Internet Engineering Task Force (IETF)
Request for Comments: 5882
Category: Standards Track
ISSN: 2070-1721

D. Katz
D. Ward
Juniper Networks

June 2010

Generic Application of Bidirectional Forwarding Detection (BFD)

Базовое применение протокола BFD

Аннотация

Этот документ описывает базовое применение протокола обнаружения двухсторонней пересылки (Bidirectional Forwarding Detection или BFD).

Статус документа

Этот документ является проектом стандарта Internet (Internet Standards Track).

Документ подготовлен IETF[1] и содержит согласованный взгляд сообщества IETF. Документ обсуждался публично и одобрен для публикации IESG[2]. Дополнительная информация о стандартах Internet приведена в разделе 2 RFC 5741.

Информацию о текущем состоянии данного документа, обнаруженных ошибках и способах обратной связи можно найти по ссылке http://www.rfc-editor.org/info/rfc5882.

Оглавление

[1] Internet Engineering Task Force - комиссия по решению инженерных задач Internet.
[2] Internet Engineering Steering Group - комиссия по инженерным разработкам Internet.

1. Введение

Протокол BFD [BFD] предоставляет механизм обнаружения, который может применяться другими компонентами сети, когда их собственные механизмы проверки живучести слишком медленны, неуместны или отсутствуют. Применение BFD с конкретной инкапсуляцией описано в документах [BFD-1HOP] [BFD-MULTI] [BFD-MPLS]. По мере того, как полезность BFD становилась понятной, возникли призывы задать взаимодействие BFD с растущим списком сетевых функций. Вместо создания серии коротких документов по использованию BFD представляется разумным описать взаимодействие BFD с другими сетевыми функциями (клиентами BFD) в широком смысле.

Этот документ описывает базовое применение BFD. Конкретные протоколы рассматриваются для иллюстрации.

1.1. Используемые соглашения

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

2. Обзор

Спецификация BFD определяет протокол с простой и конкретной семантикой. Единственной целью протокола является проверка связности между парой систем для конкретного протокола передачи данных по определённому пути (путь может использовать любую технологию, иметь любую длину и относиться к любому уровню OSI). Оперативность обнаружения отказов путей может контролироваться с обеспечением компромисса между издержками протокола, загрузкой систем и временем обнаружения.

Протокол BFD **не** предназначен для прямого предоставления сведений о живучести протоколов управления (обычно эти протоколы имеют свои механизмы и причуды). Вместо этого протоколы управления могут пользоваться услугами BFD для информирования о своих операциях. BFD можно считать услугой на уровне, где протокол работает.

Сервисный интерфейс с BFD прост. Приложение предоставляет параметры сессии (адрес соседа, временные параметры, опции протокола), а BFD представляет состояние сессии, для которого наиболее интересны переходы в статус Up и из него. Предполагается запуск сессии BFD приложением, поскольку протокол BFD не имеет средств обнаружения (соседей).

Реализации **следует** создавать лишь одну сессию BFD для каждого пути протокола данных, независимо от числа использующих путь приложений. Нет никакого смысла создавать несколько сессий BFD для одной пары конечных точек. Если разным приложениям нужны свои параметры сессии, такой конфликт разрешается локально. При изменении параметров сессии BFD будет уведомлять все привязанные к сессии приложения.

BFD следует считать «консультантом» для протоколов или иных сетевых функций, с которыми протокол взаимодействует и которые используют свои механизмы реагирования на смену состояния. Взаимодействие происходит «на расстоянии вытянутой руки», что делает его простым и отвязанным от других функций. В частности, BFD не передаёт связанной с приложениями информации, отчасти из архитектурных соображений, отчасти потому, что BFD может иметь странные и непредсказуемые характеристики, что делает протокол непригодным в качестве транспорта.

Важно помнить, что взаимодействие между BFD и клиентскими приложениями практически не вызывает проблем функциональной совместимости, поскольку BFD выступает в качестве «консультанта» (подобно аппаратной сигнализации о пропадании луча в оптическом канале) и для реакции на события BFD применяются имеющиеся механизмы клиентских приложений. Фактически BFD может взаимодействовать лишь с одной системой из пары для клиентского приложения без каких-либо побочных эффектов.

3. Базовые взаимодействия между сессиями и клиентами BFD

Взаимодействие сессии BFD со связанными с ней клиентскими приложениями по большей части является вопросом реализации и выходит за рамки этого документа. Однако полезно описать некоторые механизмы, которые разработчики могут применять для продвижения полнофункциональных реализаций. Одним из способов моделирования этого взаимодействия является создание уровня адаптации между конечным автоматом BFD и клиентским приложением. Уровень адаптации будет знать детали внутренней реализации BFD и потребности клиентов.

3.1. Гистерезис состояния сессии

Сессия BFD может быть тесно связана со своими клиентскими приложениями, например, любой выход из состояния Up может вызывать сигнал, заставляющий клиента выполнить действия по преодолению отказа. Одна в некоторых случаях это может быть не лучшим решением.

Реализации могут скрывать быстрые переходы Up-Down-Up сессий BFD от своих клиентов. Это полезно, например, для предотвращения ненужных перезапусков сложных протокольных механизмов.

Таким образом, система **может** не уведомлять клиентов о переходе сеанса BFD из состояния Up в Down с возвратом в состояние Up, если это происходит в течение разумного интервала времени (размер интервала выходит за рамки этой спецификации). Если сессия BFD не возвращается в состояние Up в течение заданного времени, клиентов **следует** уведомить об отказе сессии.

3.2. Статус AdminDown

Механизм AdminDown в BFD предназначен для информирования об административном отключении сессии BFD и этот статус ничего не говорит о живучести пути данных.

Поэтому системе **не следует** указывать клиенту отказ в связности при переходе локального или удалённого (если оно известно) статуса сессии в AdminDown, если у клиента есть независимый способ проверки живучести (обычно протокол управления). Если у клиента нет независимого способа контроля живучести, системе **следует** сообщить клиенту о нарушении связности и принять семантику статуса Down при переходе состояния сессии на локальной или удалённой стороне в состояние AdminDown. Иначе клиент не сможет определить жизнеспособность пути, что может приводить к нежелательным результатам.

3.3. Организация и восстановление статуса BFD без влияния на клиентов

Полезно иметь возможность настройки сессии BFD между парой систем без влияния на связанных с сессией клиентов. Точно так же полезно восстанавливать статус BFD без нарушения работы клиентов при потере состояния BFD (например, в случае перезапуска). Это связано со способностью клиентов поддерживать свой статус независимо от BFD.

Переходам конечного автомата BFD в процессе активизации сессии BFD в таких случаях **не следует** вызывать уведомления клиентов о нарушении связности. Клиент, способный установить свой статус да настройки или перезапуска сессии BFD, **может** делать это при необходимости. Способы реализации этого выходят за рамки этой спецификации.

4. Взаимодействия с протоколами управления

Очень распространёнными клиентскими приложениями BFD являются протоколы управления, такие как протоколы маршрутизации. Когда BFD взаимодействует с протоколом управления, цель состоит в информировании того о связности протокола передачи данных. В случае протоколов маршрутизации это позволяет быстрее перенаправить трафик в случае отказа на действующий путь, нежели это делают естественные механизмы обнаружения.

4.1. Организация смежности

Если на локальной или удалённой (если известно) стороне сессия имеет статус AdminDown, протокол BFD будет отключён административно и **должна** быть разрешена организация смежности в протоколе управления.

Сессии BFD обычно организуются протоколом управления с использованием механизма (обнаружение, настройка), служащего протоколу управления для поиска соседей. Отметим, что при некоторых отказах сеть может находиться в состоянии, позволяющем работать протоколу управления, но не разрешающем организовать сессию BFD и, в частности, пересылать данные. Для предотвращения таких ситуаций полезно не разрешать протоколу управления создавать отношения смежности. Однако это помешало бы работе протокола управления в среде, где не все системы поддерживают BFD.

Поэтому организацию смежности в протоколах управления **следует** блокировать, если обе системы хотят организовать сессию BFD, но такая сессия невозможна. Одним из методов определить желание обеих систем организовать сессию BFD заключается в том, что протокол управления явно сигнализирует о таком желании. Если явной сигнализации нет, желание организовать сессию BFD можно определить средствами, выходящими за рамки этой спецификации.

Если предполагается, что соседняя система не поддерживает BFD, **не следует** блокировать организацию смежности протоколом управления.

Настройка режимов и временных параметров BFD не подлежит стандартизации. Отметим, что все протоколы, использующие сессию, будут работать с одним набором параметров. Механизм выбора параметров из числа желаемых разными протоколами, выходит за рамки этой спецификации. Обычно полезно выбирать параметры, обеспечивающие наименьшее значение Detection Time. Если какому-то приложению требуется гистерезис для уведомлений от BFD, оно будет предпочитать большее значение Detection Time.

Многие протоколы управления предполагают полную связность между всеми системами в средах с множественным доступом, таких как ЛВС. Если BFD работает лишь в части систем такой сети, организация смежности блокируется отсутствием сессии BFD, а допущения протокола управления могут нарушаться с непредсказуемыми результатами.

4.2. Реакция на смену состояния сессии BFD

Если сессия BFD переходит из состояния Up в AdminDown или из Up в Down в результате индикации удалённой системой статуса AdminDown, клиентам **не следует** выполнять каких-либо действий в протоколе управления.

Для других переходов из состояния Up в Down механизм реагирования протокола управления на индикацию отказа пути протоколом BFD зависит от возможностей протокола управления, как описано ниже.

4.2.1. Протоколы управления с одним протоколом данных

Протоколу управления, тесно связанному с отказом одного протокола данных, **следует** предпринимать действия, гарантирующие отправку трафика данных в обход отказавшего пути. Отметим, что это не следует считать заменой протоколом BFD механизмов проверки живучести протокола управления (если они есть), поскольку протокол управления может опираться на механизмы, не проверяемые BFD (например, групповая передача), поэтому BFD скорей всего будет обнаруживать не все отказы, влияющие на протокол управления. Однако протокол управления **может** выбрать применение сведений о статусе сессии BFD для более быстрого обнаружения надвигающегося отказа протокола управления, в частности, при работе этого протокола по основному каналу (in-band, по протоколу данных).

Поэтому при переходе сессии BFD из состояния Up в Down в протоколе управления **следует** выполнять действия, сигнализирующие об отсутствии связности на пути, по которому работает BFD. Если протокол управления имеет явный механизм анонсирования статуса пути, системе следует использовать такой механизм, не влияя на связность протокола управления, в частности, при работе протокола управления по отдельному от протокола данных каналу (out-of-band). Однако при недоступности такого механизма протоколу управления **следует** имитировать тайм-аут для соответствующего соседа.

4.2.2. Протоколы управления с несколькими протоколами данных

Несколько отличающиеся механизмы применяются в случае поддержки протоколом управления маршрутизации нескольких протоколов данных в зависимости от поддержки протоколом управления одной или разных топологий для протоколов данных.

4.2.2.1. Общая топология

При общей топологии в случае отказа одного протокола данных (как указывает соответствующая сессия BFD) необходимо считать этот путь отказавшим для всех протоколов данных. Иначе протокол управления не сможет переместить на другой путь трафик отказавшего протокола (и тот будет заблокирован на неопределённое время).

Поэтому при переходе сессии BFD из состояния Up в Down в протоколе управления **следует** выполнить действия по сигнализации отсутствия связности на пути в топологии, соответствующей сессии BFD. Если нет иной возможности передачи такого сигнала, **следует** имитировать тайм-аут для соответствующего соседа.

4.2.2.2. Независимые топологии

При использовании для каждого протокола своей топологии требуется перенаправлять лишь трафик протокола, столкнувшегося с отказом. Поэтому при переходе сессии BFD из состояния Up в Down, в протоколе управления **следует** выполнить действия по сигнализации потери связности на пути в топологии, где работает BFD. Обычно это можно сделать без влияния на связность других топологий (иначе очень сложно поддерживать разные топологии для нескольких протоколов данных).

4.3. Взаимодействие с механизмами аккуратного перезапуска

Многие протоколы управления, включая IS-IS [ISIS-GRACE], OSPF [OSPF-GRACE], BGP [BGP-GRACE], поддерживают механизм аккуратного перезапуска (Graceful Restart). Эти механизмы позволяют перезапустить протокол управления без нарушения статуса связности сети (чтобы не возникало впечатления об отказе системы и/или всех её каналов). Механизмы основаны на наличии отдельной плоскости пересылки, которая не обязательно разделяет судьбу плоскости управления, где работает протокол. В частности, предполагается возможность работы плоскости пересылки во время перезапуска и установки состояния протокола.

Реализации BFD указывают флагом C (Control Plane Independent), имеет ли BFD общую судьбу с плоскостью управления. Это служит для определения действий, применяемых вместе с Graceful Restart. Если BFD не зависит от плоскости управления какой-либо из систем, протокол можно использовать для обнаружения нежизнеспособности Graceful Restart в протоколе управления (нарушение работы плоскости пересылки).

Если протокол управления имеет механизм аккуратного перезапуска, BFD можно применять в комбинации с этим механизмом. Взаимодействие между BFD и протоколом управления зависит от возможностей протокола и наличия (отсутствия) связи между судьбой BFD и плоскости управления. В частности, может оказаться желательным при отказе сессии BFD прерывать процесс Graceful Restart, чтобы отказ можно было увидеть в сети.

4.3.1. BFD с независимой от плоскости управления судьбой

Если протокол BFD реализован в плоскости пересылки и не зависит от судьбы плоскости управления какой-либо из систем (бит C установлен в пакетах BFD Control для обоих направлений), перезапуск протокола управления не должен влиять на сессии BFD. В этом случае отказ сессии BFD предполагает невозможность пересылки данных, поэтому запущенные на момент отказа сессии BFD процедуры Graceful Restart **следует** прервать для предотвращения «чёрных дыр», а протокол управления **следует** информировать об изменении топологии.

4.3.2. BFD с зависимой от плоскости управления судьбой

Если BFD разделяет судьбу плоскости управления любой из систем (бит C сброшен для любого из направлений), отказ сессии BFD невозможно отделить от других событий в плоскости управления. Во многих случаях отказ сессии BFD будет побочным эффектом перезапуска. Поэтому следует по возможности избегать прерывания процедуры Graceful Restart, поскольку иначе BFD не сможет сосуществовать с Graceful Restart. При этом возникает некоторый риск, поскольку одновременный отказ или перезапуск плоскости пересылки не будет обнаружен, но это возникает во всех случаях, когда BFD разделяет судьбу плоскости управления.

4.3.2.1. Протоколы управления с сигнализацией запланированного перезапуска

Некоторые протоколы управления могут сообщать заранее о планируемом перезапуске. В этом случае при возникновении отказа сессии BFD в процессе перезапуска запланированный перезапуск **не следует** прерывать, а также **не следует** сообщать об изменении топологии в протоколе управления.

4.3.2.2. Протоколы управления без сигнализации запланированного перезапуска

Протоколы управления, не способные сообщать о запланированном перезапуске, зависят от перезапущенной системы в плане сигнализации Graceful Restart до завершение тайм-аута смежности в протоколе управления. В многих случаях, независимо от планирования перезапуска, вполне вероятно, что в сессии BFD возникнет тайм-аут до начала Graceful Restart и тогда **следует** сообщать об изменении топологии в протоколе управления, как указано в параграфе 3.2.

Однако, если перезапуск действительно запланирован, реализация **может** настроить временные параметры сессии BFD до перезапуска так, чтобы интервал Detection Time в каждом направлении был больше продолжительности перезапуска протокола управления, обеспечивая перезапускаемой системе такую же возможность использовать Graceful Restart, как это было бы без BFD. Перезапускаемой системе **не следует** передавать пакеты BFD Control, пока не будет высокой вероятности того, что соседи знают о процедуре Graceful Restart, поскольку первый пакет BFD Control будет вызывать отказ сессии BFD.

4.4. Взаимодействия с несколькими протоколами управления

Если несколько протоколов управления хотят организовать сессии BFD с одной удалённой системой для одного протокола данных, они **должны** использовать одну общую сессию BFD.

Если применяются иерархические или зависимые один от другого уровни протоколов управления (например, OSPF и IBGP), взаимодействие более одного из них с BFD может оказаться бесполезным. В упомянутом примере IBGP зависит от OSPF в плане маршрутных данных, поэтому быстрое информирование IBGP о возникающих отказах может оказаться даже вредным. Издержки смены статуса соседа в BGP достаточно велики, а OSPF естественным способом найдёт путь через сеть при обнаружении отказа.

В общем случае взаимодействовать с BFD лучше всего протоколу самого нижнего уровня в иерархии, а затем использовать взаимосвязи между протоколами для внесение требуемых изменений. Это обеспечит скорейшее обнаружение отказов и восстановление работы сети.

5. Взаимодействия с непротокольными функциями

Статус сессии BFD может использоваться для воздействия на функции системы, не основанные на протоколе (например, статические маршруты). При отказе пути к удалённой системе может быть желательным предотвращение передачи трафика в эту систему, поэтому локальная система может принять внутренние меры (такие как отзыв статических и динамических маршрутов) для достижения этого.

Если известна или предполагается поддержка BFD удалённой системой, а сессия BFD не находится в состоянии Up, **следует** предпринять соответствующий действия (такие как отзыв статических маршрутов). Если известно или предполагается отсутствие поддержки BFD в удалённой системе, такие действия, как отзыв маршрутов, **не следует** предпринимать.

Запуск сессии BFD для взаимодействия с непротокольными функциями, вероятно будет определяться конфигурацией. Не требуется обмена сведениями о конечных точках или дискриминаторах с помощью какого-либо механизма сверх данных конфигурации, поскольку конечные точки задаются и настраиваются одними средствами.

6. Протоколы данных и демультиплексирование

Протокол BFD предназначен для «защиты» одного «протокола данных» и инкапсулируется в этот протокол. Пара систем может организовать несколько BFD сессий в одной топологии, если они поддерживают (и инкапсулируются) в разные протоколы. Например, если две системы используют IPv4 и IPv6 по одному каналу между ними, это будет считаться двумя путями и потребует организации двух сессий BFD.

Такой же метод можно применять для более тонкого разделения путей. Например, при работе нескольких дифференцированных служб [DIFFSERV] через IPv4 можно использовать сессию BFD для каждого уровня обслуживания. Пакеты BFD Control должны маркироваться как и пакеты данных для обеспечения общей судьбы трафика BFD и данных, а также для демультиплексирования начального пакета, пока ещё не произошёл обмен дискриминаторами.

7. Подсети с множеством каналов

Имеется много технологий объединения нескольких параллельный каналов на уровне N-1, рассматриваемых как один путь на уровне N. Механизмы работы с такими каналами выходят за рамки спецификации, однако в этом разделе приведено несколько примеров. Возможны и другие варианты применения.

7.1. Полная развязка

Простейшим вариантом является просто организация BFD на пути уровня N без взаимодействия с механизмами уровня N-1. При этом предполагается, что механизм уровня N-1 обеспечивает обработку проблем связности в отдельных каналах уровня N-1. BFD будет фиксировать отказ на пути уровня N лишь по тайм-ауту сессии. Этот подход будет работать независимо от того, является ли сосед уровня N-1 также соседом уровня N.

7.2. Подсказки от уровня N-1

Несколько отличающийся вариант использует информирование уровнем N-1 протокола BFD уровня N о нежизнеспособности агрегированного канала. В этом случае сессия BFD будет обнаруживать отказы более быстро, поскольку не требуется ждать тайм-аута. Это аналогично фиксации отказа сессии при обнаружении аппаратного сбоя на одиночном канале. Этот подход будет работать независимо от того, является ли сосед уровня N-1 также соседом уровня N.

7.3. Агрегирование сессий BFD

Другой вариант использует сессию BFD на каждом канале уровня N-1 и объединяет состояния нескольких сессий в один сигнал индикации для клиентов уровня N. Преимуществом такого подхода является независимость от технологии уровня N-1. Однако этот подход работает лишь в случае, когда сосед уровня N является соседом и на уровне N-1 (один интервал пересылки на уровне N-1).

7.4. Комбинированный вариант

В некоторых случаях может быть полезна комбинация нескольких описанных выше (или иных) вариантов. Например, если соседи уровня N не соединены напрямую на уровне N-1, система может организовать сессию BFD через каждый канал уровня N-1 с непосредственным соседом на этом уровне и другую сессию BFD с соседом на уровне N. Агрегатное состояние сессий BFD на уровне N-1 можно использовать в качестве триггера отказа сессии BFD уровня N.

8. Другие вопросы с приложениями

BFD может обеспечивать обнаружение активности функций, связанных с OAM[1], в туннельных и псевдопроводных протоколах. Рекомендуется запуск BFD вне туннеля, поскольку при этом используется больше элементов пути. Одним из способов реализации является адресация пакетов BFD по конечным точкам туннеля в предположении наличия у них адресов.

[1]Operations, Administration, and Maintenance - эксплуатация, администрирование и поддержка (управление).

Если на пути, где работает BFD, происходит плановая остановка, предпочтительно заранее отключить сеанс BFD, переведя его в статус AdminDown. Системе, заявившей статус AdminDown, **следует** сохранять его в течение по меньшей мере интервала Detection Time, чтобы удалённая система гарантированно увидела изменение статуса.

BFD следует исключить из конфигурации системы, если желательно не вызывать каких-либо действий клиентского приложения. Простое прекращение передачи пакетов BFD Control приведёт лишь к тому, что удалённая система обнаружит отказ сессии. Для предотвращения этого системе, в которой BFD исключается из конфигурации, **следует** перевести сессию в состояние AdminDown и поддерживать его в течение интервала Detection Time, чтобы удалённая система гарантированно увидела изменение статуса.

9. Вопросы совместимости

Протокол BFD устроен так, что он всегда взаимодействует на базовом уровне - асинхронный режим является обязательным и доступен всегда, а другие режимы и функции согласуются в процессе работы. Поскольку предоставляемые BFD услуги не зависят от используемых вариантов, выбор опций BFD не влияет на совместимость.

Взаимодействие BFD с другими протоколами и функциями управления не требует сильных привязок. Действия предпринимаются на основе имеющихся в протоколах и функциях механизмов, поэтому проблемы взаимодействия маловероятны, пока BFD не применяется несовместимым способом (таким, как реакция на отказ сессии BFD отключением в одной реализации и активизацией в другой). Фактически BFD может рекомендовать разным системам разные функции управления и единственным последствием этого будет возможная асимметрия обнаружения отказов протокола управления.

10. Конкретные взаимодействия протокола

Как отмечено выше, проблем совместимости при взаимодействии BFD с протоколами управления не возникает. Однако в этой сфере достаточно непонимания и путаницы, поэтому ниже приведены некоторые примеры взаимодействия с конкретными протоколами. Поскольку взаимодействия не влияют на совместимость, этот раздел не является нормативным.

10.1. Взаимодействие BFD и OSPFv2, OSPFv3 и IS-IS

Для двух версий OSPF ([OSPFv2] и [OSPFv3]), а также протокола IS-IS [ISIS] характерны архитектурные ограничения, связанные с тем, что протоколы Hello не способны достаточно точно фиксировать время обнаружения отказа. В частности, для OSPF минимальное время обнаружения составляет 2 секунды, а для IS-IS - 1 секунду.

BFD можно использовать для обеспечения в этих протоколах сколь угодно малого времени обнаружения за счёт дополнения к протоколам Hello.

10.1.1. Организация сессии

Наиболее очевидным выбором для запуска организации сессии BFD в этих трёх протоколах является применение механизма обнаружения протоколов Hello в OSPF и IS-IS. Сессии BFD для поддержки OSPF и IS-IS через одни интервал пересылки IP (hop) должны работать в соответствии с [BFD-1HOP].

10.1.2. Реакция на смену состояния BFD

Базовые механизмы описаны в разделе 3. В настоящее время OSPFv2 и OSPFv3 передают маршрутные сведения для одного протокола данных (IPv4 и IPv6, соответственно) поэтому при потребности в сигнализации о смене топологии после отказа сессии BFD это следует делать путём отключения соответствующего соседа OSPF.

IS-IS можно использовать для поддержки одного или нескольких протоколов данных. В [ISIS] задана общая топология для нескольких протоколов данных, а работа над поддержкой нескольких топологий ещё не завершена. При использовании разных топологий для поддержки нескольких протоколов данных (или нескольких классов обслуживания в одном протоколе) определяемый топологией путь, связанный с отказавшей сессией BFD, следует прекратить анонсировать в IS-IS LSP (Link[1] Switched Path), чтобы сообщить о потере связности. В иных случаях об отказе в сессии BFD следует сообщать имитацией потери смежности IS-IS.

В OSPF имеется механизм сигнализации о плановом перезапуске, а в IS-IS такого механизма нет. Следует использовать подходящий механизм из числа описанных в параграфе 3.3.

10.1.3. Виртуальные каналы OSPF

Если нужно применять BFD для обнаружения отказов OSPF Virtual Link, **должен** применяться механизм, описанный в [BFD-MULTI], поскольку виртуальные каналы OSPF могут включать произвольное число пересылок (hop). В таких случаях **следует** и настоятельно рекомендуется применять аутентификацию BFD.

10.2. Взаимодействия с BGP

Протокол BFD может быть полезен в сессиях протокола EBGP (External Border Gateway Protocol) [BGP] для более быстрой смены топологии в случае отказа пути. Как отмечено в параграфе 4.4, сеансам IBGP обычно нецелесообразно взаимодействовать с BFD, если базовый протокол IGP уже делает это.

Сессии EBGP с применением BFD, можно организовать через один [BFD-1HOP] или несколько [BFD-MULTI] интервалов пересылки в зависимости от наличия прямой смежности с соседом. Сессию BFD следует организовывать с соседом BGP (в отличие от других Next Hop, анонсируемых в BGP). В таких случаях **следует** и настоятельно рекомендуется применять аутентификацию BFD.

В [BGP-GRACE] описан механизм Graceful Restart для протокола BGP. Если этот механизм не применяется в сессии EBGP и в соответствующей сессии BFD возникает отказ, сессию EBGP следует отключить в соответствии с параграфом 3.2. Если используется Graceful Restart, применимы базовые процедуры параграфа 4.3. в BGP Graceful

[1] В оригинале ошибочно сказано Label. См. https://www.rfc-editor.org/errata/eid4812. *Прим. перев.*

Restart нет сигнализации планового перезапуска, поэтому применим параграф 4.3.2.2. При прерывании Graceful Restart в соответствии с параграфом 4.3 принимающему узлу следует действовать как при завершении отсчёта таймера перезапуска [BGP-GRACE].

10.3. Взаимодействия с RIP

Протокол RIP (Routing Information Protocol) [RIP] отличается тем, что состояние смежности с соседом, как таковое, не сохраняется (по крайней мере в соответствии со спецификацией). Вместо этого установленные маршруты включают адрес next hop, который в большинстве случаев является адресом анонсирующего соседа (но может не быть им).

В случае RIP при отказе сессии BFD, связанной с соседом, следует имитировать завершение отсчёта тайм-аута для каждого маршрута, полученного от этого соседа. Отметим, что при отказе сессии BFD в случае несовпадения полученного от соседа адреса next hop с адресом самого соседа маршрут будет сохраняться до естественного тайм-аута (180 секунд). Однако при отслеживании реализацией всех маршрутов от каждого соседа все маршруты от соседа, соответствующего отказавшей сессии BFD, следует исключить по тайм-ауту, независимо от указанного в них next hop и таким способом избежать проблемы «застревания» маршрута.

11. Вопросы безопасности

С этой спецификацией не связано вопросов безопасности сверх указанных в нормативных спецификациях (см. ниже).

12. Литература

12.1. Нормативные документы

[BFD] Katz, D. and D. Ward, "Bidirectional Forwarding Detection", RFC 5880, June 2010.

[BFD-1HOP] Katz, D. and D. Ward,"Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, June 2010.

[BFD-MPLS] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, June 2010.

[BFD-MULTI] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for Multihop Paths", RFC 5883, June 2010.

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

12.2. Дополнительная литература

[BGP] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[BGP-GRACE] Sangli, S., Chen, E., Fernando, R., Scudder, J., and Y. Rekhter, "Graceful Restart Mechanism for BGP", RFC 4724, January 2007.

[DIFFSERV] 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, December 1998.

[ISIS] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, December 1990.

[ISIS-GRACE] Shand, M. and L. Ginsberg, "Restart Signaling for IS-IS", RFC 5306, October 2008.

[OSPFv2] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.

[OSPFv3] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, July 2008.

[OSPF-GRACE] Moy, J., Pillay-Esnault, P., and A. Lindem, "Graceful OSPF Restart", RFC 3623, November 2003.

[RIP] Malkin, G., "RIP Version 2", STD 56, RFC 2453, November 1998.

Адреса авторов

Dave Katz
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089-1206
USA
Phone: +1-408-745-2000
EMail: dkatz@juniper.net

Dave Ward
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089-1206
USA
Phone: +1-408-745-2000
EMail: dward@juniper.net

Перевод на русский язык

Николай Малых
nmalykh@protokols.ru