE;
- переходить в состояние Idle.

Если событие HoldTimer_Expires (Событие 10) происходит до получения сообщения KEEPALIVE, локальная система будет:

- передавать сообщение NOTIFICATION с кодом ошибки Hold Timer Expired,
- устанавливать ConnectRetryTimer = 0;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- увеличивать значение ConnectRetryCounter на 1;
- (*необязательно*) выполнять процедуру подавления осцилляций, если DampPeerOscillations = TRUE;
- переходить в состояние Idle.

Если локальная система получает сигнал KeepaliveTimer_Expires (Событие 11), она будет:

- передавать сообщение KEEPALIVE;

- заново запускать таймер KeepaliveTimer;
- сохранять состояние OpenConfirmed.

Событие TcpConnection_Valid (14) или успешная организация соединения TCP (Событие 16 или 17) в состоянии OpenConfirm требуют от локальной системы проверки отсутствия второго соединения (с тем же партнером).

При попытке организации соединения TCP через некорректный порт (Событие 15) локальная система будет игнорировать вторую попытку организации соединения.

Если локальная система получает сигнал TcpConnectionFails (Событие 18) от нижележащего уровня TCP или сообщение NOTIFICATION (Событие 25), она будет:

- устанавливать ConnectRetryTimer = 0;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- увеличивать значение ConnectRetryCounter на 1;
- (*необязательно*) выполнять процедуру подавления осцилляций, если DampPeerOscillations = TRUE;
- переходить в состояние Idle.

Если локальная система получает сообщение NOTIFICATION с кодом ошибки несоответствия версий (NotifMsgVerErr - Событие 24)), она будет:

- устанавливать ConnectRetryTimer = 0;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- переходить в состояние Idle.

Если локальная система получает корректное сообщение OPEN (BGPOpen - Событие 19), выполняется процесс детектирования конфликтов, описанный в параграфе 6.8. Если в результате данное соединение будет разрываться, локальная система будет:

- передавать сообщение NOTIFICATION с кодом Cease;
- устанавливать ConnectRetryTimer = 0;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP (пакет TCP FIN);
- увеличивать значение ConnectRetryCounter на 1;
- (*необязательно*) выполнять процедуру подавления осцилляций, если DampPeerOscillations = TRUE;
- переходить в состояние Idle.

Если получено сообщение OPEN, проверяется корректность всех полей этого сообщения. Если обнаружены ошибки при проверке заголовка BGP (Событие 21) или сообщения OPEN (Событие 22) (см. параграф 6.2), локальная система будет:

- передавать сообщение NOTIFICATION с соответствующим кодом ошибки;
- устанавливать ConnectRetryTimer = 0;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- увеличивать значение ConnectRetryCounter на 1;
- (*необязательно*) выполнять процедуру подавления осцилляций, если DampPeerOscillations = TRUE;
- переходить в состояние Idle.

Если (в процессе обработки другого сообщения OPEN) реализация BGP определяет (способы детектирования выходят за пределы данного документа), что произошел конфликт при соединении и данное соединение будет закрыто, локальная система будет подавать сигнал OpenCollisionDump (Событие 23). При получении сигнала OpenCollisionDump (Событие 23) локальная система будет:

- передавать сообщение NOTIFICATION с кодом Cease;
- устанавливать ConnectRetryTimer = 0;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- увеличивать значение ConnectRetryCounter на 1;
- (*необязательно*) выполнять процедуру подавления осцилляций, если DampPeerOscillations = TRUE;
- переходить в состояние Idle.

При получении сообщения KEEPALIVE (KeepAliveMsg - Событие 26) локальная система будет:

- заново запускать таймер HoldTimer;
- переходить в состояние Established.

В ответ на все остальные события (9, 12-13, 20, 27-28) локальная система будет:

- передавать сообщение NOTIFICATION с соответствующим кодом Finite State Machine Error;
- устанавливать ConnectRetryTimer = 0;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- увеличивать значение ConnectRetryCounter на 1;
- (*необязательно*) выполнять процедуру подавления осцилляций, если DampPeerOscillations = TRUE;
- переходить в состояние Idle.

Состояние Established

В состоянии Established машина BGP FSM может обмениваться сообщениями UPDATE, NOTIFICATION и KEEPALIVE со своим партнером. Любые стартовые события (1, 3-7) игнорируются в состоянии Established.

В ответ на событие ManualStop (Событие 2), инициированное оператором, локальная система будет:

- передавать сообщение NOTIFICATION с кодом Cease;
- устанавливать ConnectRetryTimer = 0;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- устанавливать ConnectRetryCounter = 0;
- переходить в состояние Idle.

В ответ на событие AutomaticStop (8) локальная система будет:

- передавать сообщение NOTIFICATION с кодом Cease;
- устанавливать ConnectRetryTimer = 0;
- удалять все маршруты, связанные с этим соединением;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- увеличивать значение ConnectRetryCounter на 1;
- (*необязательно*) выполнять процедуру подавления осцилляций, если DampPeerOscillations = TRUE;
- переходить в состояние Idle.

Одной из причин сигнала AutomaticStop может быть получение сообщений UPDATE с числом префиксов, превышающим заданный предел общего числа префиксов. Локальная система будет автоматически разрывать соединение с данным партнером.

В ответ на событие HoldTimer_Expires (10) локальная система будет:

- передавать сообщение NOTIFICATION с кодом ошибки Hold Timer Expired;
- устанавливать ConnectRetryTimer = 0;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- увеличивать значение ConnectRetryCounter на 1;
- (*необязательно*) выполнять процедуру подавления осцилляций, если DampPeerOscillations attribute = TRUE;
- переходить в состояние Idle.

По сигналу KeepaliveTimer_Expires (Событие 11) локальная система будет:

- передавать сообщение KEEPALIVE;
- заново запускать таймер KeepaliveTimer, если согласованное значение HoldTime не равно 0.

Каждый раз при получении локальной системой сообщения KEEPALIVE или UPDATE она заново запускает таймер KeepaliveTimer, если согласованное значение HoldTime не равно 0.

Сигнал TcpConnection_Valid (Событие 14), принятый для корректного порта, будет инициировать систему отслеживания второго соединения.

Некорректные соединения (Tcp_CR_Invalid - Событие 15) будут игнорироваться.

В ответ на индикацию завершения организации соединения TCP (Событие 16 или 17) **следует** отслеживать второе соединение, пока не будет передано сообщение OPEN.

Если получено корректное сообщение OPEN (BGPOpen - Событие 19) и CollisionDetectEstablishedState = TRUE, сообщение OPEN будет проверяться на предмет конфликта (параграф 6.8) с другими соединениями. Если реализация BGP принимает решение о разрыве данного соединения, она будет генерировать сигнал OpenCollisionDump (Событие 23). Если соединение нужно разорвать, локальная система будет:

- передавать сообщение NOTIFICATION с кодом Cease;
- устанавливать ConnectRetryTimer = 0;
- удалять все маршруты, связанные с соединением;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- увеличивать значение ConnectRetryCounter на 1;
- (*необязательно*) выполнять процедуру подавления осцилляций, если DampPeerOscillations = TRUE;
- переходить в состояние Idle.

Если локальная система получает сообщение NOTIFICATION (Событие 24 или 25) или сигнал TcpConnectionFails (Событие 18) от нижележащего уровня TCP, она будет:

- устанавливать ConnectRetryTimer = 0;
- удалять все маршруты, связанные с соединением;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- увеличивать значение ConnectRetryCounter на 1;
- переходить в состояние Idle.

При получении сообщения KEEPALIVE (Событие 26) локальная система будет:

- заново запускать таймер HoldTimer, если согласованное значение HoldTime не равно 0;
- сохранять состояние Established.

В ответ на сообщение UPDATE (Событие 27) локальная система будет:

- обрабатывать принятое сообщение;
- заново запускать таймер HoldTimer, если согласованное значение HoldTime не равно 0;
- сохранять состояние Established.

Если локальная система получает сообщение UPDATE и процедура контроля ошибок в сообщениях UPDATE (см. параграф 6.3) обнаруживает ошибку (Событие 28), локальная система будет:

- передавать сообщение NOTIFICATION с кодом Update Error;
- устанавливать ConnectRetryTimer = 0;
- удалять все маршруты, связанные с соединением;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- увеличивать значение ConnectRetryCounter на 1;
- (*необязательно*) выполнять процедуру подавления осцилляций, если DampPeerOscillations = TRUE;
- переходить в состояние Idle.

В ответ на все прочие события (9, 12-13, 20-22) локальная система будет:

- передавать сообщение NOTIFICATION с кодом Finite State Machine Error,
- удалять все маршруты, связанные с соединением;
- устанавливать ConnectRetryTimer = 0;
- освобождать все ресурсы BGP;
- сбрасывать соединение TCP;
- увеличивать значение ConnectRetryCounter на 1;
- (*необязательно*) выполнять процедуру подавления осцилляций, если DampPeerOscillations = TRUE;
- переходить в состояние Idle.

9. Обработка сообщений UPDATE

Сообщение UPDATE может быть получено только в состоянии Established. Получение такого сообщения в любом другом состоянии является ошибкой. При получении сообщения UPDATE проверяется корректность каждого поля в соответствии с параграфом 6.3.

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

Если дополнительный атрибут распознан и имеет корректное значение, тогда (в зависимости от типа дополнительного атрибута) этот атрибут обрабатывается локально, сохраняется и обновляется (при необходимости) для возможной передачи другим узлам BGP.

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

Если сообщение UPDATE содержит доступный маршрут, таблицу Adj-RIB-In следует обновить, добавив этот маршрут, как указано здесь - если поле NLRI нового маршрута идентично одному из маршрутов, хранящихся в Adj-RIB-In, новый маршрут **нужно** поместить взамен имеющегося в Adj-RIB-In (таким образом, неявно отзывая более старый маршрут). В противном случае, если Adj-RIB-In не содержит маршрута с идентичным значением NLRI, новый маршрут **следует** включить в таблицу Adj-RIB-In.

После того, как узел BGP обновит базу Adj-RIB-In, ему **следует** инициировать процесс выбора маршрутов (Decision Process).

9.1. Процесс выбора маршрутов (Decision Process)

Процесс выбора маршрутов (Decision Process) обеспечивает выбор маршрутов для последующего анонсирования путем применения правил, заданных в локальной базе PIB (Policy Information Base), к маршрутам из базы Adj-RIB-In. Результатом процесса является набор маршрутов, которые будут анонсироваться всем партнерам, – эти маршруты хранятся в локальной базе Adj-RIB-Out в соответствии с политикой.

BGP Decision Process описывается здесь концептуально и не обязательно должен быть реализован в точном соответствии с этим описанием. Если реализация поддерживает описанную функциональность, она будет демонстрировать внешнее поведение, соответствующее данному описанию.

Процесс выбора формализуется путем определения функций, принимающих атрибуты данного маршрута в качестве аргументов и возвращающих (a) неотрицательное целое число, которое задает уровень предпочтения для данного маршрута, или (b) значение, показывающее, что данный маршрут нежелательно включать в Loc-RIB и он будет исключен рассмотрения на последующих этапах процесса выбора.

В функции, вычисляющей уровень предпочтения для данного маршрута, **не следует** использовать в качестве входных данных перечисленной здесь информации – существование других маршрутов, отсутствие других маршрутов, атрибуты пути для других маршрутов. Процесс выбора маршрутов состоит из независимого определения уровня предпочтения каждого доступного маршрута и последующего выбора одного из маршрутов с максимальным уровнем предпочтения.

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

- выбор маршрутов, которые будут использоваться локальным узлом;
- выбор маршрутов, которые будут анонсироваться партнерам BGP;
- агрегирование (объединение) маршрутов и снижение объема маршрутной информации.

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

a) Фаза 1 отвечает за расчет уровня предпочтения для каждого маршрута, полученного от партнеров.

b) 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 Loc-RIB.

c) Фаза 3 начинается после обновления Loc-RIB и отвечает за распространение маршрутов из Loc-RIB всем партнерам в соответствии с политикой, содержащейся в PIB. На этой фазе также может выполняться объединение маршрутов и снижение объема маршрутной информации.

9.1.1. Фаза 1: Расчет предпочтений (Calculation of Degree of Preference)

Фаза 1 активизируется всякий раз, когда локальный узел BGP получает от партнера сообщение UPDATE, анонсирующее новый маршрут, замену или отзыв маршрута.

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

Функция, используемая в фазе 1, блокирует базу Adj-RIB-In прежде, чем начать работу с содержащимися в ней маршрутами и снимет блокировку по завершении работы со всеми новыми или недоступными маршрутами в этой базе.

При получении нового маршрута или замене доступного маршрута локальный узел BGP определяет уровень предпочтения, как описано ниже:

Если маршрут получен от внутреннего партнера в качестве уровня предпочтения принимается значение атрибута LOCAL_PREF или локальная система рассчитывает этот уровень на основе заданных конфигурацией параметров политики. Отметим, что последний вариант может приводить к возникновению постоянных маршрутных петель.

Если маршрут получен от внешнего партнера, локальный узел BGP рассчитывает уровень предпочтения на основе заданных конфигурацией параметров политики. Если полученное значение показывает, что маршрут является неподходящим, этот маршрут **может не** передаваться на следующий этап процесса выбора, в противном случае возвращенное значение **должно** использоваться как значение LOCAL_PREF в любом повторном анонсе IBGP.

Конкретные правила политики и методы расчета уровня предпочтения задаются локально.

9.1.2. Фаза 2: Выбор маршрута (Route Selection)

Фаза 2 выполняется после завершения фазы 1 и представляет собой отдельный процесс, который завершается после выполнения всех необходимых операций. В фазе 2 используются все маршруты из базы Adj-RIBs-In.

Активизация фазы 2 блокируется, пока не будет завершена работа фазы 3. Функция фазы 2 блокирует целиком базу Adj-RIBs-In перед началом работы и снимает блокировку по завершении расчета.

Если атрибут NEXT_HOP маршрута BGP указывает непреобразуемый адрес[1] или этот адрес не будет преобразовываться после установки маршрута, такой маршрут BGP **должен** исключаться из обработки в фазе 2.

Если атрибут AS_PATH маршрута BGP содержит петлю (AS loop), маршрут BGP следует исключить из рассмотрения в фазе 2. Детектирование петель осуществляется путем полного сканирования пути, указанного в атрибуте AS_PATH, и проверки отсутствия в нем номера локальной AS. Обсуждение работы узлов BGP, настроенных на восприятие маршрутов с номером AS этого узла в атрибутах пути, выходит за пределы данного документа.

Важно, чтобы узлы BGP внутри AS при выборе маршрутов не принимали конфликтующих решений, которые будут приводить к возникновению петель при пересылке пакетов.

Для каждого набора адресатов, к которому существует доступный маршрут в таблице Adj-RIBs-In, локальный узел BGP идентифицирует маршрут, соответствующий одному из перечисленных условий:

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

b) маршрут является единственным для данного множества адресатов;

c) маршрут выбран в результате применения в фазе 2 правил «отбрасывания лишнего» (tie breaking) описанных в параграфе 9.1.2.2.

После этого локальному узлу **следует** установить маршрут в таблицу Loc-RIB, заменяя любой маршрут к тому же адресату, который в настоящее время хранится в Loc-RIB. После включения нового маршрута BGP в таблицу маршрутизации следует принять меры по удалению из таблицы других маршрутов к тому же адресату. Принятие решения о замене имеющегося в таблице маршрута, полученного не от BGP, на маршрут BGP определяется локальной политикой узла BGP.

Локальный узел **должен** определить адрес ближайшего маршрутизатора (immediate next-hop) из атрибута NEXT_HOP выбранного маршрута (см. параграф 5.1.3). При смене ближайшего маршрутизатора или стоимости IGP[2] до NEXT_HOP (NEXT_HOP преобразуется через маршрут IGP) выбор маршрута в фазе 2 **должен** быть проведен заново.

Отметим, что даже в тех случаях, когда маршруты BGP не включаются в таблицу маршрутизации с immediate next-hop, реализация **должна** принять меры для того, чтобы адрес NEXT_HOP был преобразован в адрес подключенного напрямую следующего маршрутизатора[3] до того, как по маршруту BGP будет начата пересылка пакетов, и этот адрес (адреса) использовался при реальной пересылке пакетов.

Маршруты, которые невозможно преобразовать, **следует** удалить из базы Loc-RIB и таблицы маршрутизации. Однако соответствующие непреобразуемые маршруты **следует** сохранить в базе Adj-RIBs-In (впоследствии их преобразование может стать возможным).

9.1.2.1. Возможность преобразования маршрута

Как сказано в параграфе 9.1.2, узлам BGP **следует** исключать непреобразуемые маршруты из рассмотрения в фазе 2. Это позволяет включать в базу Loc-RIB и таблицу маршрутизации только действующие маршруты.

Возможность преобразования маршрута определяется перечисленными ниже условиями:

1) Маршрут Rte1, указанный только промежуточным адресом, рассматривается как преобразуемый, если таблица маршрутизации содержит по крайней мере один маршрут Rte2, который соответствует промежуточному адресу маршрута Rte1 и преобразуется без рекурсии (непосредственно или опосредованно) через Rte1. Пр наличии более одного такого маршрута **следует** рассматривать только маршрут с максимальным соответствием.

[1] Локальный узел не имеет маршрута для этого адреса в своей таблице. *Прим. перев.*
[2] Протокола внутренней маршрутизации. *Прим. перев.*
[3] immediate (directly connected) next-hop address

2) Маршруты, указанные интерфейсами (с промежуточным адресом или без него) считаются преобразуемыми, если указанный для маршрута интерфейс активен и для него включена обработка IP.

Маршруты BGP не включают интерфейсов, но могут быть преобразованы в маршруты из таблицы маршрутизации, которые могут относиться к обоим перечисленным выше типам (т. е., задавать или не задавать интерфейс). Предполагается, что маршруты IGP и маршруты в непосредственно подключенные сети задают выходной интерфейс. Статические маршруты могут задавать выходной интерфейс, промежуточный адрес или оба параметра.

Отметим, что маршрут BGP рассматривается как непреобразуемый в ситуации, когда таблица маршрутизации узла BGP не включает маршрута, соответствующего значению NEXT_HOP из маршрута BGP. Взаимно-рекурсивные маршруты (маршруты, преобразующие друг друга или самого себя) также не проходят проверку на возможность преобразования.

Важно также обеспечить исключение доступных маршрутов, которые станут непреобразуемыми после их включения в таблицу маршрутизации даже если значение NEXT_HOP для такого маршрута может быть преобразовано в текущем контексте таблицы маршрутизации (примером могут служить взаимно-рекурсивные маршруты). Эта проверка гарантирует, что узел BGP не включит в свою таблицу маршруты, которые будут удалены и не будут использоваться узлом. Следовательно, в дополнение к стабильности локальной таблицы маршрутизации эта проверка также делает более эффективной работу протокола в сети.

В тех случаях, когда узел BGP идентифицирует маршрут как непреобразуемый по причине взаиморекурсии, **следует** записывать сообщение об ошибке в журнальный файл системы.

9.1.2.2. “Отбрасывание лишнего” (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. К рассмотрению принимаются все маршруты с одинаковым уровнем предпочтения, полученные как от внутренних, так и от внешних партнеров.

В описанной ниже процедуре 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, и применяется один алгоритм выбора маршрутов.

Работа алгоритма tie-breaking начинается с рассмотрения всех маршрутов к одному множеству адресатов, имеющих одинаковый уровень предпочтения, и выбором маршрутов, которые будут исключаться из рассмотрения. Работа алгоритма завершается после того, как останется единственный маршрут. Критерии отбора **должны** применяться в указанном ниже порядке.

Некоторые критерии описаны с использованием псевдокода. Отметим, что выбор псевдокода был продиктован соображениями ясности, а не эффективности. Он не предназначен для конкретной реализации. Реализации протокола BGP **могут** использовать любой алгоритм, который будет давать такой же результат, как описано здесь.

a) Исключаются из рассмотрения все маршруты, с числом номеров AS в атрибуте AS_PATH, большим минимального значения. Отметим, что при расчете этого значения AS_SET учитывается как 1, независимо от количества AS в данном наборе.

b) Исключаются из рассмотрения все маршруты, для которых значение атрибута Origin превышает минимальное.

c) Исключаются из рассмотрения маршруты с менее предпочтительными атрибутами MULTI_EXIT_DISC. Значения MULTI_EXIT_DISC можно сравнивать только для маршрутов, полученных из одной соседней AS (эта AS определяется из атрибута AS_PATH). Маршруты без атрибута MULTI_EXIT_DISC рассматриваются как маршруты с наименьшим возможным значением MULTI_EXIT_DISC.

   Описанный выше алгоритм можно представить в виде следующей процедуры:

```
for m = число остающихся в рассмотрении маршрутов
    for n = число остающихся в рассмотрении маршрутов
        if (neighborAS(m) == neighborAS(n)) and (MED(n) < MED(m))
            исключить маршрут m из рассмотрения
```

   В приведенном выше псевдокоде функция MED(n) возвращает значение атрибута MULTI_EXIT_DISC для маршрута n. Если маршрут n не имеет атрибута MULTI_EXIT_DISC, функция возвращает минимальное из возможных значений MULTI_EXIT_DISC (т. е., 0).

   Функция neighborAS(n) возвращает номер соседней AS, из которой был получен маршрут n. Если маршрут получен через IBGP и другой узел IBGP не является исходной точкой этого маршрута, это будет номер соседней AS, из которой другой узел IBGP получил маршрут. Если маршрут получен через IBGP и другой узел IBGP (a) является исходной точкой маршрута или (b) создал маршрут путем агрегирования и атрибут AS_PATH агрегированного маршрута пуст или начинается с AS_SET, это локальная AS.

   Если атрибут MULTI_EXIT_DISC удаляется до повторного анонсирования маршрута в IBGP, **можно** провести сравнение с использованием полученного через EBGP атрибута MULTI_EXIT_DISC. Если реализация решает удалить MULTI_EXIT_DISC, тогда дополнительное сравнение MULTI_EXIT_DISC, если оно выполняется, **должно** учитывать только маршруты, полученные через EBGP. Наилучший маршрут от EBGP можно тогда сравнивать с маршрутами от IBGP после удаления атрибута MULTI_EXIT_DISC. Если атрибут MULTI_EXIT_DISC удаляется из подмножества маршрутов от EBGP и из выбранного “лучшего” маршрута от EBGP не будет удален атрибут MULTI_EXIT_DISC, тогда этот атрибут должен использоваться для сравнения с маршрутами от IBGP. Для полученных через IBGP маршрутов атрибут MULTI_EXIT_DISC **должен** использоваться при сравнении маршрутов, которые не исключены на предыдущих этапах выбора (Decision Process). Включение атрибута MULTI_EXIT_DISC маршрутов от EBGP в сравнение с маршрутами от IBGP с последующим удалением атрибута MULTI_EXIT_DISC и анонсированием маршрута будет предотвращать возникновение маршрутных петель.

d) Если хотя бы один из маршрутов-кандидатов получен через EBGP, исключаются из рассмотрения все маршруты, полученные от IBGP.

e) Исключаются из рассмотрения все маршруты с наименее предпочтительной внутренней стоимостью (interior cost). Внутренняя стоимость маршрута определяется путем расчета метрики до NEXT_HOP для маршрута с использованием таблицы маршрутизации. Если маршрутизатор NEXT_HOP для этого маршрута доступен, но стоимость пути невозможно определить, этот этап следует пропустить (возможно, рассматривая все маршруты, как равноценные).

Описанный выше алгоритм можно представить псевдокодом.

```
for m = число остающихся в рассмотрении маршрутов
    for n = число остающихся в рассмотрении маршрутов
        if (cost(n) < cost(m))
            исключить маршрут m из рассмотрения
```

В приведенном псевдокоде функция cost(n) возвращает стоимость пути (внутренняя дистанция) до адреса, указанного в атрибуте NEXT_HOP рассматриваемого маршрута.

f) Исключаются из рассмотрения все маршруты, кроме того, который был анонсирован узлом BGP с наименьшим значением BGP Identifier[1].

g) Выбирается маршрут, полученный от партнера с наименьшим адресом.

9.1.3. Фаза 3: Распространение маршрутов (Route Dissemination)

Фаза 3 выполняется после завершения операций фазы 2 или по любому из перечисленных ниже событий:

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

b) изменение локально сгенерированных маршрутов, которые не были получены от BGP;

c) организация соединения с новым узлом BGP.

Функция фазы 3 представляет собой отдельный процесс, работа которого завершается после выполнения всех требуемых действий. Функция фазы 3 блокируется на время работы функции фазы 2.

Все маршруты базы Loc-RIB обрабатываются для включения в Adj-RIBs-Out, согласно заданной конфигурацией политике. Эта политика **может** исключать маршруты, содержащиеся в Loc-RIB, из числа добавляемых в базу Adj-RIB-Out. Маршрут **не следует** устанавливать в Adj-Rib-Out, если для адресатов и NEXT_HOP этого маршрута а таблице маршрутизации нет соответствующей записи. Если маршрут из базы Loc-RIB не включается в ту или иную базу Adj-RIB-Out, ранее анонсированный маршрут этой базы Adj-RIB-Out **должен** быть отозван с помощью сообщения UPDATE (см. параграф 9.2).

В этой фазе могут дополнительно применяться методы агрегирования маршрутов и снижения объема маршрутных данных (см. параграф 9.2.2.1).

Вопросы локальной политики, которая может приводить к включению маршрутов в базу Adj-RIB-Out без их добавления в таблицу пересылки локального узла BGP выходят за пределы данного документа.

После завершения процессов обновления Adj-RIBs-Out и таблицы маршрутизации локальный узел BGP запускает процесс передачи обновлений (Update-Send - параграф 9.2).

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.

Набор адресатов, описываемый перекрытием, представляет часть менее специфичного маршрута, которая доступна, но в настоящее время не используется. Если более специфичный маршрут впоследствии отзывается, описываемые перекрытием адресаты остаются доступными через менее специфичный маршрут.

Если узел BGP получает перекрывающиеся маршруты, процесс выбора маршрутов (Decision Process) **должен** рассматривать оба маршрута на основе заданной конфигурацией политики восприятия маршрутов. Если приемлемы оба маршрута (менее специфичный и более специфичный), процесс выбора **должен** установить в Loc-RIB оба маршрута или объединить их и установить в Loc-RIB агрегированный маршрут, что обеспечивается наличием в обоих маршрутах одинакового значения атрибута NEXT_HOP.

Если узел BGP выбирает агрегирование маршрутов, ему **следует** включить все AS, используемые при формировании агрегированного маршрута, в AS_SET или добавить в маршрут атрибут ATOMIC_AGGREGATE. Данный атрибут в настоящее время используется в основном как информационный. По мере избавления от протоколов маршрутизации IP, хостов и маршрутизаторов, не поддерживающих бесклассовую маршрутизацию, необходимость в деагрегировании маршрутов отпадет. Маршруты **не следует** деагрегировать. В частности, маршруты с атрибутом ATOMIC_AGGREGATE деагрегировать **недопустимо**. Таким образом, значение NLRI такого маршрута не может быть более специфичным. Пересылка по такому маршруту не обеспечивает гарантии, что пакеты IP будут в реальности проходить только через AS, указанные в атрибуте AS_PATH этого маршрута.

9.2. Процесс передачи обновлений (Update-Send)

Процесс передачи обновлений (Update-Send) отвечает за анонсирование сообщений UPDATE всем партнерам. Например, он распространяет маршруты, выбранные Decision Process, другим узлам BGP, которые могут располагаться в той же или соседних AS.

Когда узел BGP получает сообщение UPDATE от внутреннего партнера, принимающему узлу BGP **не следует** заново распространять содержащуюся в сообщении UPDATE информацию другим внутренним узлам (если данный узел не используется как BGP Route Reflector [RFC2796]).

В фазе 3 процесса выбора маршрутов узел BGP обновляет свою базу Adj-RIBs-Out. Все вновь включенные маршруты и все маршруты, ставшие недоступными (если для них нет замены), **следует** анонсировать партнерам в сообщениях UPDATE.

[1]RFC 4456 вносит изменение в это правило и добавляет еще одно правило между f) и g). *Прим. перев.*

Узлу BGP **не следует** анонсировать доступный маршрут BGP из своей базы Adj-RIB-Out, если это будет порождать сообщение UPDATE, содержащее маршрут BGP, который уже был анонсирован.

Все маршруты из базы Loc-RIB, помеченные как недоступные, **следует** удалять. Изменения в состоянии доступности адресатов внутри своей автономной системы также **следует** анонсировать в сообщениях UPDATE.

Если единичный маршрут в силу ограничений на размер сообщений UPDATE (см. главу 4) не помещается в сообщение, для узла BGP **недопустимо** анонсирование этого маршрута; реализация **может** записывать информацию о таких фактах в системный журнал.

9.2.1. Контроль служебного трафика

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

9.2.1.1. Частота анонсирования маршрутов

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

Два сообщения UPDATE, передаваемые партнеру узлом BGP, с анонсами доступных и/или недоступных маршрутов к некоторому общему набору адресатов, должны быть разделены промежутком времени не менее MinRouteAdvertisementInterval. Очевидно, что для достижения этого требуется использовать отдельный таймер для каждого общего набора адресатов. Такой подход будет порождать недопустимую нагрузку (overhead). Для практического применения подходит любой метод, обеспечивающий между двумя последовательными сообщениями UPDATE с анонсом доступных и/или недоступных маршрутов к некому множеству адресатов, адресованными одному партнеру, интервал не менее MinRouteAdvertisementInterval и способный гарантировать приемлемое постоянное значение верхней границы для такого интервала.

Поскольку внутри AS требуется быстрое схождение маршрутов, (a) значение MinRouteAdvertisementIntervalTimer, используемое для внутренних партнеров, **следует** делать меньше значения этого параметра для внешних партнеров или (b) описанную здесь процедуру **не следует** применять для маршрутов, передаваемых внутренним партнерам.

Эта процедура не ограничивает скорость выбора маршрута, внося лишь ограничение на частоту анонсирования. Если новые маршруты были выбраны несколько раз в течение ожидания MinRouteAdvertisementInterval, по завершении этого периода **следует** анонсировать последний выбранный маршрут.

9.2.1.2. 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.

9.2.2. Эффективная организация маршрутных данных

Имея маршрутную информацию для анонсирования, узел BGP может воспользоваться несколькими методами эффективной организации маршрутных данных.

9.2.2.1. Снижение объема информации

Снижение объема информации означает снижение уровня гранулярности контроля над политикой маршрутизации – после сжатия информации одни и те же правила будут применяться ко всем адресатам и путям одного класса.

Процесс выбора маршрутов (Decision Process) может дополнительно снижать объем информации, включаемой в базу Adj-RIBs-Out, любым из перечисленных ниже методов.

a) **Информация о доступности на сетевом уровне (NLRI):**

   IP-адреса получателей могут быть представлены как префиксы 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 об AS в пути может быть упорядоченной (AS_SEQUENCE) или неупорядоченной (AS_SET). Вариант AS_SET используется в алгоритме агрегирования маршрутов, описанном в параграфе 9.2.2.2. Агрегирование снижает объем данных AS_PATH за счет однократного указания номера каждой AS (независимо от числа ее упоминаний во множестве агрегируемых атрибутов AS_PATH).

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

9.2.2.2. Агрегирование маршрутной информации

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

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

Маршруты с разными атрибутами MULTI_EXIT_DISC **не следует** агрегировать.

Если агрегированный маршрут имеет сегмент AS_SET в качестве первого элемента атрибута AS_PATH, маршрутизатору, от которого исходит маршрут, **не следует** анонсировать с этим маршрутом атрибут MULTI_EXIT_DISC.

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

NEXT_HOP

При агрегировании маршрутов с разными атрибутами NEXT_HOP в атрибуте NEXT_HOP агрегированного маршрута **следует** указывать интерфейс узла BGP, выполняющего агрегирование.

Атрибут ORIGIN

Если хотя бы один из агрегируемых маршрутов имеет ORIGIN = INCOMPLETE, для объединенного 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 = IGP.

Атрибут AS_PATH

Если агрегируемые маршруты имеют идентичные атрибуты AS_PATH, объединенный 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, для агрегированного атрибута AS_PATH **следует** обеспечить выполнение всех перечисленных ниже требований:

- всем парам типа AS_SEQUENCE агрегированного AS_PATH **следует** присутствовать в каждом атрибуте AS_PATH исходного набора агрегируемых маршрутов;
- всем парам типа AS_SET агрегированного AS_PATH **следует** присутствовать хотя бы в одном атрибуте AS_PATH исходного набора (возможно, 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 одной паре типа AS_SET **не следует** появляться в агрегированном AS_PATH более одного раза;
- множество пар типа AS_SEQUENCE с одинаковыми значениями может присутствовать в агрегированном AS_PATH только по соседству с другой однотипной парой с совпадающим значением.

Разработчики могут выбирать любой алгоритм, который обеспечивает соответствие приведенным правилам. Соответствующей требованиям этого документа реализации **следует** поддерживать по крайней мере описанный ниже алгоритм для выполнения всех приведенных выше требований:

- определить наиболее длинную последовательность лидирующих пар (как описано выше), присутствующую в атрибутах AS_PATH всех объединяемых маршрутов и сделать ее лидирующей в AS_PATH объединенного атрибута;
- установить для оставшихся пар из атрибутов AS_PATH объединяемых маршрутов тип AS_SET и присоединить их в конце агрегированного атрибута AS_PATH;
- если объединенный атрибут AS_PATH содержит несколько одинаковых пар (независимо от типа), лишние (все, кроме одной) пары типа AS_SET следует удалить из объединенного атрибута AS_PATH;
- для каждых двух смежных пар в агрегированном AS_PATH следует произвести операцию их слияния, если пары имеют одинаковый тип и размер сегмента не будет превышать 255.

В приложении F (параграф F.6) представлен другой алгоритм, соответствующий условиям и допускающий более сложную конфигурационную политику.

Атрибут ATOMIC_AGGREGATE

Если хотя бы один из объединяемых маршрутов имеет атрибут ATOMIC_AGGREGATE, в объединенный маршрут также **следует** включать этот атрибут.

Атрибут AGGREGATOR

Любые атрибуты AGGREGATOR из агрегируемых маршрутов **недопустимо** включать в агрегированный маршрут. Выполняющий агрегирование узел BGP **может** включить в маршрут новый атрибут AGGREGATOR (см. параграф 5.1.7).

9.3. Критерии выбора маршрута

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

- если локальная AS присутствует в пути нового маршрута, этот маршрут не может считаться лучше какого-либо из имеющихся путей (предполагается, что узел принимает такие маршруты); нарушение этого правила ведет к возникновению маршрутных петель;
- для обеспечения эффективной распределенной обработки следует выбирать только маршруты, представляющиеся стабильными. Таким образом, AS **следует** избегать применения нестабильных маршрутов и **не следует** вносить скороспелых спонтанных изменений при выборе путей. Трактовка слов «нестабильный» и «скороспелый» в предыдущем предложении требует некоторого опыта, но, в принципе, достаточно понятна. Нестабильные маршруты могут быть “оштрафованы” (например, с использованием процедур, описанных в документе [RFC2439]).

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 указывает для таких маршрутов уровень предпочтения (например, в соответствии с локальной политикой), используя для этого Decision Process (см. параграф 9.1). Эти маршруты **могут** также рассылаться другим узлам BGP в локальной AS, как часть процесса обновления (см. параграф 9.2). Решение о целесообразности рассылки полученной из других источников информации внутри AS с использованием BGP зависит от используемой в AS среды (например, типа IGP) и его **следует** задавать на уровне конфигурации.

10. Таймеры BGP

BGP поддерживает пять таймеров - ConnectRetryTimer (см. главу 8), HoldTimer (см. параграф 4.2), KeepaliveTimer (см. главу 8), MinASOriginationIntervalTimer (см. параграф 9.2.1.2) и MinRouteAdvertisementIntervalTimer (см. параграф 9.2.1.1).

Могут также поддерживаться два дополнительных таймера – DelayOpenTimer и IdleHoldTimer (см. главу 8). Использование этих таймеров описано в главе 8. Полное описание работы этих дополнительных таймеров выходит за пределы данного документа.

Параметр ConnectRetryTime является обязательным атрибутом FSM и хранит начальное значение для таймера ConnectRetryTimer. Предлагается по умолчанию устанавливать значение ConnectRetryTime равным 120 секундам.

Параметр HoldTime является обязательным атрибутом FSM и сохраняет начальное значение таймера HoldTimer. Предлагается по умолчанию использовать для HoldTime значение 90 секунд.

На некоторых этапах (см. главу 8) для HoldTimer устанавливается большое значение. Предлагается в качестве такого значения устанавливать 4 минуты.

Параметр KeepaliveTime является обязательным атрибутом FSM и сохраняет начальное значение таймера KeepaliveTimer. По умолчанию предлагается устанавливать для KeepaliveTime значение 1/3 HoldTime.

Для таймера MinASOriginationIntervalTimer предлагается по умолчанию использовать значение 15 секунд.

Для таймера MinRouteAdvertisementIntervalTimer предлагается устанавливать значение 30 секунд на соединениях EBGP.

Для таймера MinRouteAdvertisementIntervalTimer предлагается устанавливать значение 5 секунд на соединениях IBGP.

Реализация BGP **должна** обеспечивать возможность установки значения HoldTimer с помощью конфигурационного параметра независимо для каждого партнера и **может** обеспечивать возможность выбора значений для других таймеров.

Для снижения вероятности возникновения пиков при распространении сообщений BGP данным узлом **следует** использовать флуктуации (jitter) для таймеров, связанных с MinASOriginationIntervalTimer, KeepaliveTimer, MinRouteAdvertisementIntervalTimer и ConnectRetryTimer. Данный узел BGP **может** использовать одинаковые флуктуации для каждого из этих таймеров независимо от адресатов передаваемых обновлений (т. е., флуктуации не требуется делать независимыми для каждого партнера).

Предлагаемую по умолчанию величину флуктуаций **следует** определять путем умножения базового значения соответствующего таймера на случайное значение из диапазона 0,75 — 1,0. При каждой установке таймера **следует** выбирать новое случайное значение. Диапазон флуктуаций **может** быть настраиваемым.

Приложение A. Сравнение с RFC 1771

В настоящем документе имеется множество редакторских правок спецификации [RFC1771] (слишком много для перечисления).

Ниже приводится список технических изменений:

Внесены изменения, связанные с использованием TCP MD5 [RFC2385], BGP Route Reflectors [RFC2796], BGP Confederations [RFC3065] и BGP Route Refresh [RFC2918].

Разъяснено использование поля BGP Identifier в атрибуте AGGREGATOR.

Процедуры задания верхней границы для числа префиксов, которые узел BGP будет принимать от партнера.

Возможность включать более одного экземпляра своей AS в атрибут AS_PATH для управления картиной трафика между AS.

Разъяснены различные типы NEXT_HOP.

Разъяснено использование атрибута ATOMIC_AGGREGATE.

Соотношения между ближайшим маршрутизатором (immediate next hop) и следующим маршрутизатором, указанным атрибутом пути NEXT_HOP.

Разъяснены процедуры “отбрасывания лишнего” (tie-breaking).

Разъяснены требования по частоте анонсирования маршрутов.

Отменено использование дополнительного параметра типа 1 (Authentication Information).

Отменен субкод 7 (AS Routing Loop) для ошибок в сообщениях UPDATE.

Отменен субкод 5 (Authentication Failure) для ошибок в сообщениях OPEN.

Отменено использование поля Marker для аутентификации.

Реализация **должна** поддерживать механизм TCP MD5 [RFC2385] для аутентификации.

Разъяснена работа BGP FSM.

Приложение B. Сравнение с RFC 1267

Все изменения, перечисленные в Приложении A, а также указанные ниже изменения.

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, связанной с префиксами IP. Такое расширение позволяет BGP-4 поддерживать предложенную схему supernet [RFC1518, RFC1519].

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 выполнили агрегирование.

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

Приложение C. Сравнение с RFC 1163

Все изменения, перечисленные в Приложениях A и B, а также указанные ниже изменения.

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).

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 D. Сравнение с RFC 1105

Все изменения, перечисленные в Приложениях A, B и C, а также указанные ниже изменения.

Потребовалось внесение незначительных изменений в машину конечных состояний RFC1105 для согласования с пользовательским интерфейсом TCP в системах BSD версии 4.3.

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

Внесен ряд изменений в формат сообщений RFC1105:

1. Поле Hold Time было удалено из заголовка BGP и включено в сообщение OPEN.
2. Поле номера версии было удалено из заголовка BGP и включено в сообщение OPEN.
3. Из сообщений OPEN было удалено поле Link Type.
4. Вместо подтверждений OPEN CONFIRM используются сообщения KEEPALIVE.
5. Существенно изменен формат сообщений UPDATE, добавлены новые поля для поддержки множества атрибутов пути.
6. 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 E. 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 установку поля DSCP [RFC2474] для соединений TCP, транспортные соединения для BGP **следует** открывать с битами 0-2 поля DSCP, имеющими двоичное значение 110.

Реализация **должна** поддерживать опцию TCP MD5 [RFC2385].

Приложение F. Рекомендации для разработчиков

В этом приложении даются некоторые рекомендации разработчикам.

Приложение F.1. Множество префиксов сетей в одном сообщении

Протокол BGP позволяет указывать в одном сообщении множество адресных префиксов с одинаковыми атрибутами пути. Настоятельно рекомендуется использовать эту возможность. Передача сообщений с единственным префиксом существенно повышает нагрузку на получателя. В результате передачи множества сообщений растет не только нагрузка на системы, но и издержки при сканировании таблиц маршрутизации для передачи обновлений партнерам BGP и другим протоколам маршрутизации (увеличивается и объем передаваемых обновлений).

Одним из способов создания сообщений с множеством префиксов для каждого набора атрибутов пути из таблицы маршрутизации, не организованной по наборам атрибутов, является создание множества сообщений при сканировании таблицы. При обработке каждого префикса создается сообщение для связанного набора атрибутов пути, если оно не существует и в это сообщение добавляется новый префикс. Если сообщение уже создано, новый префикс просто добавляется в конец этого сообщения. Если в сообщение уже нельзя добавить новый префикс по соображениям размера, имеющееся сообщение передается, а для префикса создается 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 набором атрибутов пути. В этом случае сообщение может содержать столько префиксов, сколько позволяет ограничение на размер сообщений BGP (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 маршрутизации. Недостатком этого способа является увеличение задержек при распространении маршрутной информации. Выбор минимального интервала обновлений, который незначительно превышает время обработки множества сообщений, позволяет минимизировать эту задержку. Наилучшим решением будет просмотр всех полученных сообщений до передачи обновлений.

Приложение F.2. Снижение числа переключений маршрутов

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

Приложение F.3. Упорядочение атрибутов пути

Реализации, комбинирующие обновления (как описано в параграфе 6.1), могут предпочесть просмотр всех атрибутов пути, представленных в определенном 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. Такая оптимизация не является обязательной.

Приложение F.4. Сортировка AS_SET

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

Приложение F.5. Контроль за согласованием версий

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

Приложение F.6. 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 задает 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 выполняется любое из приведенных ниже условий:
   - X предшествует Y в обоих атрибутах AS_PATH;
   - 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) из обоих атрибутов объединяются в сегмент пути AS_SET. Этот сегмент пути помещается в агрегированном атрибуте между двумя последовательными AS, идентифицированными в пункте (a).

c) Для каждых из двух смежных пар в агрегированном AS_PATH (если они имеют одинаковый тип) выполняется слияние, если оно не будет приводить к генерации сегмента пути размером более 255.

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.

Вопросы безопасности

Реализация BGP **должна** поддерживать механизм аутентификации, определенный в RFC 2385 [RFC2385]. Аутентификация на основе этого механизма может осуществляться независимо для каждого партнера.

BGP использует протокол TCP для организации надежного обмена трафиком между маршрутизаторами-партнерами. Для обеспечения целостности соединений и аутентификации источников данных в соединениях между парами узлов спецификация BGP задает использование механизма, определенного в RFC 2385. Этот механизм предназначен для детектирования и предотвращения активного перехвата (wiretapping attacks) данных из соединений TCP между маршрутизаторами. В отсутствие такого рода механизмов обеспечения безопасности атакующие могут разрывать соединения TCP и/или маскироваться под легитимные партнерские маршрутизаторы. Поскольку определенный в RFC механизм не обеспечивает аутентификацию партнеров, протокольные соединения могут служить объектом некоторых атак с повторным использованием перехваченных ранее данных (replay attack), которые не будут детектироваться уровнем TCP. Такие атаки могут приводить к доставке (от уровня TCP) “испорченных” или “подмененных” сообщений BGP.

Механизм, определенный в RFC 2385, добавляет к обычным контрольным суммам TCP 16-байтовый код аутентификации сообщения (MAC[1]) который рассчитывается на основе тех же данных, что и контрольная сумма TCP. Расчет MAC основан на использовании необратимой хэш-функции (MD5) и закрытых ключей. Ключ известен паре маршрутизаторов-партнеров и используется для генерации значений MAC, которые не могут быть вычислены атакующим без знания ключа. Соответствующие спецификации реализации протокола должны поддерживать этот механизм и позволять администратору активизировать его использование независимо для каждого партнера.

RFC 2385 не задает механизмов поддержки ключей (например, их генерации, распространения и замены), используемых для расчета MAC. Документ RFC 3562 [RFC3562] (он имеет статус информационного) содержит некоторые рекомендации в этом направлении с обоснованием приведенных рекомендаций. В документе отмечается, что следует использовать разные ключи для связи с каждым защищенным партнером. Если один ключ используется для множества партнеров, это может привести к снижению уровня безопасности (например, за счет того, что при компрометации одного маршрутизатора становятся известными ключи, используемые для других маршрутизаторов).

Используемые для расчета MAC ключи следует периодически заменять для минимизации возможности компрометации ключа или успешной криптоаналитической атаки. В RFC 3562 предлагается устанавливать крипто-период (срок действия ключа) не более 90 дней. Более частая смена ключей снижает вероятность успеха атак с повторным использованием перехваченных данных. Однако отсутствие стандартного механизма эффективной координированной замены ключа, используемого парой маршрутизаторов, не позволяет надеяться, что реализации BGP-4, соответствующие данной спецификации, будут поддерживать такую частоту смены ключей.

[1] message authentication code

Очевидно, что ключи следует выбирать так, чтобы атакующему было сложно угадать или подобрать ключ. Описанные в RFC 1750 методы генерации случайных чисел обеспечивают руководство по созданию значений, которые могут использоваться в качестве ключей. RFC 2385 предлагает разработчикам использовать ключи, представляющие собой строки печатных символов ASCII размером 80 байтов или меньше. В RFC 3562 предлагается в таком контексте использовать ключи размером от 12 до 24 байтов, состоящие из случайных (псевдослучайных) битов. Это полностью совместимо с предложениями для аналогичных алгоритмов MAC, которые обычно используют ключи размером от 16 до 20 байтов. В части обеспечения достаточного уровня случайности при использовании ключей минимальной длины в RFC 3562 также отмечается,что типичная срока текста ACSII будет близка к верхней границе диапазона длины ключей, заданного в RFC 2385.

Анализ уязвимостей протокола BGP приводится в документе [RFC4272].

Согласование с IANA

Все сообщения BGP содержат 8-битовые идентификаторы типа сообщения, для которых агентство IANA создало и поддерживает реестр “BGP Message Types[1]”. В данном документе определены следующие типы сообщений:

Имя	*Значение*	*Определение*
OPEN	1	См. параграф 4.2.
UPDATE	2	См. параграф 4.3.
NOTIFICATION	3	См. параграф 4.5.
KEEPALIVE	4	См. параграф 4.4.

Выделение новых значений для типов сообщений происходит на основе процесса стандартизации (Standards Action), определенного в [RFC2434], или путем “Заблаговременного выделения агентством IANA[2]”, как описано в [RFC4020]. Типы сообщений задаются именем и числовым идентификатором.

Сообщения BGP UPDATE могут содержать один или множество атрибутов пути (Path Attribute), каждый из которых включает 8-битовый код типа (Attribute Type Code). Агентство IANA поддерживает реестр таких кодов, названный "BGP Path Attributes[3]". В этом документе определяются следующие типы атрибутов пути (Path Attributes Type Code):

Имя	*Значение*	*Определение*
ORIGIN	1	См. параграф 5.1.1.
AS_PATH	2	См. параграф 5.1.2.
NEXT_HOP	3	См. параграф 5.1.3.
MULTI_EXIT_DISC	4	См. параграф 5.1.4.
LOCAL_PREF	5	См. параграф 5.1.5.
ATOMIC_AGGREGATE	6	См. параграф 5.1.6.
AGGREGATOR	7	См. параграф 5.1.7.

Выделение новых значений для кодов атрибутов пути происходит на основе процесса стандартизации (Standards Action), определенного в [RFC2434], или путем “Заблаговременного выделения агентством IANA”, как описано в [RFC4020]. Типы атрибутов задаются именем и числовым идентификатором.

Сообщения BGP NOTIFICATION содержат 8-битовые значения кода ошибки (Error Code), для которых агентство IANA создало и поддерживает реестр "BGP Error Codes[4]". В этом документе определены следующие коды ошибок:

Имя	*Значение*	*Определение*
Message Header Error	1	См. параграф 6.1.
OPEN Message Error	2	См. параграф 6.2.
UPDATE Message Error	3	См. параграф 6.3.
Hold Timer Expired	4	См. параграф 6.5.
Finite State Machine Error	5	См. параграф 6.6.
Cease	6	См. параграф 6.7.

Выделение новых значений для кодов ошибок происходит на основе процесса стандартизации (Standards Action), определенного в [RFC2434], или путем “Заблаговременного выделения агентством IANA”, как описано в [RFC4020]. Коды ошибок задаются именем и числовым идентификатором.

Сообщения BGP NOTIFICATION содержат 8-битовые значения субкода ошибки (Error Subcode) и каждое значение субкода определяется в контексте соответствующего кода ошибки (Error Code) и, таким образом, является уникальным только в этом контексте.

Агентство IANA создало и поддерживает набор реестров "Error Subcodes[1]", в котором для каждого кода ошибки BGP имеется отдельный реестр. Выделение новых значений для субкодов ошибок происходит на основе процесса

[1]Типы сообщений BGP. Реестр доступен по адресу http://www.iana.org/assignments/bgp-parameters. *Прим. перев.*
[2]Early IANA Allocation
[3]Атрибуты путей BGP. Реестр доступен по адресу http://www.iana.org/assignments/bgp-parameters. *Прим. перев.*
[4]Коды ошибок BGP. Реестр доступен по адресу http://www.iana.org/assignments/bgp-parameters. *Прим. перев.*
[1]Субкоды ошибок BGP. Реестр доступен по адресу http://www.iana.org/assignments/bgp-parameters. *Прим. перев.*

стандартизации (Standards Action), определенного в [RFC2434], или путем “Заблаговременного выделения агентством IANA”, как описано в [RFC4020]. Субкоды ошибок задаются именем и числовым идентификатором.

В этом документе определяются следующие субкоды для ошибок в заголовках сообщений (Message Header Error):

Имя	*Значение*	*Определение*
Connection Not Synchronized	1	См. параграф 6.1.
Bad Message Length	2	См. параграф 6.1.
Bad Message Type	3	См. параграф 6.1.

В этом документе определяются следующие субкоды для ошибок в сообщениях OPEN (OPEN Message Error):

Имя	*Значение*	*Определение*
Unsupported Version Number	1	См. параграф 6.2.
Bad Peer AS	2	См. параграф 6.2.
Bad BGP Identifier	3	См. параграф 6.2.
Unsupported Optional Parameter	4	См. параграф 6.2.
[отменено]	5	См. Приложение A.
Unacceptable Hold Time	6	См. параграф 6.2.

В этом документе определяются следующие субкоды для ошибок в сообщениях UPDATE (UPDATE Message Error):

Имя	*Значение*	*Определение*
Malformed Attribute List	1	См. параграф 6.3.
Unrecognized Well-known Attribute	2	См. параграф 6.3.
Missing Well-known Attribute	3	См. параграф 6.3.
Attribute Flags Error	4	См. параграф 6.3.
Attribute Length Error	5	
Invalid ORIGIN Attribute	6	См. параграф 6.3.
[отменено]	7	См. Приложение A.
Invalid NEXT_HOP Attribute	8	
Optional Attribute Error	9	
Invalid Network Field	10	
Malformed AS_PATH	11	

Нормативные документы

[RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791[1], September 1981.

[RFC793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793[1], September 1981.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119[1], March 1997.

[RFC2385] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385[1], August 1998.

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

Дополнительная литература

[RFC904] Mills, D., "Exterior Gateway Protocol formal specification", RFC 904, April 1984.

[RFC1092] Rekhter, J., "EGP and policy based routing in the new NSFNET backbone", RFC 1092, February 1989.

[RFC1093] Braun, H., "NSFNET routing architecture", RFC 1093, February 1989.

[RFC1105] Lougheed, K. and Y. Rekhter, "Border Gateway Protocol (BGP)", RFC 1105, June 1989.

[RFC1163] Lougheed, K. and Y. Rekhter, "Border Gateway Protocol (BGP)", RFC 1163, June 1990.

[RFC1267] Lougheed, K. and Y. Rekhter, "Border Gateway Protocol 3 (BGP-3)", RFC 1267, October 1991.

[RFC1771] Rekhter, Y. and T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771[1], March 1995.

[RFC1772] Rekhter, Y. and P. Gross, "Application of the Border Gateway Protocol in the Internet", RFC 1772[1], March 1995.

[RFC1518] Rekhter, Y. and T. Li, "An Architecture for IP Address Allocation with CIDR", RFC 1518[1], September 1993.

[RFC1519] Fuller, V., Li, T., Yu, J., and K. Varadhan, "Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy", RFC 1519[1], September 1993.

[1]На сайте www.protocols.ru имеется перевод этого документа на русский язык. *Прим. перев.*

[RFC1930] Hawkinson, J. and T. Bates, "Guidelines for creation, selection, and registration of an Autonomous System (AS)", BCP 6, RFC 1930[1], March 1996.

[RFC1997] Chandra, R., Traina, P., and T. Li, "BGP Communities Attribute", RFC 1997[1], August 1996.

[RFC2439] Villamizar, C., Chandra, R., and R. Govindan, "BGP Route Flap Damping", RFC 2439, November 1998.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474[1], December 1998.

[RFC2796] Bates, T., Chandra, R., and E. Chen, "BGP Route Reflection - An Alternative to Full Mesh IBGP", RFC 2796[1], April 2000.

[RFC2858] Bates, T., Rekhter, Y., Chandra, R., and D. Katz, "Multiprotocol Extensions for BGP-4", RFC 2858[1], June 2000.

[RFC3392] Chandra, R. and J. Scudder, "Capabilities Advertisement with BGP-4", RFC 3392[2], November 2002.

[RFC2918] Chen, E., "Route Refresh Capability for BGP-4", RFC 2918[1], September 2000.

[RFC3065] Traina, P., McPherson, D., and J. Scudder, "Autonomous System Confederations for BGP", RFC 3065[1], February 2001.

[RFC3562] Leech, M., "Key Management Considerations for the TCP MD5 Signature Option", RFC 3562[1], July 2003.

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

[RFC4272] Murphy, S., "BGP Security Vulnerabilities Analysis", RFC 4272[1], January 2006

[RFC4020] Kompella, K. and A. Zinin, "Early IANA Allocation of Standards Track Code Points", BCP 100, RFC 4020[1], February 2005.

Адреса редакторов

Yakov Rekhter

Juniper Networks

EMail: yakov@juniper.net

Tony Li

EMail: tony.li@tony.li

Susan Hares

NextHop Technologies, Inc.

825 Victors Way

Ann Arbor, MI 48108

Phone: (734)222-1610

EMail: skh@nexthop.com

Перевод на русский язык

Николай Малых

nmalykh@protocols.ru

Полное заявление авторских прав

Интеллектуальная собственность

Подтверждение

Финансирование функций RFC Editor обеспечено IETF Administrative Support Activity (IASA).

[1]На сайте www.protocols.ru имеется перевод этого документа на русский язык. *Прим. перев.*

[1]Этот документ устарел и заменен RFC 4760. Перевод документов имеется на сайте www.protocols.ru. *Прим. перев.*

[2]Этот документ устарел и заменен RFC 5492. Перевод документов имеется на сайте www.protocols.ru. *Прим. перев.*