Протокол BGP

1. Задача внешней маршрутизации
2. Внутренний BGP, маршрутные серверы и атрибут NEXT_HOP
3. Атрибуты пути (Path Attributes)
3.1. ORIGIN
3.2. AS_PATH
3.3. NEXT_HOP
3.4. MULTI_EXIT_DISC
3.5. LOCAL_PREF
3.6. Атрибуты агрегирования
4. Обработка маршрутной информации (Decision Process) и маршрутные политики
4.1. Decision Process
4.2. Конфликты маршрутных политик
4.3. Формулировка маршрутных политик
5. Реализация BGP
1. Типы BGP-сообщений
2. Формат BGP-сообщения
3. Формат сообщения OPEN
4. Формат сообщения KEEPALIVE
5. Формат сообщения UPDATE
6. Формат сообщения NOTIFICATION

Источник -
http://telenetwork.narod.ru/books/cisco/Mamaev/telecomtech/bgp.html#7.3.6

Протокол BGP

1. Задача внешеней маршрутизации

В предыдущих разделах мы рассмотрели протоколы маршрутизации, работающие внутри автономных систем. Задача следующего уровня – построение маршрутов между сетями, принадлежащими разным автономным системам.

Рассматриваемый на этом уровне, Интернет представляет собой множество автономных систем, произвольным образом соединенных между собой (рис. 7.1.1). Внутреннее строение автономных систем скрыто, известны только адреса IP-сетей, составляющих АС.

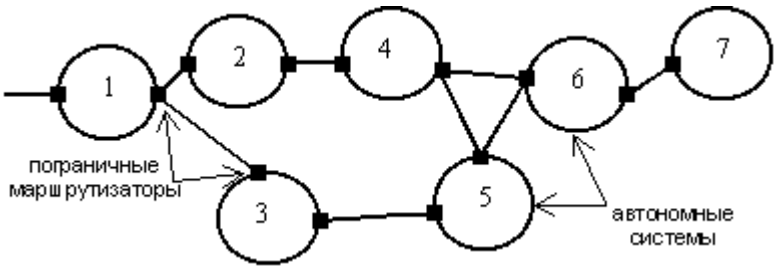

Рис. 1. Автономные системы

Автономные системы соединяются с помощью *пограничных маршрутизаторов*. Связи между маршрутизаторами могут иметь разную физическую природу: например, это может быть сеть Ethernet, к которой подключено сразу несколько пограничных маршрутизаторов разных автономных систем, выделенный канал типа "точка-точка" между двумя маршрутизаторами и т.п. Часто сама связующая сеть не принадлежит ни к одной автономной системе, в таком случае она называется *демилитаризованной зоной (DMZ)*.

В зависимости от своего расположения в общей структуре Интернет автономная система может быть *тупиковой (stub)* – имеющей связь только с одной АС (АС7 на рис. 7.1.1), – или *многопортовой (multihomed)*, т.е. имеющей связи с несколькими АС. Если административная политика автономной системы позволяет передавать через свои сети транзитный трафик других АС, то такую автономную систему можно назвать *транзитной (transit).*

Пограничный маршрутизатор сообщает связанным с ним другим пограничным маршрутизаторам, какие сети каких автономных систем достижимы через него. Обмен подобной информацией позволяет пограничным маршрутизаторам занести в таблицу маршрутов записи о сетях, находящихся в других АС. При необходимости эта информация потом распространяется внутри своей автономной системы с помощью протоколов внутренней маршрутизации (см., например, *внешние маршруты* в OSPF) с тем, чтобы обеспечить внешнюю коннективность своей АС.

Задачей этого пункта является обсуждение протокола обмена маршрутной информацией между пограничными маршрутизаторами.

Принципиальным отличием внешней маршрутизации от внутренней является наличие маршрутной политики, то есть при расчете маршрута рассматривается не столько метрика, сколько политические и экономические соображения. Это обстоятельство не позволяет адаптировать под задачу внешней маршрутизации готовые протоколы внутренней маршрутизации, просто применив их к графу автономных систем, как раньше они применялись к графу сетей. По той же причине существующие подходы – дистанционно-векторный и состояния связей – непригодны для решения поставленной задачи.

Например, рассмотрим систему маршрутизаторов, аналогичную графу автономных систем, изображенному на рис.1. Алгоритм SPF гарантирует, что если узел (1) вычислил маршрут в узел (6) как (1)-(3)-(5)-(6), то узел (3) также будет использовать маршрут (3)-(5)-(6). Это происходит потому, что все узлы одинаково интерпретируют метрики связей и используют одни и те же математические процедуры для вычисления маршрутов. Однако если применять протокол состояния связей на уровне автономных систем может произойти следующее: АС1 установила, что оптимальный маршрут по соображениям пропускной способности сетей в АС6 выглядит 1-3-5-6. Но АС3 считает, что выгодней добираться в АС6 по маршруту 3-1-2-4-6, так как АС5 дорого берет за передачу транзитного трафика. В итоге, из-за того, что у каждой АС свое понятие метрики, то есть качества маршрута, происходит зацикливание.

Казалось бы, дистанционно-векторный подход мог бы решить проблему: АС3, не желающая использовать АС5 для транзита в АС6, просто не прислала бы в АС1 элемент вектора расстояний для АС6, следовательно первая система никогда не узнала бы о маршруте 1-3-5-6. Но с другой стороны при использовании дистанционно-векторного подхода получателю вектора расстояний неизвестно описание маршрута на всей его протяженности. Рассмотрим другую политическую ситуацию на примере того же рис. 1:

АС1 получает от АС3 вектор АС6=2 и направляет свой трафик в АС6 через АС3, не подозревая, что на этом маршруте располагается АС5, которая находится с АС1 в состоянии войны и, естественно, не пропускает транзитный трафик от и для АС1. В результате АС1, установив внешне безобидный маршрут в АС6, осталась без связи с этой автономной системой. Если бы АС1 получила полное описание маршрута, то она несомненно бы выбрала альтернативный вариант 1-2-4-6, однако в дистанционно-векторных протоколах такая информация не передается.

Для решения задачи внешней маршрутизации был разработан протокол BGP (*Border Gateway Protocol*). Используемая в настоящий момент версия этого протокола имеет номер 4, соответствующий стандарт – RFC-1771, 1772.

Общая схема работы BGP такова. BGP-маршрутизаторы соседних АС, решившие обмениваться маршрутной информацией, устанавливают между собой соединения по протоколу BGP и становятся *BGP-соседями* (*BGP-peers*).

Далее BGP использует подход под названием *path vector*, являющийся развитием дистанционно-векторного подхода. BGP-соседи рассылают (*анонсируют*, *advertise*) друг другу *векторы путей* (*path vectors)*. Вектор путей, в отличие от вектора расстояний, содержит не просто адрес сети и расстояние до нее, а адрес сети и список атрибутов (*path attributes*), описывающих различные характеристики маршрута от маршрутизатора-отправителя в указанную сеть. В дальнейшем для краткости мы будем называть набор данных, состоящих из адреса сети и атрибутов пути до этой сети, маршрутом в данную сеть.

Данных, содержащихся в атрибутах пути, должно быть достаточно, чтобы маршрутизатор-получатель, проанализировав их с точки зрения политики своей АС, мог принять решение о приемлемости или неприемлемости полученного маршрута.

Например, наиболее важным атрибутом маршрута является AS_PATH – список номеров автономных систем, через которые должна пройти дейтаграмма на пути в указанную сеть. Атрибут AS_PATH можно использовать для:

- обнаружения циклов – если номер одной и той же АС встречается в AS_PATH дважды;
- вычисления метрики маршрута – метрикой в данном случае является число АС, которые нужно пересечь;
- применения маршрутной политики – если AS_PATH содержит номера политически неприемлемых АС, то данный маршрут исключается из рассмотрения.

Анонсируя какой-нибудь маршрут, BGP-маршрутизатор добавляет в AS_PATH номер своей автономной системы.

Другие атрибуты будут рассмотрены в п. 3.

2. Внутренний BGP, маршрутные серверы и атрибут NEXT_HOP

Очевидно, что BGP-маршрутизаторы, находящиеся в одной АС, также должны обмениваться между собой маршрутной информацией. Это необходимо для согласованного отбора внешних маршрутов в соответствии с политикой данной АС и для передачи транзитных маршрутов через автономную систему. Такой обмен производится также по протоколу BGP, который в этом случае часто называется *IBGP* (*Internal BGP*),

(соответственно, протокол обмена маршрутами между маршрутзаторами разных АС обозначается *EBGP –External BGP*).

Отличие IBGP от EBGP состоит в том, что при объявлении маршрута BGP-соседу, находящемуся в той же самой АС, маршрутизатор не должен добавлять в AS_PATH номер своей автономной системы. Действительно, если номер АС будет добавлен, и сосед анонсирует этот маршрут далее (опять с добавлением номера той же АС), то одна и та же АС будет перечислена AS_PATH дважды, что расценивается как цикл.

Это очевидное правило влечет за собой интересное следствие: чтобы не возникло циклов, маршрутизатор не может анонсировать по IBGP маршрут, полученный также по IBGP, поскольку нет способов определить зацикливание при объявлении BGP-маршрутов внутри одной АС.

Следствием этого следствия является необходимость полного графа IBGP-соединений между пограничными маршрутизаторами одной автономной системы: то есть каждая пара маршрутизаторов должна устанавливать между собой соединение по протоколу IBGP. При этом возникает проблема большого числа соединений (порядка N^2, где N-число BGP-маршрутизаторов в АС). Для уменьшения числа соединений применяются различные решения: разбиение АС на конфедерации (подсистемы), применение серверов маршрутной информации и др.

Сервер маршрутной информации (аналог выделенного маршрутизатора в OSPF), обслуживающий группу BGP-маршрутизаторов, работает очень просто: он принимает маршрут от одного участника группы и рассылает его всем остальным. Таким образом, участникам группы нет необходимости устанавливать BGP-соединения попарно; вместо этого каждый участник устанавливает одно соединение с сервером. Группой маршрутизаторов могут быть, например, все BGP-маршрутизаторы данной АС, однако маршрутные серверы могут применяться для уменьшения числа соединений также и на внешних BGP-соединениях – в случае, когда в одной физической сети находится много BGP-маршрутизаторов из различных АС (например, в точке обмена трафиком между провайдерами, рис. 2.1.).

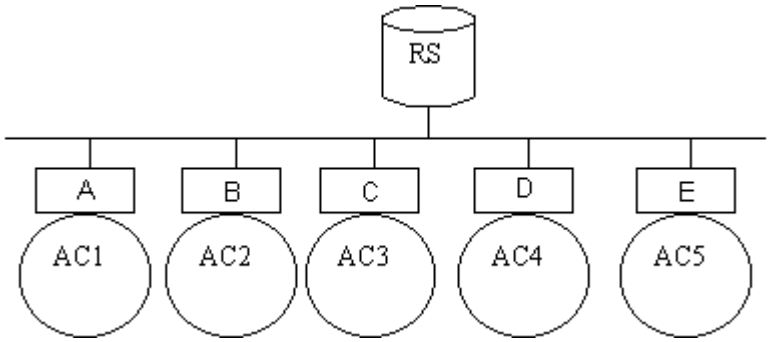

Рис. 2.1. Точка обмена трафиком (Internet Exchange Point)
A-E – пограничные BGP-маршрутизаторы, AC1-AC5 – сети автономных систем, RS – сервер маршрутной информации.

Следует особо отметить, что сервер маршрутной информации обслуживает только анонсы маршрутов, а не сам трафик по этим маршрутам. Например (рис 2.1), маршрутизатор А анонсирует серверу RS маршруты в с*е*ти АС1. Маршрутизатор Е получает информацию об этих маршрутах от сервера RS, но при этом в таблице маршрутов узла Е следующим маршрутизатором на пути в АС1 значится узел А, что абсолютно разумно, поскольку эти узлы могут передавать данные друг другу непосредственно. Исключение маршрутного

сервера из маршрута производится путем установки значения атрибута NEXT_HOP: анонсируя маршруты в сеть AC1, сервер RS указывает NEXT_HOP=A. Таким образом, маршрутизатор (например, E), получивший и принявший к использованию такой маршрут, будет пересылать данные, предназначенные для AC1, непосредственно маршрутизатору A.

Из вышесказанного можно сделать два важных вывода. Во-первых, узел, указанный как NEXT_HOP, должен быть достижим, то есть в таблице маршрутов маршрутизатора, принявшего маршрут с этим атрибутом, должна быть запись об узле NEXT_HOP или его сети. Если такой записи нет, то маршрутизатор должен забраковать полученный маршрут, потому что он не знает, как отправлять дейтаграммы к узлу NEXT_HOP.

Во-вторых, очевидно, что сервер маршрутной информации не является маршрутизатором (в соответствии с определением, данным в п. 1). То есть в общем случае узел, на котором работает модуль BGP, – не обязательно маршрутизатор. В технических документах этот факт подчеркивается тем, что для обозначения BGP-узла используется термин *BGP-speaker* (не *router*).

3. Атрибуты пути (Path Attributes)

Ниже перечислены все атрибуты пути, определенные для протокола BGP.

3.1. ORIGIN

ORIGIN (тип 1) – обязательный атрибут, указывающий источник информации о маршруте:

0 – IGP (информация о достижимости сети получена от протокола внутренней маршрутизации или введена администратором),

1 – EGP (информация о достижимости сети импортирована из устаревшего протокола EGP),

2 – INCOMPLETE (информация получена другим образом, например, RIP->OSPF->BGP или BGP->OSPF->BGP).

Атрибут ORIGIN вставляется маршрутизатором, который генерирует информацию о маршруте, и при последующем анонсировании маршрута другими маршрутизаторами не изменяется. Атрибут фактически определяет надежность источника информации о маршруте (наиболее надежный ORIGIN=0).

3.2. AS_PATH

AS_PATH (тип 2) – обязательный атрибут, содержащий список автономных систем, через которые должна пройти дейтаграмма на пути в указанную в маршруте сеть. AS_PATH представляет собой чередование сегментов двух типов: AS_SEQUENCE – упорядоченный список AC, и AS_SET – множество AC (последнее может возникнуть при агрегировании нескольких маршрутов со схожими, но не одинаковыми AS_PATH в один общий маршрут).

Каждый BGP-узел при анонсировании маршрута (за исключением IBGP-соединений) добавляет в AS_PATH номер своей АС. Возможно (в зависимости от политики) дополнительно добавляются номера других АС.

3.3. NEXT_HOP

NEXT_HOP (тип 3) – обязательный атрибут, указывающий адрес следующего BGP-маршрутизатора на пути в заявленную сеть (см. обсуждение в п. 2); может совпадать или не совпадать с адресом BGP-узла, анонсирующего маршрут. Указанный в NEXT_HOP маршрутизатор должен быть достижим для получателя данного маршрута. При передаче маршрута по IBGP NEXT_HOP не меняется.

3.4. MULTI_EXIT_DISC

MULTI_EXIT_DISC (тип 4) – необязательный атрибут, представляющий собой приоритет использования объявляющего маршрутизатора для достижения через него анонсируемой сети, то есть фактически это метрика маршрута с точки зрения анонсирующего маршрут BGP-узла. Имеет смысл не само значение, а разница значений, когда несколько маршрутизаторов одной АС объявляют о достижимости через себя одной и той же сети, предоставляя таким образом получателям несколько вариантов маршрутов в одну сеть. При *прочих равных условиях* дейтаграммы в объявляемую сеть будут посылаться через маршрутизатор, заявивший меньшее значение MULTI_EXIT_DISC.

Атрибут сохраняется при последующих объявлениях маршрута по IBGP, но не по EBGP.

3.5. LOCAL_PREF

LOCAL_PREF (тип 5) – необязательный атрибут, устанавливающий для данной АС приоритет данного маршрута среди всех маршрутов к заявленной сети, известных внутри АС. Атрибут вычисляется каждым пограничным маршрутизатором для каждого присланного ему по EBGP маршрута и потом распространяется вместе с этим маршрутом по IBGP в пределах данной АС. Способ вычисления значения атрибута определяется политикой приема маршрутов (по умолчанию берется во внимание только длина AS_PATH). LOCAL_PREF используется для согласованного между маршрутизаторами одной АС выбора маршрута из нескольких вариантов.

3.6. Атрибуты агрегирования

ATOMIC_AGGREGATE (тип 6) и **AGGREGATOR** (тип 7) – необязательные атрибуты, связанные с операциями агрегирования (объединения) нескольких маршрутов в один. Для более детального ознакомления с ними отсылаем читателей к документу RFC-1771.

ATOMIC_AGGREGATE это фактически флаг. Который говорит, что на пути произошла агрегация адресов АС, т.е. какая-то АС увидела общей префикс у всех адресов АС, которые стоят до нее. Тогда агрегирующая АС с помощью флага **ATOMIC_AGGREGATE** анонсирует себя, как proxy? А все АС за ней убирает из AC_Path. Такой механизм чреват циклами. Ведь теперь АС не может проверить есть она в AC_Path или нет. Чтобы избежать таких случаев в поле атрибута **AGGREGATE** перечисляется множество АС. Это не последовательность, а множество.

4. Обработка маршрутной информации (Decision Process) и маршрутные политики

4.1. Decision Process

Рассмотрим действия BGP-маршрутизатора при получении и анонсировании маршрута (рис. 4.1). Маршрутизатор использует три базы данных: Adj-RIBsIn, Loc-RIB и Adj-RIBsOut, в которых содержатся маршруты, соответственно, полученные от соседей, используемые самим маршрутизатором и объявляемые соседям. Также на маршрутизаторе сконфигурированы две политики: политика приема маршрутов (*accept policy*) и политика анонсирования маршрутов (*announce policy*). Для обработки маршрутов в базах данных в соответствии с имеющимися политиками маршрутизатор выполняет процедуру под названием *процесс отбора (decision process)*, описанную ниже.

Маршруты, полученные от BGP-соседей, помещаются в базу данных Adj-RIBsIn. В соответствии с политикой приема для каждого маршрута в Adj-RIBsIn вычисляется приоритет (это называется фазой 1 процесса отбора). Атрибут Loacl_pref. В результате этих действий некоторые маршруты могут быть отбракованы (признаны неприемлемыми).

Далее (фаза 2) для каждой сети из всех имеющихся (полученных и не отбракованных) вариантов выбирается маршрут с большим приоритетом. Результаты заносятся в базу Loc-RIB, откуда менеджер маршрутной таблицы IP-модуля может их взять для установки в таблицу маршрутов маршрутизатора и для экспорта во внутренний протокол маршрутизации с тем, чтобы и другие узлы автономной системы имели маршруты к внешним сетям. И наоборот, чтобы другие автономные системы имели маршруты к сетям данной АС, из таблиц протокола (протоколов) внутренней маршрутизации могут извлекаться номера сетей своей АС и заноситься в Loc-RIB.

Задача третьей фазы – отбор маршрутов для анонсирования (рассылки соседям). Из LocRIB выбираются маршруты, соответствующие политике анонсирования, и результат помещается в базу Adj-RIBsOut, содержимое которой и рассылается BGP-соседям. Возможно, что маршрутизатор имеет разные политики анонсирования для каждого соседа.

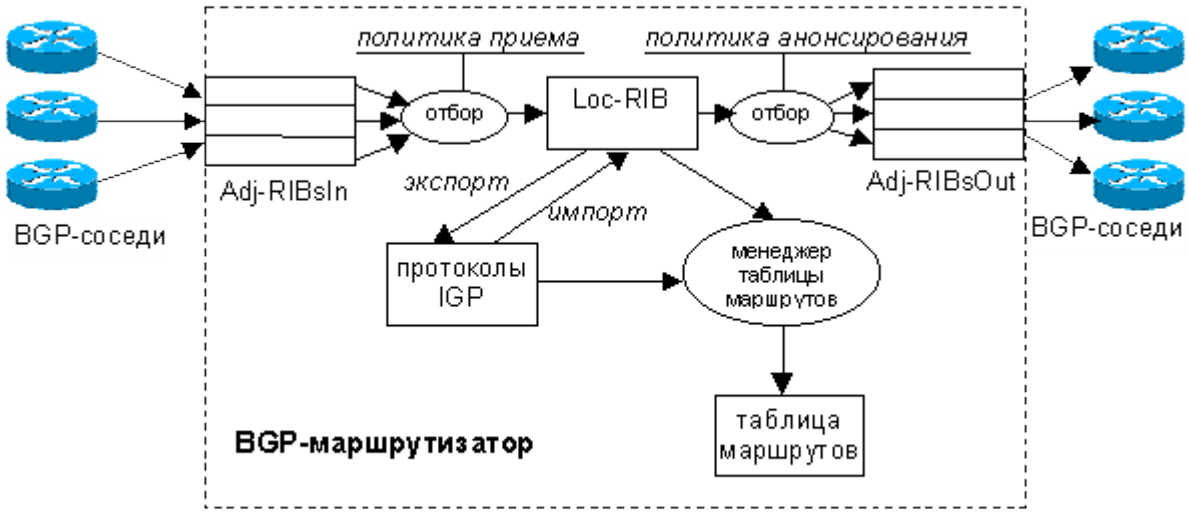

Рис. 4.1. Обработка маршрутной информации модулем BGP

Важным свойством процесса отбора является то, что BGP-маршрутизатор объявляет только те маршруты, которые он сам использует. Это обстоятельство является следствием природы IP-маршрутизации: при выборе маршрута для дейтаграммы учитывается только адрес получателя и никогда – адрес отправителя. Таким образом, если маршрутизатор сам использует один маршрут в сеть X, а соседу объявил другой, то дейтаграммы от соседа все равно будут пересылаться в сеть X по тому маршруту, который использует сам маршрутизатор, поскольку адрес отправителя при выборе маршрута IP-модулем не рассматривается.

4.2. Конфликты маршрутных политик

Рассмотрим пример (рис. 4.2).

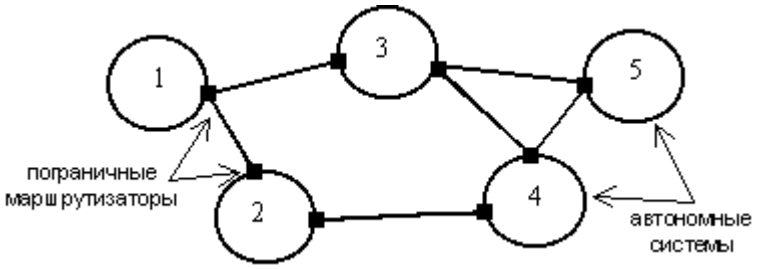

Рис. 4.2. Автономные системы

Предположим, автономная система 5 формирует маршрут до сетей автономной системы 1. Политическая ситуация такова, что АС3 берет с АС5 существенно б***о***льшие деньги за транзитный трафик, чем АС4 и АС2, поэтому АС5 формирует политику приема маршрутов, в соответствии с которой на маршруты, полученные от АС3, но ведущие в сети других АС (АС1 и АС2), назначается меньший приоритет, чем на маршруты в те же сети, но полученные от АС4. Таким образом, АС5 ожидает, что ее маршрутизатор установит маршрут в АС1 как 5-4-2-1 (естественно, что к сетям самой АС3 маршрут будет кратчайший: 5-3).

Однако выясняется, что маршрутизатор АС5 выбрал маршрут 5-3-1, то есть политика не действует, хотя все настройки в АС5 произведены верно. Дело в том, что к четвертой системе АС5 относится гораздо более дружелюбно и тарифицирует ее трафик по обычным расценкам, вследствие чего для АС4 нет разницы, как добираться в АС1: 4-3-1 или 4-2-1. АС4 по какой-то причине выбирает маршрут 4-3-1, и поскольку маршрутом 4-2-1 она не пользуется, она его и не объявляет. В результате у АС5 нет вариантов: эта автономная система просто не получает анонс маршрута 4-2-1 и вынуждена использовать маршрут 3-1.

Для решения проблемы АС5 должна обратиться к АС4 с просьбой установить политику приема маршрутов так, чтобы использовать маршрут 4-2-1. Однако, если АС3 тарифицирует трафик АС4 не просто по обычным тарифам, а со скидками, то для АС4 невыгодно вносить требуемые изменения, и АС5 остается перед выбором: либо дорого платить за трафик через АС3, либо не иметь связи с АС1.

Описанная ситуация представляет собой конфликт интересов, который не может быть разрешен программным обеспечением протокола BGP. Отметим, что в этом нет вины самого протокола; мы наблюдаем здесь побочный эффект технологии маршрутизации "только по адресу получателя", используемой протоколом IP.

4.3. Формулировка маршрутных политик

Способы описания маршрутных политик не являются частью протокола BGP и отличаются в различных реализациях BGP. Однако в любом случае политики базируются на критериях отбора маршрутов и модификации атрибутов маршрутов, попавших под критерии отбора. Модификация атрибутов маршрута в свою очередь влияет на приоритет этого маршрута при отборе нескольких альтернативных маршрутов во время фазы 2.

Для полного запрета принятия или объявления маршрута используется фильтрация, которую можно рассматривать как назначение низшего приоритета, не позволяющего использовать маршрут вообще.

Отбор маршрутов из базы Adj-RIBsIn (для реализации политики приема) может производиться, например, по следующим критериям:

- регулярное выражение для значения AS_PATH (частные случаи: номер конечной АС маршрута, АС соседа, от которого получен маршрут);
- адрес сети, в которую ведет маршрут;
- адрес соседа, приславшего информацию о маршруте;
- происхождение маршрута (атрибут ORIGIN).

К маршруту, удовлетворяющему установленному критерию, можно применить следующие политики:

- не принимать маршрут – удалить из Adj-RIBsIn (фильтрация);
- установить административный вес маршрута,
- установить значение атрибута LOCAL_PREF,
- установить маршрут в качестве маршрута по умолчанию.

Административный вес маршрута не является атрибутом BGP, он устанавливает внутренний приоритет маршрута на данном маршрутизаторе (в то время, как LOCAL_PREF устанавливает приоритет маршрута в рамках автономной системы).

Если (после выполнения фильтрации) в базе Adj-RIBsIn имеется несколько альтернативных маршрутов, ведущих в одну сеть назначения, то отбор лучшего из них производится фазой 2 по приведенным ниже критериям (на примере маршрутизаторов Cisco). Критерии последовательно применяются в указанном порядке, пока не останется единственный маршрут:

- наибольший административный вес;
- наибольшее значение LOCAL_PREF;
- кратчайший AS_PATH (маршрут, порожденный в локальной АС, имеет самый короткий – пустой – AS_PATH);
- наименьшее значение ORIGIN (IGP<EGP<INCOMPLETE);
- наименьшее значение MULTI_EXIT_DISC (отсутствующий MULTI_EXIT_DISC считается нулевым);
- маршрут, полученный по EBGP, против маршрута, полученного по IBGP;
- если все маршруты получены по IBGP, то выбирается маршрут через ближайшего соседа;
- маршрут, полученный от BGP-соседа с наименьшим идентификатором (IP-адресом).

Аналогично, отбор маршрутов в базу Adj-RIBsOut (для реализации политики объявлений) может производиться, например, по следующим критериям:

- регулярное выражение для значения AS_PATH (частные случаи: номер конечной АС маршрута, АС соседа, от которого получен маршрут);
- адрес сети, в которую ведет маршрут;
- адрес соседа, которому этот маршрут объявляется;
- происхождение маршрута (атрибут ORIGIN).

К маршруту, удовлетворяющему установленному критерию, можно применить следующие политики:

- не объявлять маршрут (фильтрация);
- MULTI_EXIT_DISC: не устанавливать, установить указанное значение, взять в качестве значения метрику маршрута из IGP;
- произвести агрегирование сетей в общий префикс;
- модифицировать AS_PATH указанным образом;
- заменить маршрут на default.

5. Реализация BGP

Пара BGP-соседей устанавливает между собой соединение по протоколу TCP, порт 179. Соседи, принадлежащие разным АС, должны быть доступны друг другу непосредственно; для соседей из одной АС такого ограничения нет, поскольку протокол внутренней маршрутизации обеспечит наличие всех необходимых маршрутов между узлами одной автономной системы.

Поток информации, которым обмениваются BGP-соседи по протоколу TCP, состоит из последовательности BGP-сообщений. Максимальная длина сообщения 4096 октетов, минимальная – 19. Имеется 4 типа сообщений.

5.1. Типы BGP-сообщений

OPEN – посылается после установления TCP-соединения. Ответом на OPEN является сообщение KEEPALIVE, если вторая сторона согласна стать BGP-соседом; иначе посылается сообщение NOTIFICATION с кодом, поясняющим причину отказа, и соединение разрывается.

KEEPALIVE – сообщение предназначено для подтверждения согласия установить соседские отношения, а также для мониторинга активности открытого соединения: для этого BGP-соседи обмениваются KEEPALIVE-сообщениями через определенные интервалы времени.

UPDATE – сообщение предназначено для анонсирования и отзыва маршрутов. После установления соединения с помощью сообщений UPDATE пересылаются все маршруты, которые маршрутизатор хочет объявить соседу (*full update*), после чего пересылаются только данные о добавленных или удаленных маршрутах по мере их появления (*partial update*).

NOTIFICATION – сообщение этого типа используется для информирования соседа о причине закрытия соединения. После отправления этого сообщения BGP-соединение закрывается.

5.2. Формат BGP-сообщения

Сообщение протокола BGP состоит из заголовка и тела. Заголовок имеет длину 19 октетов и состоит из следующих полей:

- 16 октетов - маркер: в сообщении OPEN всегда, и при работе без аутентификации - в других собщениях, заполнен единицами. Иначе содержит аутентификационную информацию. Сопутствующая функция маркера - повышение надежности выделения границы сообщения в потоке данных.
- 2 октета - длина сообщения в октетах, включая заголовок.
- 1 октет - тип сообщения:

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

5.3. Формат сообщения OPEN

Сообщение OPEN состоит из следующих полей:

- 19 октетов - заголовок с маркером, заполненным единицами.
- 1 октет - версия BGP (=4).
- 2 октета - номер своей АС.
- 2 октета - Hold Time - максимальное вермя (в секундах) между приходами сообщений KEEPALIVE, используемых для монитронига активности соединения; значение 0 - KEEPALIVE не посылаются вообще (мониторинг не производится); значения 1 и 2 не разрешаются. При получении сообщения маршрутизатор принимает для данного соединения значение Hold Time, равное минимуму из того, что он получил в сообщении OPEN, и значения, указанного в конфигурации маршрутизатора. Если сообщение KEEPALIVE не было получено в течении Hold Time секунд, соединение считается закрытым и вся связанная с ним маршрутная информация ликвидируется.
- 4 октета - идентификатор маршрутизатора (один из адресов интерфейсов).
- 1 октет - длина опции в октетах. Опции кодируются в виде: октет "Тип опции", октет "Длина опции в октетах", содержимое опции. Стандарт определяет одну опцию (тип 1) - "Определение типа аутентификации", содержимое которой состоит из октета "Тип аутентификации" и аутентификационных данных, интерпретация которых зависит от типа аутентификации. Предполагается, что на основании этой информации будет заполняться маркер заголовка.

5.4. Формат сообщения KEEPALIVE

Сообщение KEEPALIVE состоит только из BGP-заголовка.

5.5. Формат сообщения UPDATE

Сообщение UPDATE состоит из трех частей переменной длины: список недействительных маршрутов, список атрибутов и список сетей, к которым эти атрибуты относятся. Две последние части представляют собой собственно информацию о маршруте в указанные сети. (То есть, если маршруты в несколько разных сетей имеют одинаковые атрибуты, то они объединяются в одном сообщении UPDATE. Разумеется,

предварительно адреса сетей везде, где это возможно, агрегируются в общий префикс.) Как первая часть сообщения, так и две последние могут отсутствовать.

Формат сообщения:

- 19 октетов - заголовок,
- 2 октета - длина списка недействительных маршрутов L1;
- L1 октетов - список недействительных маршрутов;
- 2 октета - длина списка атрибутов L2;
- L2 октетов - список атрибутов для указанных ниже сетей;
- X октетов - список адресов сетей (Network Layer Reachability Information, NLRI).

X= Длина_всего_сообщения - 19(длина заголовка) - 4 - L1 - L2.

Список адресов сетей и список недействительных маршрутов представляют собой списки элементов, каждый элемент состоит из 2 частей:

- 1 октет - длина префикса L
- N октетов - префикс, где N - верхняя целая часть от L/8.

Например, префикс 10.0.0.0/8 представляется в виде двух октетов:

8	10

Префикс 172.16.192.0/19 представляется в виде 4 октетов:

19	172	16	192

Напомним, что формально префикс представляет собой адрес некой сети, а длина префикса интерпретируется также как длина сетевой маски. В реальном адресном пространстве префикс может соответствовать одной IP-сети или агрегировать в себе несколько IP-сетей.

Для отмены маршрута достаточно указать только адрес сети (префикс) назначения, по которому сосед найдет и удалит из своей базы все данные по этому маршруту. В одном сообщении может быть отменено несколько маршрутов.

Для объявления маршрута следует представить префикс назначения и список атрибутов пути для этого префикса. Только один список атрибутов передается в одном собщении, но указанные атрибуты могут относиться к нескольким префиксам, список которых приводится за списком атрибутов.

Список атрибутов представляет собой список элементов, каждый элемент состоит из 4 частей:

- 1 октет - флаги атрибута,
- 1 октет - тип атрибута,
- 1 или два октета, в зависимости от бита 3 флагов - длина данных атрибута L,
- L октетов, может быть 0, - данные атрибута.

Флаги атрибута:

- бит 0=1 - атрибут всеобщий (обязан обрабатываться любым BGP-процессом: например, ORIGIN, AS_PATH, NEXT_HOP, LOCAL_PREF, ATOMIC_AGGREGATE),
- бит 0=0 - атрибут дополнительный (BGP-процесс может проигнорировать этот атрибут: например, MULTI_EXIT_DISC, AGGREGATOR).
- бит 1=1 - для дополнительных атрибутов: атрибут транзитивный (должен передаваться при переобъявлении маршрута другому соседу, например, AGGREGATOR). Для прочих атрибутов бит 1 всегда установлен.
- бит 1=0 - Для дополнительных атрибутов: атрибут не транзитивный (не передается при переобъявлении маршрута другому соседу, например, MULTI_EXIT_DISC).
- бит 2=1 - (только дополнительных транзитивных атрибутов) какой-то из маршрутизаторов, через которые проходил атрибут, проигнорировал его,
- бит 2=0 - (только дополнительных транзитивных атрибутов) все маршрутизаторы по пути следования обработали атрибут. Для прочих атрибутов бит 2 всегда обнулен.
- бит 3=1 - поле "Длина данных атрибута" занимает 2 октета,
- бит 3=0 - поле "Длина данных атрибута" занимает 1 октет.

Длина и интерпертация данных атрибута зависит от типа атрибута.

- ORIGIN (тип 1) - 1 октет данных, содержащий значение атрибута (целое число 0, 1 или 2).
- ASPATH (тип 2) - данные атрибута состоят из списка элементов, каждый элемент состоит из 3 частей:
    - 1 октет - тип сегмента пути: AS_SEQUENCE (=2) или AS_SET (=1),
    - 1 октет - число N номеров АС в сегменте,
    - 2N октетов - список номеров АС этого сегмента пути (по два октета на номер).
- NEXT_HOP (тип 3) - 4 октета, содержащих значение атрибута (IP-адрес).
- MULTI_EXIT_DISC (тип 4) - 4 октета, содержащих значение атрибута (беззнаковое целое число).
- LOCAL_PREF (тип 5)- 4 октета, содержащих значение атрибута (беззнаковое целое число).
- ATOMIC_AGGRGATE (тип 6) - нет данных (значением атрибута является его присутствие).
- AGGREGATOR (тип 7) - 6 октетов, содержащих значение атрибута (2 октета - номер АС, 4 октета - IP-адрес).

5.6. Формат сообщения NOTIFICATION

Сообщение NOTIFICATION состоит из следующих полей:

- 1 октет - Код ошибки,
- 1 октет - Субкод ошибки,
- X октетов - данные, X=Длина_всего_сообщения - 19(длина заголовка) - 1 - 1. Количество и интерпретация данных зависят от кода и субкода ошибки.

Определены следующие коды ошибок:

1 - Ошибка заголовка
2 - Ошибка в сообщении OPEN
3 - Ошибка в сообщении UPDATE

4 - Истекло время ожидания сообщения KEEPALIVE
5 - Ошибка последовательности состояний
6 - Закрытие соединения по желанию участника