Internet Engineering Task Force (IETF) D. Katz
Request for Comments: 5880 D. Ward
Category: Standards Track Juniper Networks
ISSN: 2070-1721 June 2010

# Bidirectional Forwarding Detection (BFD)

Обнаружение двухсторонней пересылки

**Аннотация**

Этот документ описывает протокол, предназначенный для обнаружения отказов на двухстороннем пути между двумя машинами пересылки, включая интерфейсы и каналы данных с возможностью расширения на сами машины пересылки, с потенциально очень малой задержкой. Протокол работает независимо от среды, протоколов данных и маршрутизации.

**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/rfc5880.



# Оглавление



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

# *1. Введение*

Всё более важным свойством сетевого оборудования становится быстрое обнаружение коммуникационных отказов между смежными системами для ускоренной организации альтернативных путей. Обнаружение в некоторых случаях может быть достаточно быстрым, если в процессе участвует оборудование (например, сигнализация в Synchronous Optical Network или SONET). Однако в некоторых средах (таких как Ethernet) подобная сигнализация отсутствует, а в отдельных средах невозможно обнаружить некоторые типы отказов на пути, например, неисправность интерфейсов или компонент машины пересылки.

Сети используют сравнительно медленные механизмы Hello (обычно в протоколах маршрутизации) для обнаружения отказов при отсутствии аппаратной сигнализации. Время обнаружения отказа (Detection Time) в имеющихся протоколах составляет не меньше 1 секунды, что слишком много для некоторых приложений и ведёт к потере больших объёмов данных при гигабитных скоростях. Кроме того, Hello в протоколах маршрутизации невозможно воспользоваться при отсутствии маршрутизации и семантика этих сообщений несколько отличается от аппаратных сигналов - они позволяют обнаруживать отказы между двумя машинами протокола маршрутизации.

Целью BFD является быстрое и с незначительными издержками обнаружение отказов на пути между смежными машинами пересылки, включая интерфейсы, каналы данных и по возможности сами машины пересылки. Дополнительная цель состоит в обеспечении единого механизма проверки живучести в любой среде на любом протокольном уровне с широким диапазоном времени обнаружения и разными издержками.

Этот документ задаёт детали базового протокола. Использование некоторых механизмов зависит от приложения и будет описано в отдельных документах. Многие из механизмов зависят от реализации и не оказывают влияния на взаимодействие (совместимость), поэтому в данной спецификации не рассматриваются (это отмечено явно).

## *1.1. Уровни требований*

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. Устройство протокола*

BFD служит для обнаружения коммуникационных отказов в плоскости пересылки следующего интервала (hop). Протокол предназначен для реализации в некоторых компонентах машин пересылки, когда машины управления и пересылки разделены. Это не только дополнительно привязывает протокол к плоскости пересылки, но и отделяет его от машины протокола маршрутизации, делая протокол полезным в сочетании с «мягким перезапуском (graceful restart) протоколов маршрутизации. BFD можно реализовать и в машине управления, хотя это может помешать обнаружению некоторых типов отказов.

BFD работает на основе любого протокола данных (сетевой и канальный уровень, туннели и пр.) с пересылкой между двумя системами. Протокол всегда работает с индивидуальными адресами в режиме «точка-точка». Пакеты BFD передаются как данные (payload) любым протоколом инкапсуляции, подходящим для среды передачи и сети. BFD может работать в системе на нескольких уровнях. Контекст работы любой конкретной сессии BFD привязан к инкапсуляции.

BFD может обнаруживать отказы для любого типа путей между системами, включая прямое физическое соединение, виртуальное устройство (канал), путь с коммутацией по меткам MPLS 9MPLS Label Switched Path или LSP), путь с множеством маршрутизаторов и односторонний канал (при наличии обратного пути). Между парой систем можно организовать несколько сессий BFD при наличии между системами разных путей хотя бы в одном направлении (например, наличие множества параллельных односторонних каналов или MPLS LSP) даже при наличии в обратном направлении меньшего числа путей.

Конечный автомат BFD реализует трехэтапное согласование при организации сеанса BFD и разрыве сессии по любой причине, чтобы обе стороны знали о смене состояния.

BFD можно абстрагировать как простую службу. Сервисные примитивы BFD позволяют создавать, уничтожать и менять сессию на основе адреса получателя и других параметров. BFD предоставляет клиентам сигналы, указывающие активизацию или отключение сессии BFD.

# 3. Обзор протокола

BFD является простым протоколом Hello, во многом похожим на обычные компоненты обнаружения в общеизвестных протоколах маршрутизации. Пара систем периодически передаёт пакеты BFD по всем путям между ними и если система достаточно долго не получает пакетов BFD, какая-то часть двухстороннего пути от соседней системы считается отказавшей. При некоторых условиях системы могут согласовать отказ от периодической передачи BFD для снижения издержек.

Путь считается рабочим лишь в том случае, когда между системами установлена двухсторонняя связь, хотя это не исключает использование однонаправленных каналов.

Создаются раздельные сессии BFD для каждого коммуникационного пути и протокола данных, используемого между системами.

Каждая система оценивает, сколь быстро она может передавать и принимать пакеты BFD, чтобы согласовать с соседом, как быстро может быть обнаружен отказ при его возникновении. Эти оценки могут меняться в реальном масштабе времени для учёта необычных ситуаций. Такой подход позволяет быстрой системе, работающей в разделяемой среде с медленной системой, быстрее обнаруживать отказы между быстрыми системами, позволяя медленным системам работать в меру своих возможностей.

## 3.1. Адресация и организация сессии

Сессия BFD организуется на основе потребностей приложения, которое будет её использовать. Приложение должно определить потребность в BFD и используемые адреса (в BFD нет механизма обнаружения). Например, реализация OSPF [OSPF] может запросить сеанс BFD для соседнего узла, обнаруженного с помощью протокола OSPF Hello.

## 3.2. Режимы работы

BFD имеет два режима работы, которые можно выбирать, а также дополнительную функцию, которая может применяться с этими режимами.

Основным режимом является асинхронный (Asynchronous). В этом режиме системы периодически отправляют пакеты BFD Control на другую сторону и сессия считается не работающей (down), если другая сторона не получает несколько пакетов подряд.

Вторым является режим работы по запросу (Demand). В этом режиме предполагается, что у системы есть независимый способ проверки связности с другой системой. После организации сессии BFD такая система может попросить партнёра прекратить передачу пакетов BFD Control за исключением ситуаций, когда та чувствует необходимость явной проверки связности, для чего выполняется обмен короткой последовательностью пакетов BFD Control и удалённая система снова «успокаивается». Режим Demand работает в каждом направлении независимо или одновременно для обоих направлений.

Дополнением к обоим режимам является функция Echo. Когда эта функция активна, передаётся поток пакетов BFD так, чтобы другая система возвращала их по своему пути пересылки. Если какое-то число пакетов отражённого потока не получено, сессия считается разорванной (down). Функцию Echo можно использовать в режимах Asynchronous и Demand. Поскольку Echo выполняет задачу обнаружения, частота периодической отправки пакетов Control может быть снижена (в режиме Asynchronous) или их передача совсем прекращена (в режиме Demand).

Чистый режим Asynchronous имеет преимущество в том, что он требует вдвое меньшего числа пакетов для достижения определённого времени обнаружения (Detection Time) по сравнению с функцией Echo. Он также может применяться в случае отсутствия поддержки функции Echo.

Преимуществом функции Echo является фактическая проверка только пути пересылки удалённой системы. Это может снизить вариации периода кругового обхода (round-trip jitter) и обеспечить более быстрое обнаружение, а также обеспечить обнаружение некоторых отказов, которые иначе были бы не замечены.

Функцию Echo можно включать независимо для каждого направления. Функция включается в определённом направлении лишь в том случае, когда возвращающая пакеты Echo сторона указала, что она разрешает это, а система, передающая пакеты Echo, хочет этого.

Режим Demand полезен в ситуациях, где издержки периодической отправки могут быть нежелательны, например, в системах с большим числом сессий BFD. Он также полезен при симметричном использовании функции Echo. Недостатком режима Demand является то, что время обнаружения определяется эвристикой реализации системы и не известно протоколу BFD. Режим Demand не может применяться, когда время кругового обхода превышает желаемое значение Detection Time (см. параграф 6.6).

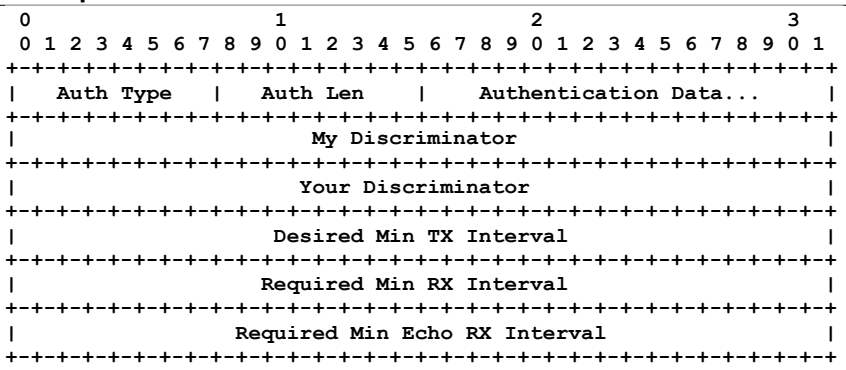
# 4. Формат пакетов BFD Control

## 4.1. Базовый формат BFD Control

Пакеты BFD Control передаются с подходящей для среды инкапсуляцией, выбор которой выходит за рамки документа.

Пакеты BFD Control имеют обязательный раздел (Mandatory Section) и раздел аутентификации (Authentication Section). Формат раздела аутентификации, если он применяется, зависит от способа проверки подлинности. Формат Mandatory Section в пакете BFD Control показан на рисунке.

В пакет **может** включаться показанный на рисунке раздел аутентификации (Authentication Section).




***Version (Vers)***

Версия протокола. Данный документ определяет протокол версии 1.

***Diagnostic (Diag)***

Диагностический код, указывающий локальную причину смены состояния.

0 - нет диагностики;
1 - истекло время обнаружения (Control Detection Time);
2 - отказ функции Echo;
3 - сосед сообщил об отказе (Down) сессии;
4 - сброс плоскости пересылки;
5 - отказ пути;
6 - отказ конкатенации путей;
7 - административное отключение;
8 - отказ обратной конкатенации путей;
9-31 - резерв на будущее.

Это поле позволяет удалённой системе, например, определить причину отказа предыдущей сессии.

***State (Sta)***

Статус текущей сессии BFD с точки зрения передающей системы.

0 - AdminDown (административно отключена);
1 - Down (отключена);
2 - Init (инициализация);
3 - Up (активна).

***Poll (P)***

Установка этого флага показывает, что передающая система запрашивает проверку связности или смены параметра и ожидает в ответ пакета с установленным битом Final (F). При сброшенном флаге передающая система не запрашивает проверки.

***Final (F)***

Установленный флаг показывает, что передающая система отвечает на пакет BFD Control с установленным флагом Poll (P). Сброшенный флаг указывает, что передающая система не отвечает на Poll.

***Control Plane Independent (C)***

Установка этого флага показывает, что реализация BFD передающей системы не имеет «общей судьбы» со своей плоскостью управления (т. е. BFD реализуется в плоскости пересылки и может продолжать работу при сбоях в плоскости управления). Сброшенный флаг указывает, что реализация BFD разделяет участь своей плоскости управления.

Использование этого бита зависит от приложения и выходит за рамки спецификации.

***Authentication Present (A)***

Установленный флаг указывает наличие Authentication Section и проверки подлинности сессии (см. параграф 6.7).

***Demand (D)***

Установленный флаг указывает, что в передающей системе активен режим Demand (система хочет работать в режиме Demand, знает об активности сессии в обоих направлениях и просит удалённую систему прекратить периодическую отправку пакетов BFD Control). Сброшенный флаг указывает, что в передающей системе не установлен режим Demand.

***Multipoint (M)***

Этот бит зарезервирован для будущих расширений point-to-multipoint BFD. Флаг **должен** быть сброшен при передаче и получении.

***Detect Mult***

Коэффициент для времени обнаружения. Согласованный интервал передачи, умноженные на это значение, задаёт Detection Time для принимающей системы в режиме Asynchronous.

***Length***

Размер пакета BFD Control в байтах.

***My Discriminator***

Уникальный ненулевой дискриминатор, генерируемый передающей системой и служащий для демультиплексирования разных сессий BFD между парой систем.

***Your Discriminator***

Дискриминатор, полученный от соответствующей удалённой системы. Это поле содержит полученное значение My Discriminator или 0, если дискриминатор удалённой системы неизвестен.

***Desired Min TX Interval***

Минимальный интервал (в миллисекундах), который локальная система желает использовать для передачи пакетов BFD Control без учёта применяемых вариаций (см. параграф 6.8.2). Значение 0 является резервным.

***Required Min RX Interval***

Минимальный интервал между пакетами BFD Control (в миллисекундах), который локальная система способна поддерживать без учёта применяемых отправителем вариаций (см. параграф 6.8.2). Нулевое значение показывает, что передающая система не желает, чтобы удалённая сторона передавала ей периодические пакеты BFD Control.

***Required Min Echo RX Interval***

Минимальный интервал между пакетами BFD Echo (в миллисекундах), который локальная система способна поддерживать без учёта применяемых отправителем вариаций (см. параграф 6.8.9). Нулевое значение показывает, что передающая система не приём пакетов BFD Echo.

***Auth Type***

Используемый тип аутентификации, если установлен бит Authentication Present (A).

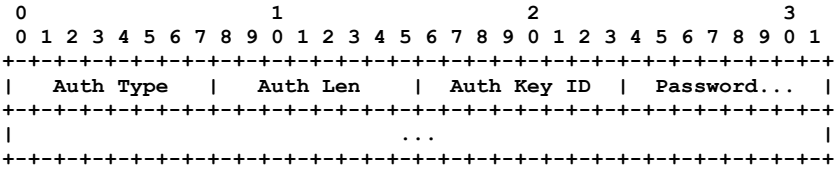
0 - резерв;
1 - простой пароль;
2 - Keyed MD5;
3 - Meticulous Keyed MD5;
4 - Keyed SHA1;
5 - Meticulous Keyed SHA1;
6-255 - резерв на будущее.

***Auth Len***

Размер раздела проверки подлинности (в байтах), включая поля Auth Type и Auth Len.

## 4.2. Формат Authentication Section для парольной аутентификации

Если в заголовке установлен бит Authentication Present (A) и поле Authentication Type имеет значение 1 (Simple Password), раздел Authentication имеет показанный на рисунке формат.




***Auth Type***

Тип аутентификации, который в данном случае имеет значение 1 (простой пароль).

***Auth Len***

Размер раздела Authentication в байтах. Для простой парольной аутентификации это будет размер пароля + 3.

***Auth Key ID***

Идентификатор ключа, используемого для пакета. Это позволяет поддерживать одновременно несколько ключей.

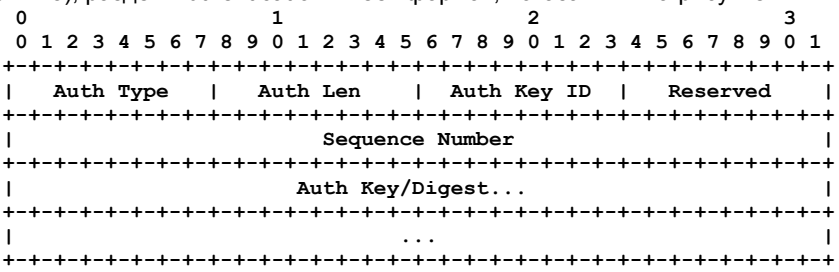
***Password***

Простой пароль, применяемый в данной сессии. Пароль представляет собой двоичную строку и **должен** иметь размер от 1 до 16 байтов. Пароль **должен** кодироваться и настраиваться в соответствии с параграфом 6.7.2.

## 4.3. Формат раздела аутентификации Keyed MD5 и Meticulous Keyed MD5

Настоятельно не рекомендуется применять аутентификацию на основе MD5, однако она описана здесь из соображений совместимости с имеющимися реализациями.

Если в заголовке установлен бит Authentication Present (A) и поле Authentication Type имеет значение 2 (Keyed MD5) или 3 (Meticulous Keyed MD5), раздел Authentication имеет формат, показанный на рисунке.




***Auth Type***

Тип аутентификации - 2 (Keyed MD5) или 3 (Meticulous Keyed MD5).

***Auth Len***

Размер раздела Authentication в байтах. Для аутентификации Keyed MD5 и Meticulous Keyed MD5 это 24.

***Auth Key ID***

Идентификатор ключа аутентификации, позволяющий поддерживать одновременно несколько ключей.

***Reserved***

Резервное поле, которое **должно** устанавливаться в 0 при передаче и игнорироваться при получении.

***Sequence Number***

Порядковый номер пакета. Для аутентификации Keyed MD5 это поле увеличивается на 1 время от времени, для Meticulous Keyed MD5 оно увеличивается на 1 в каждом следующем пакете сессии[1]. Это служит для защиты от атак с воспроизведением (replay).

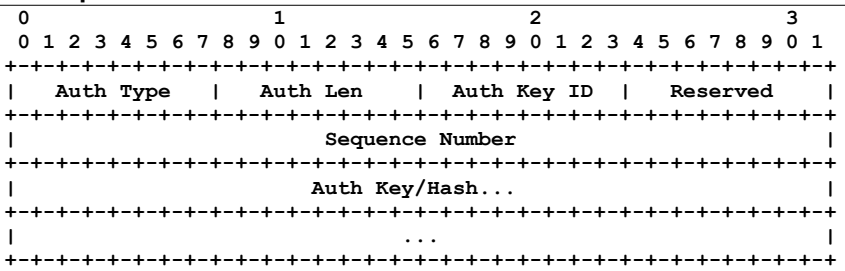
***Auth Key/Digest***

Это поле содержит 16-байтовый дайджест MD5 для пакета. При расчёте дайджеста в это поле помещается общий ключ MD5 с дополнением при необходимости нулями в конце до размера 16 байтов. Общий ключ **должен** кодироваться и настраиваться в соответствии с параграфом 6.7.3.

## 4.4. Формат раздела аутентификации Keyed SHA1 и Meticulous Keyed SHA1

Если в заголовке установлен бит Authentication Present (A) и поле Authentication Type имеет значение 4 (Keyed SHA1) или 5 (Meticulous Keyed SHA1), раздел Authentication имеет формат, показанный на рисунке.

[1] В оригинале не указано, что значение должно увеличиваться на 1. См. https://www.rfc-editor.org/errata/eid2530. *Прим. перев.*




***Auth Type***

Тип аутентификации - 4 (Keyed SHA1) или 5 (Meticulous Keyed SHA1).

***Auth Len***

Размер раздела Authentication в байтах. Для аутентификации Keyed SHA1 и Meticulous Keyed SHA1 это 28.

***Auth Key ID***

Идентификатор ключа аутентификации, позволяющий поддерживать одновременно несколько ключей.

***Reserved***

Резервное поле, которое **должно** устанавливаться в 0 при передаче и игнорироваться при получении.

***Sequence Number***

Порядковый номер пакета. Для аутентификации Keyed SHA1 это поле увеличивается время от времени, для Meticulous Keyed SHA1 оно увеличивается в каждом следующем пакете сессии. Это служит для защиты от атак с воспроизведением (replay).

***Auth Key/Hash***

Это поле содержит 20-байтовое хэш-значение SHA1 для пакета. При расчёте в поле помещается общий ключ SHA1, дополняемый при необходимости нулями в конце до 20 байтов. Общий ключ **должен** кодироваться и настраиваться в соответствии с параграфом 6.7.4.

# *5. Формат пакета BFD Echo*

Пакет BFD Echo передаются с подходящей для окружения инкапсуляцией, которая описывается в документации соответствующего приложения. Данные (payload) пакетов BFD Echo определяются локальными задачами, поскольку содержимое пакетов обрабатывает лишь локальная система. Единственным требованием является наличие информации, достаточной для демультиплексирования в нужную сессию BFD после возврата пакета отправителю. В остальном содержимое пакетов выходит за рамки этой спецификации.

Для пакетов Echo **следует** использовать ту или иную форму аутентификации, поскольку эти пакеты можно подделать.

# *6. Элементы процедуры*

В этом разделе рассматриваются нормативные требования протокола для обеспечения совместимости. Для разработчиков важно лишь соблюдение приведённых в этом разделе требований, поскольку опыт показывает, что излишняя педантичность препятствует совместимости реализаций.

Отметим, что все ссылки вида bfd.Xx относятся к внутренним переменным состояния (6.8.1. Переменные состояния), а ссылки вида «поле Xxx» относятся к самим полям протокола (4. Формат пакетов BFD Control).

## 6.1. Обзор

При инициализации сессии система может быть активной или пассивной. Активная система **должна** передавать пакеты BFD Control для конкретной сессии, независимо от получения каких-либо пакетов BFD для этой сессии. Пассивной системе **недопустимо** начинать передачу пакетов BFD для конкретной сессии, пока она не получит пакет BFD для этой сессии, из которого узнает дискриминатор удалённой системы. В сессии хотя бы одна из систем **должна** быть активной (возможно, обе). Роль системы определяется применением BFD и выходит за рамки спецификации.

Сессия начинается с периодического медленного обмена пакетами BFD Control. После достижения двухсторонней связи сессия BFD становится активной (Up). После активизации сеанса BFD система может запустить функцию Echo, если она хочет этого и другая система разрешает такую функцию. При активной функции Echo обычно сохраняется малая скорость передачи пакетов Control. Если функция Echo не активизирована, скорость передачи пакетов Control может быть повышена до уровня выполнения требований Detection Time для сессии.

Когда сессия активна (Up), система может сигнализировать о переходе в режим запроса (Demand) и передача пакетов BFD Control удалённой системой прекратится. Для поддержки состояния связности сессии будут применяться иные средства. Если любая из систем захочет проверить двухстороннюю связность, она может инициировать короткий обмен пакетами BFD Control (6.5. Последовательность опроса).

Если режим Demand не активен и не было получено пакетов Control в интервале Detection Time (6.8.4. Расчёт времени обнаружения), сессия считается разорванной (Down). Это указывается удалённому хосту в поле State (Sta) исходящих пакетов. Если потеряно достаточно много пакетов Echo, сессия также считается разорванной (см. 6.8.5. Обнаружение отказов с помощью функции Echo). Если активен режим Demand и не получено подходящих пакетов Control в ответ на Poll Sequence, сессия считается прерванной (см. 6.6. Режим Demand).

Если сессия разрывается (Down), передача пакетов Echo прекращается, а для пакетов Control возвращается малая скорость передачи. После того, как сессия сочтена разорванной (Down),она не может быть восстановлена, пока удалённая сторона не сообщит об отключении (выходом из состояния Up), реализуя 3-этапное согласование.

Сессию **можно** административно сохранять отключённой путём перевода в состояние AdminDown и передачи разъясняющего диагностического кода в поле Diagnostic.

## 6.2. Конечный автомат BFD

Конечный автомат BFD достаточно прост и включает 3 состояния, через которые обычно проходит сессия. Два состояния (Init и Up) относятся к организации сессий, а Down служит для разрыва. Это обеспечивает 3-этапное

согласование при создании и разрыве сессий (в предположении известности смены состояний обеим сторонам). Четвёртое состояние AdminDown служит для административного блокирования сессии на неопределённый срок.

Каждая система указывает своё состояние в поле State (Sta) пакетов BFD Control и эти сведения в комбинации с локальным состоянием сессии определяют состояние конечного автомата.

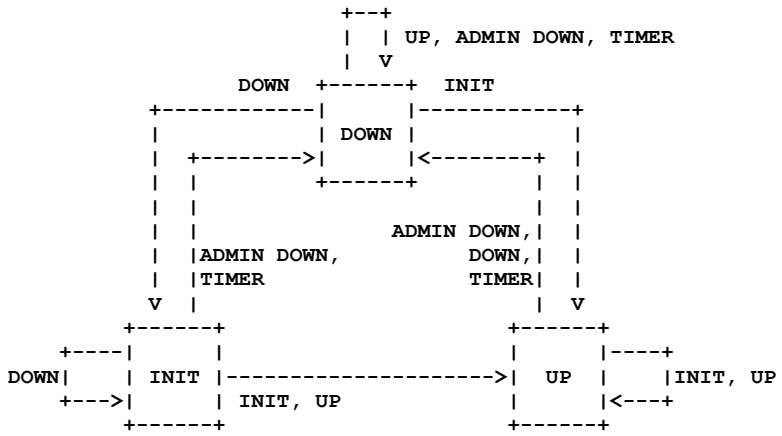
Состояние Down указывает, что сессия не работает или только что создана. Это состояние сохраняется, пока удалённая сторона не покажет, что она согласна с тем, что сессия не работает, передавая пакет BFD Control с полем State, отличным от Up. Если пакет указывает состояние Down, сессия переходит в состояние Init, если же указано состояние Init, сессия перейдёт в состояние Up. Семантически статус Down показывает, что путь пересылки недоступен и следует предпринять подобающие действия приложениям, отслеживающим состояние сессии BFD. Система **может** держать сессию в состоянии Down неограниченно долго (просто отказываясь от смены состояния). Это может быть обусловлено операционными, административными или иными причинами.

Состояние Init означает наличие связи с удалённой системой и желание локальной системы активизировать сессии, о чем удалённая система ещё не знает. Сессия остаётся в состоянии Init, пока не будет получен пакет BFD Control, указывающий статус Init или Up (в этом случае сессия перейдёт в состояние Up), или завершится время обнаружения (Detection Time), что означает потерю связи с удалённой точкой и переход сессии в состояние Down.

Состояние Up означает, что сессия BFD организована и это предполагает наличие связности между системами. Сессия будет сохранять статус Up до отказа в связности или административного отключения. Если любая из систем укажет статус Down или завершится Detection Time, сессия переходит в состояние Down.

Состояние AdminDown указывает административное отключение сессии. Это заставляет удалённую систему перейти в состояние Down и сохранять его, пока локальная система имеет статус AdminDown. Состояние AdminDown не говорит ничего о доступности пути пересылки.

На рисунке представлены конечный автомат протокола. Переходы, включающий статус AdminDown, не показаны для простоты (они полностью описаны в параграфах 6.8.6 и 6.8.16). Надписи около линий со стрелками указывают статус удалённой системы (из поля State в пакете BFD Control) или завершение отсчёта Detection Timer.




## 6.3. Демультиплексирование и дискриминаторы

Поскольку между системами можно организовать несколько сессий BFD, нужен механизм демультплексирования принятых пакетов BFD между сеансами. Каждая система **должна** выбрать необрабатываемое (opaque) значение дискриминатора для каждой своей сессии, которое **должно** быть уникальным среди всех сессий BFD в этой системе. Локальный дискриминатор передаётся в поле My Discriminator пакета BFD Control и возвращается другой стороной в поле Your Discriminator.

После возврата удалённой системой локального значения дискриминатора все последующие пакеты демультиплексируются лишь по значению поля Your Discriminator (среди прочего это позволяет изменить адрес отправителя или принимающий интерфейс без потери связи с соответствующей сессией).

Метод демультиплексирования начальных пакетов (с Your Discriminator = 0) зависит от приложения и выходит за рамки этой спецификации.

Системе разрешено менять свой дискриминатор во время работы сессии без влияния на статус сессии, поскольку дискриминатор используется этой системой лишь для демультиплексирования пакетов (удалённая система будет возвращать новый дискриминатор). Влияние смены дискриминатора на работу реализации выходит за рамки документа.

## 6.4. Функция Echo и асимметрия

Функцию Echo между парой систем можно включать независимо в каждом направлении. По какой-либо причине система может анонсировать своё желание получать (и возвращать) пакеты Echo и нежелание передавать такие пакеты самой. Возврат системой пакетов Echo не связан с передачей таких пакетов по инициативе этой системы.

При использовании системой функции Echo разумно выбрать умеренную скорость приёма пакетов Control, поскольку проверка живучести обеспечивается пакетами Echo. Это может контролироваться значением поля Required Min RX Interval (6.8.3. Манипуляции с таймерами).

Если функция Echo работает лишь в одном направлении, система, не применяющая Echo, скорей всего захочет получать пакеты Control достаточно часто для обеспечения желаемого Detection Time. Поскольку BFD разрешает разную скорость в каждом направлении, это реализуется легко.

В ином случае системе **следует** анонсировать наименьшие значения Required Min RX Interval и Required Min Echo RX Interval, которые она может применять в таких обстоятельствах, чтобы дать другой системе больше свободы при

выборе скорости передачи. Отметим, что система обязуется принимать оба потока пакетов с анонсированной скоростью и это требуется учитывать при анонсировании.

## 6.5. Последовательность опроса

Последовательность опроса (Poll Sequence) - это обмен пакетами BFD Control, используемый в некоторых случаях для гарантированного информирования удалённой системы об изменениях параметров. Он также применяется в режиме Demand (параграф 6.6) для проверки двухсторонней связности.

Poll Sequence представляет собой периодическую отправку пакетов BFD Control с установленным битом Poll (P). Когда система получает Poll, она сразу же передаёт пакет BFD Control с установленным битом Final (F), независимо от передаваемых ею периодических пакетов BFD Control (6.8.7. Передача пакетов BFD Control). Когда система, передающая Poll Sequence, получает пакет с битом Final, последовательность опроса прерывается и в последующих пакетах BFD Control бит Poll сбрасывается. В пакете BFD Control **недопустимо** устанавливать сразу оба бита Poll (P) и Final (F).

Если периодические пакеты BFD Control уже передаются (удалённая система не находится в режиме Demand), Poll Sequence **должна** применяться путём установки бита Poll (P) для запланированных периодических передач. Передача дополнительных пакетов **недопустима**.

После прерывания Poll Sequence запросившая Poll Sequence система прекратит передачу пакетов BFD Control, если удалённая система находится в режиме Demand. В ином случае она будет возвращаться к периодической отправке пакетов BFD Control со сброшенным битом Poll (P).

Типичная последовательность опроса состоит из одного пакета в каждом направлении, хотя потеря или слишком большая задержка пакета может приводить к передаче нескольких пакетов Poll до прерывания опроса.

## 6.6. Режим Demand

Режим Demand запрашивается независимо в каждом направлении установкой бита Demand (D) в пакетах BFD Control. Система, получившая бит Demand, прекращает периодически отправлять пакеты BFD Control. Если обе системы работают в режиме Demand, периодические пакеты BFD Control не передаются совсем.

Режим Demand требует какого-либо иного механизма поддержки непрерывной связности между системами. В каждом направлении может применяться свой механизм, но это выходит за рамки спецификации. Одним из возможных механизмов является приём трафика от другой стороны, другим - функция Echo.

Когда система в режиме Demand желает проверить двухстороннюю связность, она инициирует Poll Sequence (6.5. Последовательность опроса). Если на запрос не получено отклика, опрос повторяется до истечения Detection Time, после чего сессия считается не работающей (Down). Если режим Demand работает лишь на локальной системе, Poll Sequence выполняется простой установкой бита Poll (P) в обычных пакетах BFD Control, как указано в параграфе 6.5.

Время обнаружения в режиме Demand рассчитывается не так, как в асинхронном режиме и основано на скорости передачи локальной, а не удалённой системы. Это обеспечивает правильную работу механизма Poll Sequence (см. 6.8.4. Расчёт времени обнаружения).

Отметим, что механизм опроса всегда будет давать отказ, если согласованное значение Detection Time не больше времени кругового обхода между системами. Применение этого ограничения выходит за рамки спецификации.

Режим Demand **можно** включить или отключить независимо в каждом направлении путём установки или сброса бита Demand (D) в пакете BFD Control без воздействия на состояние сессии BFD. Бит Demand **недопустимо** устанавливать, пока обе стороны не считают сессию активной (Up) - локальная система считает сессию активной, а удалённая сообщила о состоянии Up в поле State (Sta) пакета BFD Control.

Когда переданное значение бита Demand (D) меняется, передающая система **должна** запустить Poll Sequence вместе со сменой бита, чтобы об изменении узнали обе системы.

Если режим Demand активен в одной или обеих системах, последовательность опроса Poll Sequence **должна** запускаться всякий раз, когда содержимое следующего передаваемого пакета отличается от предыдущего (без учёта битов Poll (P) и Final (F)). Это обеспечивает передачу удалённой системе изменений параметров для её оповещения.

Поскольку базовый механизм обнаружения не задаётся и может быть разным в паре систем, общие характеристики Detection Time для пути не будут полностью известны каждой системе. Общее время обнаружения для определённой системы является суммой времени до запуска Poll Sequence и расчётного значения Detection Time.

Если режим Demand включён лишь в одном направлении, непрерывная проверка двухсторонней связности будет потеряна и проверяться будет лишь связность в направлении от системы в режиме Demand. Решение проблемы для случая, когда одна система запрашивает режим Demand, а другая - двухстороннюю проверку связности, выходит за рамки этой спецификации.

## 6.7. Аутентификация

В пакете BFD Control **может** присутствовать раздел Authentication Section, служащий в своей базовой форме для передачи всей требуемой информации для проверки подлинности полученного пакета, определяемой типом аутентификации. Точный механизм зависит от используемого типа, но в общем случае передающая система помещает в Authentication Section сведения, подтверждающие достоверность пакета, а принимающая система проверяет эти данные и принимает пакет для дальнейшей обработки или отбрасывает его.

Очевидно, что обе системы должны использовать один тип аутентификации и ключи или иную информацию для проверки. Согласование типа аутентификации, обмен ключами и т. п. Выходят за рамки спецификации и, как ожидается, будут выполняться не входящими в состав протокола средствами.

Отметим, что в последующих параграфах восприятие (accept) пакета означает лишь прохождение аутентификации и пакет на деле может быть отброшен иными механизмами, как указано правилами обработки в параграфе 6.8.6.

Реализации с проверкой подлинности **должны** поддерживать оба типа аутентификации SHA1. Другие формы проверки являются необязательными.

### *6.7.1. Включение и отключение аутентификации*

Может оказаться желательным включение или отключение в сессии проверки подлинности без нарушения статуса сессии. Конкретный механизм такого управления выходит за рамки спецификации,однако полезно рассмотреть общие вопросы поддержки такого механизма.

В простой реализации при включении или отключении аутентификации в сессии BFD будет возникать отказ, поскольку правила восприятия пакетов требуют, чтобы локальная и удалённая система делали это в той или иной мере синхронно (в интервале Detection Time). Пакет с аутентификацией будет воспринят лишь при включённой аутентификации, а пакет без аутентификации - при отключённой.

Одним из вариантов реализации является подход, при котором аутентификация настраивается, но считается используемой лишь при получении первого пакета, соответствующего Authentication Section (обеспечивает требуемую синхронизацию). Точно так же, аутентификацию можно «отключить», но она будет применяться, пока не поступит первый пакет без Authentication Section.

Во избежание угроз безопасности реализациям такого метода **следует** разрешать смену состояния аутентификации не более одного раза без дополнительного вмешательства (что бы её нельзя было безостановочно включать и выключать лишь на основе пакетов BFD Control от удалённой системы). Если не требуется управлять аутентификацией, реализации **не следует** разрешать смену статуса проверки подлинности на основе получения пакетов BFD Control.

### *6.7.2. Простая парольная аутентификация*

Наиболее простой (и слабой) формой аутентификации является простой пароль (Simple Password). В этом случае в каждой системе настраивается один или несколько паролей (с соответствующими Key ID) и одна из пар «пароль-идентификатор» передаётся в каждом пакете BFD Control. Принимающая система воспринимает пакет, если Password и Key ID соответствуют настроенной в ней паре «пароль-идентификатор».

***Передача с использованием Simple Password Authentication***

Текущий выбранный пароль и Key ID для него **должны** указываться в Authentication Section каждого исходящего пакета BFD Control. В поле Auth Type **должно** быть указано значение 1 (Simple Password), а поле Auth Len **должно** указывать подобающий размер (4 - 19 байтов).

Пароль является двоичной строкой и **должен** иметь размер от 1 до 16 байтов. Для совместимости интерфейс управления для настройки пароля **должен** воспринимать строки ASCII и **следует** также разрешать указание произвольной двоичной строки в шестнадцатеричном формате. **Могут** поддерживаться и другие методы.

***Приём с использованием Simple Password Authentication***

Если полученный пакет BFD Control не содержит раздела Authentication Section или поле Auth Type отлично от 1 (Simple Password), принятый пакет **должен** отбрасываться.

Если Auth Key ID не соответствует идентификатору настроенного пароля, принятый пакет **должен** отбрасываться.

Если значение Auth Len не соответствует[1] размеру пароля, заданного Key ID, пакет **должен** отбрасываться.

Если поле Password не соответствует паролю, заданному Key ID, принятый пакет **должен** отбрасываться.

В остальных случаях пакет **должен** восприниматься.

### *6.7.3. Аутентификация Keyed MD5 и Meticulous Keyed MD5*

Механизмы проверки подлинности Keyed MD5 и Meticulous Keyed MD5 Authentication очень похожи на используемые в других протоколах. Для этих методов в каждой системе настраивается один или несколько общих секретных ключей (с соответствующими Key ID). Один из ключей помещается в дайджест MD5 [MD5], рассчитанный для исходящего пакета BFD Control, но сам ключ в пакете не передаётся. Для защиты от replay-атак в каждом пакете передаётся порядковый номер. Для аутентификации Keyed MD5 номер увеличивается время от времени, для Meticulous Keyed MD5 инкрементируется в каждом пакете.

Принимающая система воспринимает пакет, если Key ID соответствует одному из настроенных ключей, дайджест MD5, включающий выбранный ключ, соответствует переданному в пакете, а порядковый номер не меньше последнего полученного номера для Keyed MD5 и больше его для Meticulous Keyed MD5.

***Передача с аутентификацией Keyed MD5 и Meticulous Keyed MD5***

Поле Auth Type **должно** иметь значение 2 (Keyed MD5) или 3 (Meticulous Keyed MD5), а поле Auth Len - 24. В поле Auth Key ID **должен** указываться идентификатор текущего ключа аутентификации. Поле Sequence Number **должно** иметь значение переменной bfd.XmitAuthSeq.

Значением ключа аутентификации является двоичная строка размером до 16 байтов, которая **должна** помещаться в поле Auth Key/Digest с дополнением нулями в конце при необходимости. Для совместимости интерфейс управления, служащий для настройки ключей, **должен** воспринимать строки ASCII и **следует** также разрешать указание произвольной двоичной строки в 16-ричном формате. **Могут** поддерживаться и другие методы настройки.

Дайджест MD5 **должен** рассчитываться для всего пакета BFD Control. Результат расчёт **должен** помещаться в поле Auth Key/Digest до передачи (вместо секретного ключа, передача которого в пакете **недопустима**).

В режиме Keyed MD5 переменную bfd.XmitAuthSeq **можно** инкрементировать циклически (как 32-битовое целое число без знака). Переменную bfd.XmitAuthSeq **следует** инкрементировать при смене состояния сессии или при отправке пакета BFD Control, содержимое которого отличается от переданного ранее пакета. Решение об инкрементировании bfd.XmitAuthSeq выходит за рамки спецификации (см. 9. Вопросы безопасности).

[1] В оригинале ошибочно сказано «не равно». См. https://www.rfc-editor.org/errata/eid6818. *Прим. перев.*

В режиме Meticulous Keyed MD5 переменная bfd.XmitAuthSeq **должна** инкрементироваться циклически (как 32-битовое целое число без знака).

***Приём с аутентификацией Keyed MD5 и Meticulous Keyed MD5***

Если принятый пакет BFD Control не содержит Authentication Section, или поле Auth Type некорректно (2 для Keyed MD5 или 3 для Meticulous Keyed MD5), пакет **должен** отбрасываться.

Если поле Auth Key ID не соответствует идентификатору настроенного ключа, пакет **должен** отбрасываться.

Если значение поля Auth Len отличается от 24, пакет **должен** отбрасываться.

Если bfd.AuthSeqKnown = 1, проверяется поле Sequence Number. В режиме Keyed MD5 пакет **должен** отбрасываться, если порядковый номер не попадает в диапазон от bfd.RcvAuthSeq до bfd.RcvAuthSeq+(3*Detect Mult) включительно (как 32-битовое целое число без знака с учётом перехода через максимум). В режиме Meticulous Keyed MD5 пакет **должен** отбрасываться, если порядковый номер не попадает в диапазон от bfd.RcvAuthSeq+1 до bfd.RcvAuthSeq+(3*Detect Mult) включительно (как 32-битовое целое число без знака с учётом перехода через максимум).

В случае bfd.AuthSeqKnown = 0 **должно** устанавливаться bfd.AuthSeqKnown = 1, а в bfd.RcvAuthSeq **должно** устанавливаться значение из полученного поля Sequence Number.

Содержимое поля Auth Key/Digest заменяется ключом аутентификации, указанным в полученном поле Auth Key ID. Если дайджест MD5 для всего принятого поля BFD Control совпадает с полученным значением Auth Key/Digest, пакет **должен** быть воспринят, в противном случае пакет **должен** отбрасываться.

### *6.7.4. Аутентификация Keyed SHA1 и Meticulous Keyed SHA1*

Механизмы проверки подлинности Keyed SHA1 и Meticulous Keyed SHA1 очень похожи на применяемые в других протоколах. Для этих методов в каждой системе настраивается один или несколько общих секретных ключей (с соответствующими Key ID). Один из ключей помещается в хеш SHA1 [SHA1], вычисляемый для исходящего пакета BFD Control но сам ключ в пакете не передаётся. Для защиты от replay-атак в каждом пакете передаётся порядковый номер. Для аутентификации Keyed SHA1 номер увеличивается время от времени, для Meticulous Keyed SHA1 — в каждом пакете.

Принимающая система воспринимает пакет, если Key ID соответствует одному из настроенных ключей, хэш SHA1, включающий выбранный ключ, соответствует переданному в пакете, а порядковый номер не меньше последнего полученного номера для Keyed SHA1 и больше его для Meticulous Keyed SHA1.

***Передача с аутентификацией Keyed SHA1 и Meticulous Keyed SHA1***

В поле Auth Type **должно** быть установлено значение 4 (Keyed SHA1) или 5 (Meticulous Keyed SHA1), в поле Auth Len - 28. Поле Auth Key ID **должно** содержать идентификатор текущего ключа аутентификации, а Sequence Number - bfd.XmitAuthSeq.

Значением ключа аутентификации является двоичная строка размером до 20 байтов, которая **должна** помещаться в поле Auth Key/ Hash с дополнением нулями в конце при необходимости. Для совместимости интерфейс управления, служащий для настройки ключей, **должен** воспринимать строки ASCII и **следует** также разрешать указание произвольной двоичной строки в 16-ричном формате. **Могут** поддерживаться и другие методы настройки.

Хэш SHA1 **должен** рассчитываться для всего пакета BFD Control. Результат расчёта должен помещаться в поле Auth Key/Hash до передачи пакета (взамен секретного ключа, передавать который **недопустимо**).

В режиме Keyed SHA1 переменную bfd.XmitAuthSeq **можно** инкрементировать циклически (как 32-битовое целое число без знака). Переменную bfd.XmitAuthSeq **следует** инкрементировать при смене состояния сессии или при отправке пакета BFD Control, содержимое которого отличается от переданного ранее пакета. Решение об инкрементировании bfd.XmitAuthSeq выходит за рамки спецификации (см. 9. Вопросы безопасности).

В режиме Meticulous Keyed SHA1 переменная bfd.XmitAuthSeq **должна** инкрементироваться циклически (как 32-битовое целое число без знака).

***Приём с аутентификацией Keyed SHA1 и Meticulous Keyed SHA1***

Если принятый пакет BFD Control не содержит Authentication Section, или поле Auth Type некорректно (4 для Keyed SHA1 или 5 для Meticulous Keyed SHA1) пакет **должен** отбрасываться.

Если поле Auth Key ID не соответствует идентификатору настроенного ключа, пакет **должен** отбрасываться.

Если значение поля Auth Len отличается от 28, пакет **должен** отбрасываться.

Если bfd.AuthSeqKnown = 1, проверяется поле Sequence Number. В режиме Keyed SHA1 пакет **должен** отбрасываться, если порядковый номер не попадает в диапазон от bfd.RcvAuthSeq до bfd.RcvAuthSeq+(3*Detect Mult) включительно (как 32-битовое целое число без знака с учётом перехода через максимум). В режиме Meticulous Keyed SHA1 пакет **должен** отбрасываться, если порядковый номер не попадает в диапазон от bfd.RcvAuthSeq+1 до bfd.RcvAuthSeq+(3*Detect Mult) включительно (как 32-битовое целое число без знака с учётом перехода через максимум).

В случае bfd.AuthSeqKnown = 0 **должно** устанавливаться bfd.AuthSeqKnown = 1, а в bfd.RcvAuthSeq **должно** устанавливаться значение из полученного поля Sequence Number.

Содержимое поля Auth Key/Hash заменяется ключом аутентификации, указанным в полученном поле Auth Key ID. Если дайджест SHA1 для всего принятого поля BFD Control совпадает с полученным значением Auth Key/Hash, пакет **должен** быть воспринят, в противном случае пакет **должен** отбрасываться.

## 6.8. Функциональная специфика

Последующие параграфы этой спецификации являются нормативными без указания способов достижения.

Когда говорят об активности функции Echo в системе, это означает, что система передаёт пакеты BFD Echo, подразумевая активность сессии (Up) и согласие другой системы возвращать пакеты Echo.

Когда говорят об активности в локальной системе режима Demand, это означает установку bfd.DemandMode = 1 в локальной системе (см. параграф 6.8.1), активность сессии (Up), и наличие от удалённой системы сигнала о состоянии сессии Up. Когда говорят об активности режима Demand в удалённой системе, это означает установку bfd.RemoteDemandMode = 1 (удалённая система установила бит Demand (D) в последнем принятом от неё пакете BFD Control), активность сессии (Up) и наличие от удалённой системы сигнала о состоянии сессии Up.

### *6.8.1. Переменные состояния*

Для выполнения элементов описанной здесь процедуры требуется отслеживать минимальный объем сведений о состоянии сессии. Приведённый ниже набор переменных может быть полезен для описания механизмов BFD. Для отслеживания состояния можно применять любые способы, обеспечивающие соответствие протокола спецификации.

Когда в тексте указана инициализация переменной состояния, это относится лишь к организации сессии (и соответствующих переменных состояния). Далее переменными состояния управляет конечный автомат и они никогда не инициализируются повторно даже в случае отказа и восстановления сессии.

После организации состояния сессии и получения хотя бы одного пакета BFD Control от удалённой стороны, оно **должно** сохраняться в течение по меньшей мере 1 интервала Detection Time (см. параграф 6.8.4) с момента получения последнего пакета BFD Control, независимо от состояния сессии. Это сохраняет временные параметры в случае флуктуаций состояния. Система **может** сохранять состояние сессии более долго. Сохранение или уничтожение статуса сессии при отсутствии в ней пакетов BFD Control выходит за рамки спецификации.

Все переменные состояния в этом документе имеют вид bfd.Xx и их не следует путать с полями пакетов, которые всегда указываются по именам (см. раздел 4).

***bfd.SessionState***
Воспринимаемое состояние сессии (Init, Up, Down, AdminDown). Конкретные действия при смене состояния выходят за рамки спецификации, хотя предполагается, что об изменении (особенно при переходе в состояние Up или из него) сообщается другим компонентам системы. Переменная **должна** инициализироваться как Down.

***bfd.RemoteSessionState***
Состояние сессии, указанное в последний раз удалённой системе полем State (Sta) в пакете BFD Control. Переменная **должна** инициализироваться как Down.

***bfd.LocalDiscr***
Локальный дискриминатор данной сессии BFD, служащий для её однозначной идентификации. Дискриминатор **должен** быть уникальным для каждой сессии BFD в системе и отличным от 0. **Следует** задавать случайные (но уникальные) для повышения защищенности. Конкретные значения выходят за рамки спецификации.

***bfd.RemoteDiscr***
Удалённый дискриминатор данной сессии BFD. Этот дискриминатор выбирает удалённая система, а локальная никак не анализирует его (opaque). Переменная **должна** инициализироваться значением 0. Если прошло время Detection Time и не получено действительного, аутентифицированного пакета BFD от удалённой системы, переменная **должна** сбрасываться в 0.

***bfd.LocalDiag***
Диагностический код, указывающий причину последнего изменения локального состояния сессии. Переменная **должна** инициализирована значением 0 (No Diagnostic - нет диагностики).

***bfd.DesiredMinTxInterval***
Минимальный интервал (в микросекундах) между передачей пакетов BFD Control, который локальная система хотела бы использовать в данное время, без учёта применяемых вариаций (см. параграф 6.8.2). Фактическое значение согласуется между двумя системами. Переменная **должна** инициализироваться значением не менее 1 секунды (1000000) в соответствии с правилами параграфа 6.8.3. Выбор значения спецификация не задаёт.

***bfd.RequiredMinRxInterval***
Минимальный интервал (в микросекундах) между приёмом пакетов BFD Control, который локальная система хотела бы использовать в данное время, без учёта применяемых вариаций (см. параграф 6.8.2). выбор значения выходит за рамки спецификации. Значение 0 говорит о нежелании получать периодические пакеты BFD Control (см. параграф 6.8.18).

***bfd.RemoteMinRxInterval***
Последнее значение интервала Required Min RX от удалённой системы, полученное в пакете BFD Control. Переменная **должна** инициализироваться значением 1.

***bfd.DemandMode***
Устанавливается 1, если локальная система хочет использовать режим Demand. В противном случае 0.

***bfd.RemoteDemandMode***
Устанавливается 1, если удалённая система хочет использовать режим Demand. В противном случае 0. Это значение бита Demand (D) из последнего принятого пакета BFD Control. Переменная **должна** инициализироваться значением 0.

***bfd.DetectMult***
Желаемый коэффициент (multiplier) Detection Time для пакетов BFD Control в локальной системе. Согласованный интервал передачи пакетов Control, умноженный на значение этой переменной, будет давать значение Detection Time для этой сессии (с точки зрения удалённой системы). Переменная **должна** иметь отличное от нуля целочисленное значение, выбор которого данная спецификация не задаёт. Подробности управления таймером описаны в параграфе 6.8.4.

***bfd.AuthType***
Тип аутентификации для сессии, как указано в параграфе 4.1, или 0, если аутентификация не применяется.

***bfd.RcvAuthSeq***
32-битовое целое число без знака, содержащее последний полученный порядковый номер при аутентификации Keyed MD5 или Keyed SHA1. Начальное значение может быть любым.

***bfd.XmitAuthSeq***
32-битовое целое число без знака, содержащее следующий передаваемый порядковый номер при аутентификации Keyed MD5 или Keyed SHA1. Переменная **должна** инициализироваться случайным 32-битовым значением.

***bfd.AuthSeqKnown***

Устанавливается значение 1, если следующий порядковый номер для аутентификации Keyed MD5 или Keyed SHA1 известен. В противном случае устанавливается 0. Переменная **должна** инициализироваться значением 0.

Переменная **должна** сбрасываться в 0 при отсутствии принятых в этой сессии пакетов в течение по меньшей мере удвоенного интервала Detection Time. Это обеспечивает ресинхронизацию порядковых номеров при перезапуске удалённой системы.

### *6.8.2. Согласование таймеров*

Значение, используемое для интервала передачи пакетов BFD и Detection Time для сессии, постоянно согласуется и может меняться с течением времени. Значения согласования и времени определяются независимо для каждого направления в каждой сессии.

Каждая система сообщает в пакете BFD Control, сколь часто она хотела бы передавать пакеты BFD, а также принимать такие пакеты. Это позволяет любой системе в одностороннем порядке определять максимальную частоту (минимальный интервал) передачи пакетов в обоих направлениях. Подробности согласования и времени передачи пакетов приведены в параграфе 6.8.7.

### *6.8.3. Манипуляции с таймерами*

Значения времени, применяемые для определения интервалов передачи пакетов BFD и Detection Time в сессии, могут в любой момент изменяться без влияния на статус сессии. Требования этого параграфа применяются при любых изменениях параметров таймеров.

При изменении bfd.DesiredMinTxInterval или bfd.RequiredMinRxInterval **должна** запускаться процедура Poll Sequence (параграф 6.5). Если изменения таковы, что показывают желание получившей Poll Sequence системы поменять описанные в этом параграфе параметры, новые значения параметров **могут** передаваться в пакетах с установленным флагом Final (F), даже если последовательность опроса (Poll Sequence) ещё не была передана.

Если bfd.DesiredMinTxInterval увеличивается, а bfd.SessionState = Up, фактически используемый интервал передачи **недопустимо** менять, пока описанная выше процедура Poll Sequence не будет завершена. Это обеспечивает обновление удалённой системой своего значения Detection Time до увеличения интервала передачи.

Если bfd.RequiredMinRxInterval снижается, а bfd.SessionState = Up, **должно** применяться предыдущее значение bfd.RequiredMinRxInterval при расчёте Detection Time для удалённой системы, пока описанная выше процедура Poll Sequence не завершена. Это нужно для гарантии того, что удалённая система будет передавать пакеты с высокой скоростью (и эти пакеты будут получены) до снижения Detection Time.

Когда bfd.SessionState отличается от Up, система **должна** установить для bfd.DesiredMinTxInterval значение не меньше 1 секунды (1000000). Это предназначено для того, чтобы пропускная способность, потребляемая сессией BFD с отличным от Up состоянием была пренебрежимо мала, особенно в случае, когда сосед может не применять BFD.

Если локальная система уменьшает интервал передачи в результате снижения bfd.RemoteMinRxInterval (удалённая система анонсировала сниженное значение Required Min RX Interval), а удалённая система не находится в режиме Demand, локальная система **должна** незамедлительно начать соблюдение этого интервала. Иными словами, локальная система не может ждать дольше нового интервала между передачей предыдущего и следующего пакета. Если этот интервал уже прошёл с момента последней передачи (поскольку новый интервал существенно короче), локальная система **должна** передать следующий периодический пакет BFD Control как можно скорее.

Когда активна функция Echo, системе **следует** для bfd.RequiredMinRxInterval значение не меньше 1 секунды (1000000). Это предназначено для сохранения пренебрежимо малого принимаемого трафика BFD Control, поскольку применяется функция фактического определения с использованием пакетов BFD Echo.

Во всех случаях, кроме явно упомянутых выше, изменения временных параметров **должны** применяться незамедлительно (изменение скорости передачи и/или Detection Time).

Отметим, что механизм Poll Sequence неоднозначен при наличии более одного изменения параметров, которое требуется использовать, и эти множественные изменения распределены между разными пакетами (поскольку семантика возврата флага Final неясна). Поэтому при наличии нескольких изменений, требующих Poll Sequence, имеется 3 разных случая.

1) Изменения **должны** быть переданы в одном пакете BFD Control, чтобы семантика отклика Final была ясна.

2) Должно пройти достаточно времени с момента завершения Poll Sequence для устранения неоднозначности ситуации (по меньшей мере один период кругового обхода с момента передачи последнего Poll) до запуска другой процедуры Poll Sequence.

3) **Должен** быть получен дополнительный пакет BFD Control со сброшенным битом Final (F) после завершения процедуры Poll Sequence до запуска другой процедуры Poll Sequence (этот вариант недоступен при активном режиме Demand).

### *6.8.4. Расчёт времени обнаружения*

Detection Time (интервал без получения пакетов BFD, по истечении которого фиксируется отказ сессии) не передаётся протоколом явно. Вместо этого значение рассчитывается независимо для каждого направления принимающей системой на основе согласованного интервала передачи и коэффициента детектирования. Отметим, что значение Detection Time может быть разным в каждом направлении. Расчёт Detection Time несколько различается в режимах Demand и Asynchronous.

В асинхронном режиме значение Detection Time, рассчитанное в локальной системе, равно значению Detect Mult, полученному от удалённой системы, умноженному на согласованный интервал передачи удалённой системой (большее из bfd.RequiredMinRxInterval и последнего полученного Desired Min TX Interval). Значение Detect Mult - это (грубо) число отсутствующих подряд пакетов, при котором сессия считается неработающей.

Если режим Demand не активен и прошёл интервал времени Detection Time без получения пакетов BFD Control от удалённой системы, сессия считается неработающей - локальная система **должна** установить bfd.SessionState = Down и bfd.LocalDiag = 1 (Control Detection Time Expired - истекло время обнаружения)[1].

В режиме Demand значение Detection Time, рассчитанное локальной системой, равно bfd.DetectMult, умноженному на согласованный интервал передачи локальной системой (большее из bfd.DesiredMinTxInterval и bfd.RemoteMinRxInterval). Значение bfd.DetectMult - это (грубо) число отсутствующих подряд пакетов, при котором сессия считается неработающей.

Если режим Demand активен и прошло время, равное Detection Time, после запуска Poll Sequence (передача пакета BFD Control с флагом Poll) без получения пакета BFD Control с флагом Final (F) от удалённой системы, сессия считается неработающей - локальная система **должна** установить bfd.SessionState = Down и bfd.LocalDiag = 1 (Control Detection Time Expired - истекло время обнаружения)[2].

Здесь пакет считается полученным (для определения момента истечения Detection Time), лишь в том случае, когда он не «отброшен» в соответствии с правилами параграфа 6.8.6.

### *6.8.5. Обнаружение отказов с помощью функции Echo*

Когда активна функция Echo и достаточное число пакетов Echo не прибыло должным образом, сессия считается неработающей - локальная система **должна** установить bfd.SessionState = Down и bfd.LocalDiag = 2 (Echo Function Failed - отказ функции Echo).

Способы обнаружения отказов функции Echo выходят за рамки спецификации. Подходит любой способ.

### *6.8.6. Приём пакетов BFD Control*

При получении пакета BFD Control **должны** выполняться перечисленные ниже процедуры в указанном порядке. Если пакет отбрасывается в соответствии с этими правилами, обработка пакетов **должна** прерываться в этой точке.

Если номер версии отличается от 1, пакет **должен** быть отброшен.

Если поле Length меньше минимально допустимого (24 при сброшенном бите A, 26 при установленном), пакет **должен** быть отброшен.

Если поле Length больше размера данных инкапсулированного протокола, пакет **должен** быть отброшен.

Если Detect Mult = 0, пакет **должен** быть отброшен.

Если бит Multipoint (M) установлен (1), пакет **должен** быть отброшен.

Если поле My Discriminator = 0, пакет **должен** быть отброшен.

Если поле Your Discriminator отлично от 0, оно **должно** использоваться для выбора сессии, с которой связан пакет BFD. Если сессии не найдено, пакет **должен** быть отброшен.

Если поле Your Discriminator = 0, а поле State отличается от Down и AdminDown, пакет **должен** быть отброшен.

Если поле Your Discriminator = 0, сессия **должна** выбираться на основе той или иной комбинации других полей, возможно включающей адрес отправителя, My Discriminator и интерфейс, через который получен пакет. Точный метод выбора зависит от приложения и выходит за рамки спецификации. Если соответствующей сессии не найдено, **можно** создать новую сессию или отбросить пакет (спецификация не задаёт этот выбор).

Если установлен бит A, но аутентификация не применяется (bfd.AuthType = 0), пакет **должен** быть отброшен. Если бит A сброшен и применяется аутентификация (значение bfd.AuthType отлично от 0), пакет **должен** быть отброшен. Если бит A установлен, пакет **должен** аутентифицироваться по правилам параграфа 6.7 на основе выбранного типа аутентификации (bfd.AuthType). Это может приводить к отбрасыванию пакета.

В переменной bfd.RemoteDiscr устанавливается значение My Discriminator.

В переменной bfd.RemoteState устанавливается значение поля State (Sta).

В переменной bfd.RemoteDemandMode устанавливается значение флага Demand (D).

В переменной bfd.RemoteMinRxInterval устанавливается значение Required Min RX Interval.

Если поле Required Min Echo RX Interval имеет значение 0, передача пакетов Echo **должна** прекращаться.

Если локальная система передаёт Poll Sequence и в принятом пакете установлен бит Final (F), передача Poll Sequence **должна** прерываться.

Обновляется интервал передачи, как указано в параграфе 6.8.2. Согласование таймеров.

Обновляется время обнаружения (Detection Time), как указано в параграфе 6.8.4. Расчёт времени обнаружения.

Если bfd.SessionState = AdminDown, пакет отбрасывается.

Если принят статус AdminDown

Если bfd.SessionState отличается от Down

Устанавливается bfd.LocalDiag = 3 (сосед сообщил о неработающей сессии - down);

Устанавливается bfd.SessionState = Down

В остальных случаях

Если bfd.SessionState = Down

[1]В оригинале это предложение содержало ошибку. См. https://www.rfc-editor.org/errata/eid5205. Прим. перев.
[2]В оригинале это предложение содержало ошибку. См. https://www.rfc-editor.org/errata/eid4410. Прим. перев.

Если получено State = Down, устанавливается bfd.SessionState = Init

Если получено State = Init, устанавливается bfd.SessionState = Up

Если bfd.SessionState = Init

Если получено State = Init, устанавливается bfd.SessionState = Up

Иначе (bfd.SessionState = Up)

Если получено State = Down

Устанавливается bfd.LocalDiag = 3 (сосед сообщил о неработающей сессии - down);

Устанавливается bfd.SessionState = Down.

Проверяется, нужно ли устанавливать режим Demand (6.6. Режим Demand).

Если bfd.RemoteDemandMode = 1, bfd.SessionState = Up, bfd.RemoteSessionState = Up, это говорит об активности режима Demand в удалённой системе и локальная система **должна** прекратить периодическую отправку пакетов BFD Control (6.8.7. Передача пакетов BFD Control).

Если bfd.RemoteDemandMode = 0, bfd.SessionState отличается от Up или bfd.RemoteSessionState отличается от Up, это говорит о том, что режим Demand не активизирован на удалённой системе и локальная система **должна** передавать периодические пакеты BFD Control (6.8.7. Передача пакетов BFD Control).

Если установлен бит Poll (P), удалённой системе передаётся пакет BFD Control со сброшенным битом Poll (P) и установленным битом Final (F) (6.8.7. Передача пакетов BFD Control).

Если пакет не был отброшен, он считается полученным для применения правил проверки Detection Time, указанных в параграфе 6.8.4. Расчёт времени обнаружения.

## *6.8.7. Передача пакетов BFD Control*

За исключением перечисленных ниже случаев, системе **недопустимо** передавать пакеты BFD Control с интервалом меньше большего из двух значений bfd.DesiredMinTxInterval и bfd.RemoteMinRxInterval без учёта применяемых вариаций (см. ниже). Иными словами, скорость передачи определяет система, задавшая больший интервал.

Для периодической передачи пакетов BFD Control **должны** применяться вариации (jitter) до 25% на уровне пакета, т. е. интервал **должен** уменьшаться на случайное значение от 0 до 25%, чтобы предотвратить ненужную синхронизацию с другими системами в той же подсети. Таким образом, средний интервал будет приблизительно на 12,5% меньше согласованного.

Если bfd.DetectMult = 1, интервал между передаваемыми пакетами BFD Control **должен** быть не более 90% от согласованного интервала передачи и не менее 75% этого интервала. Это нужно для того, чтобы на удалённой системе не прошло рассчитанное время Detection Time до получения следующего пакета BFD.

Интервал передачи **должен** пересчитываться при каждом изменении bfd.DesiredMinTxInterval или bfd.RemoteMinRxInterval и равен большему из этих значений. Таймеры передачи описаны в параграфах 6.8.2 и 6.8.3.

Системе **недопустимо** передавать пакеты BFD Control, если bfd.RemoteDiscr = 0 и система играет пассивную роль.

Системе **недопустимо** периодически передавать пакеты BFD Control, если bfd.RemoteMinRxInterval = 0.

Системе **недопустимо** периодически передавать пакеты BFD Control, если на удалённой системе активен режим Demand (bfd.RemoteDemandMode = 1, bfd.SessionState = Up, bfd.RemoteSessionState = Up) и последовательность Poll Sequence не передаётся.

Если получен пакет BFD Control с битом Poll (P) = 1, принимающая система **должна** передать пакет BFD Control со сброшенным битом Poll (P) и установленным битом Final (F) как можно скорее без учёта таймера передачи и иных ограничений, независимо от состояния режима Demand в любой из систем. Система **может** ограничивать скорость передачи таких пакетов. Если такое ограничение применяется, анонсированное значение Desired Min TX Interval **должно** быть не меньше интервала между передачей пакетов, заданного функцией ограничения скорости.

Системе **недопустимо** устанавливать бит Demand (D), пока не соблюдается условие bfd.DemandMode = 1, bfd.SessionState = Up, bfd.RemoteSessionState = Up.

**Следует** передавать пакет BFD Control в интервале между периодическими пакетами Control, когда содержимое этого пакета будет отличаться от переданного ранее (за исключением битов Poll и Final), для более быстрого информирования о смене состояния.

Для содержимого передаваемых пакетов BFD Control **должны** устанавливаться приведённые ниже значения.

***Version***
Текущий номер версии (1).

***Diagnostic (Diag)***
bfd.LocalDiag.

***State (Sta)***
Значение, указанное в bfd.SessionState.

***Poll (P)***
Если локальная система передаёт Poll Sequence, устанавливается 1. В противном случае 0.

***Final (F)***
1, если локальная система система отвечает на пакет Control с установленным битом Poll (P). В ином случае 0.

***Control Plane Independent (C)***
1, если реализация BFD в локальной система не зависит от плоскости управления (может работать без неё).

***Authentication Present (A)***
1, если в сессии применяется аутентификация (bfd.AuthType отлично от 0). В противном случае 0.

***Demand (D)***
bfd.DemandMode, если bfd.SessionState = Up и bfd.RemoteSessionState = Up. В ином случае 0.

***Multipoint (M)***
0.

***Detect Mult***
bfd.DetectMult.

***Length***
Размер с учётом фиксированного размера заголовка (24) и Authentication Section.

***My Discriminator***
bfd.LocalDiscr.

***Your Discriminator***
bfd.RemoteDiscr.

***Desired Min TX Interval***
bfd.DesiredMinTxInterval.

***Required Min RX Interval***
bfd.RequiredMinRxInterval.

***Required Min Echo RX Interval***
Значение минимального требуемого интервала для пакетов Echo в данной сессии. Значение 0 указывает, что локальная система не хочет или не может возвращать пакеты BFD Echo удалённой системе и та не будет передавать Echo.

***Authentication Section***
Включается и заполняется в соответствии с правилами параграфа 6.7, если применяется аутентификация (bfd.AuthType отлично от 0). В противном случае этот раздел не включается.

### *6.8.8. Приём пакетов BFD Echo*

Полученный пакет BFD Echo **должен** демультиплексироваться в соответствующую сессию для обработки. **Должны** быть реализованы способы обнаружения отсутствующих пакетов Echo, которые скорей всего включаются в обработку полученных пакетов Echo. В остальном обработка Echo выходит за рамки этой спецификации.

### *6.8.9. Передача пакетов BFD Echo*

Пакеты BFD Echo **недопустимо** передавать при bfd.SessionState, отличном от Up. **Недопустимо** передавать BFD Echo, пока последний пакет BFD Control от удалённой системы содержит отличное от 0 значение Required Min Echo RX Interval.

Пакеты BFD Echo **можно** передавать при bfd.SessionState = Up. **Недопустима** передача BFD Echo с интервалом меньше значения, анонсированного удалённой системой Required Min Echo RX Interval, за исключением указанного ниже.

> **Может** быть применена вариация до 25% и фактический интервал **может** составлять от 75% до 100% анонсированного значения. Одиночный пакет BFD Echo **можно** передать между обычными интервалами плановой передачи Echo.

В остальном данная спецификация передачу пакетов BFD Echo не задаёт.

### *6.8.10. Изменение интервала Min Rx*

Когда желательно изменить скорость получения пакетов BFD Control от удалённой системы, можно в любой момент установить для bfd.RequiredMinRxInterval любое значение. Это значение будет передано в следующем исходящем пакете Control и удалённая система изменит свои настройки. Требования указаны в параграфе 6.8.3.

### *6.8.11. Изменение интервала Min Tx*

Когда желательно изменить скорость передачи пакетов BFD Control удалённой системе (в соответствии с её требованиями), можно в любой момент установить для bfd.DesiredMinTxInterval любое значение. Применяются правила параграфа 6.8.3.

### *6.8.12. Обнаружение смены коэффициента*

Когда желательно изменить коэффициент обнаружения, можно установить для bfd.DetectMult любое значение кроме 0. Новое значение будет передано в следующем пакете BFD Control и использовать Poll Sequence не требуется. Дополнительные требования указаны в параграфе 6.6.

### *6.8.13. Включение и отключение функции Echo*

Если желательно запустить или остановить передачу пакетов BFD Echo, это **можно** сделать в любой момент (с учётом требований к передаче, указанных в параграфе 6.8.9).

Если желательно разрешить или запретить возврат полученных пакетов BFD Echo, это **можно** сделать в любой момент, устанавливая в Required Min Echo RX Interval исходящих пакетов BFD Control 0 или иное значение.

### *6.8.14. Включение и отключение режима Demand*

Если желательно включить или отключить режим Demand, это **можно** сделать в любой момент установкой подходящего значения bfd.DemandMode. Впоследствии режим Demand активируется в соответствии с правилами параграфа 6.6.

Если на удалённой системе режим Demand больше не активен, локальная система **должна** начать передачу периодических пакетов BFD Control, как описано в параграфе 6.8.7.

### 6.8.15. Сброс плоскости пересылки

Когда плоскость пересылки в локальной системе сбрасывается (перезапускается) по той или иной причине так, что удалённая система больше не может полагаться на локальный статус пересылки, локальная система должна установить bfd.LocalDiag = 4 (Forwarding Plane Reset) и bfd.SessionState = Down.

### 6.8.16. Административный контроль

При некоторых обстоятельствах может быть желательно административно включить или отключить сессию BFD. Для этого **должна** выполняться приведённая ниже процедура:

Если сессия включается,

устанавливается bfd.SessionState = Down.

Иначе

Устанавливается bfd.SessionState = AdminDown;

Устанавливается подходящее значение bfd.LocalDiag;

Прекращается передача пакетов BFD Echo.

Если извне BFD получен сигнал об отказе базового пути, реализация **может** административно отключить сессию с диагностическим кодом Path Down. В других случаях можно использовать диагностический код Administratively Down.

Пакеты BFD Control **следует** передавать в течение интервала не менее Detection Time после перехода в состояние AdminDown, чтобы удалённая система узнала о смене статуса. Пакеты BFD Control **можно** передавать неопределённо долго после перехода в состояние AdminDown для поддержки статуса каждой системы (см. параграф 6.8.18).

### 6.8.17. Конкатенация путей

Если путь, отслеживаемый BFD, объединён (конкатенация) с другими путями (цепочка, создающая сквозной путь), может быть желательным распространение сведений об отказе на одном из путей цепочки по всей сессии BFD (обеспечение взаимодействие функции мониторинга живучести между BFD и другими технологиями). Для этого определено два диагностических кода Concatenated Path Down (отказ на объединённом пути) и Reverse Concatenated Path Down (отказ на объединённом пути возврата). Первый служит для распространения сведений об отказе в направлении к взаимодействующей системе, а второй - в обратном направлении (в предположении двухстороннего пути).

Система **может** сигнализировать об одном из этих отказов, просто устанавливая в bfd.LocalDiag нужный код диагностики. Сеанс BFD при этом не прерывается. Если в удалённой системе не активен режим Demand, других действий не требуется, поскольку код диагностики будет передаваться в периодических пакетах BFD Control. При активном режиме Demand (в удалённой системе) локальная система не передаёт периодических пакетов BFD Control и **должна** инициироваться последовательность Poll Sequence, чтобы передать код диагностики. Если впоследствии возникнет отказ в сессии BFD, код диагностики будет заменён кодом, указывающим причину отказа. Агент взаимодействия должен снова выполнить описанную выше процедуру после перехода сессии BFD в состояние Up, если нужно возобновить распространение отказов через конкатенацию путей.

### 6.8.18. Удержание отключённых сессий

Система может отказаться от создания сессии BFD, например, в целях управления скоростью создания таких сессий. Это можно сделать путём удержания сессии с состоянии Down или AdminDown (что лучше подходит).

Имеется два связанных механизма, помогающих решить эту задачу. Во-первых, от системы **требуется** поддерживать состояние сессии (включая временные параметры), даже если сессия отключена (down), пока не пройдёт интервал Detection Time без получения пакетов BFD Control. Это означает, что система может отключить сессию и передать сколь угодно большое значение в поле Required Min RX Interval для управления скоростью получения пакетов. Кроме того, система **может** передать поле Required Min RX Interval = 0, показывающее, что удалённой системе не следует передавать пакеты.

Пока локальная система продолжает передачу пакетов BFD Control, удалённая система обязана соблюдать значение, переданное в поле Required Min RX Interval. Если удалённая система не получит ни одного пакета BFD Control в интервале Detection Time, ей **следует** установить для bfd.RemoteMinRxInterval начальное значение 1 (в соответствии с параграфом 6.8.1, поскольку больше не нужно поддерживать прежнее состояние сессии), а затем она может передавать пакеты со своей скоростью.

## 7. Эксплуатационные вопросы

Скорей всего, BFD будет развёртываться как важная часть сетевой инфраструктуры, поэтому следует соблюдать осторожность, дабы избежать нарушений. Очевидно, что любой механизм, блокирующий пакеты BFD (например, межсетевой экран и средства применения политики), будет вызывать отказ BFD.

Механизмы управления планированием пакетов, такие как ограничители, формователи трафика, очереди с приоритетом и т. п., могут влиять на работу BFD, если Detection Time имеет значение близкое к параметрам планирования пакетов или интервалу их приёма. Время доставки пакетов BFD связано с величиной Detection Time, поэтому может потребоваться его учёт при реализации и развёртывании, особенно в случае малых значений Detection Time.

При использовании BFD через несколько интервалов пересылки (hop), **должны** быть реализованы механизмы контроля перегрузок и при обнаружении перегрузки реализация BFD **должна** снижать объем создаваемого трафика. Выбор механизма для этого выходит за рамки спецификации, а требования к нему могут зависеть от способа развёртывания BFD и взаимодействия протокола с другими частями системы (например, экспоненциальное снижение скорости может быть неприемлемо в случае тесного взаимодействия протоколов маршрутизации с BFD).

Отметим, что явление насыщения (congestion) связано не только с трафиком, но и с вычислениями. Оно может возникать в системах с большим числом сессий BFD и/или очень короткими межпакетными интервалами в форме зависимости от процессора. Поэтому **следует** применять механизм контроля перегрузок для для сессий с одним узлом пересылки (hop), чтобы предотвратить катастрофический коллапс системы (такие события неоднократно наблюдались в других протоколах на основе сообщений Hello).

Механизмы обнаружения перегрузки выходят за рамки этой спецификации, но могут включать обнаружение потери пакетов BFD Control (например, по пропуску порядковых номеров или отказу сессий BFD) и иные средства.

Механизмом снижения уровня трафика BFD является управление скоростью передачи пакетов на локальной и удалённой стороне с помощью полей Min RX Interval и Min TX Interval. Следует отметить, что любой механизм, увеличивающий интервал приёма или передачи, будет увеличивать значение Detection Time для данной сессии.

Одна сессия BFD не отнимает много пропускной способности. Энергичная сессия со временем обнаружения 50 мсек при использовании интервала в 16,7 мсек и коэффициента обнаружения 3 будет генерировать 60 пакетов в секунду. Максимальный размер пакетов в линии составляет примерно 100 байтов, что в сумме создаёт поток примерно 48 кбит/с для каждого направления.

# 8. Взаимодействие с IANA

Этот документ определяет 2 реестра, администрируемых IANA. Первый содержит коды диагностики и называется BFD Diagnostic Codes (4.1. Базовый формат BFD Control). Исходные значения кодов представлены в таблице. Выделение дополнительных кодов происходит по процедуре Expert Review [IANA-CONSIDERATIONS]. Запись реестра включает имя BFD Diagnostic Code и значение кода.

| ***Значение*** | ***Имя диагностического кода BFD*** | |
|---|---|---|
| 0 | No Diagnostic | Нет диагностики |
| 1 | Control Detection Time Expired | Истекло время обнаружения |
| 2 | Echo Function Failed | Отказ функции Echo |
| 3 | Neighbor Signaled Session Down | Сосед сообщил об отказе сессии |
| 4 | Forwarding Plane Reset | Сброс плоскости пересылки |
| 5 | Path Down | Отказ пути |
| 6 | Concatenated Path Down | Отказ конкатенации путей |
| 7 | Administratively Down | Административное отключение |
| 8 | Reverse Concatenated Path Down | Отказ обратной конкатенации путей |
| 9-31 | Unassigned | Резерв на будущее |

Второй реестр содержит идентификаторы типов аутентификации и называется BFD Authentication Types (4.1. Базовый формат BFD Control). Исходные значения кодов представлены в таблице. Выделение дополнительных кодов происходит по процедуре Expert Review [IANA-CONSIDERATIONS]. Запись реестра включает имя BFD Authentication Type Code и значение идентификатора типа.

| ***Значение*** | ***Имя типа аутентификации BFD*** | |
|---|---|---|
| 0 | Reserved | Резерв |
| 1 | Simple Password | Простой пароль |
| 2 | Keyed MD5 | MD5 с ключом |
| 3 | Meticulous Keyed MD5 | Скрупулёзный MD5 с ключом |
| 4 | Keyed SHA1 | SHA1 с ключом |
| 5 | Meticulous Keyed SHA1 | Скрупулёзный SHA1 с ключом |
| 6-255 | Unassigned | Резерв на будущее |

# 9. Вопросы безопасности

Поскольку протокол BFD может быть тесно связан со стабильностью сетевой инфраструктуры (например, протоколов маршрутизации), влияние атак на сессии BFD может быть очень серьёзным. Канал может быть ложно объявлен не работающим или работающим и в обоих случаях будет возникать отказ в обслуживании.

Атакующий с полным контролем над каналом между системами легко может отбросить все пакеты BFD, пересылая все остальные пакеты (канал станет ложно не работающим), или пересылать лишь пакеты BFD, не пропуская остального (канал становится ложно работающим). Такие атаки протокол BFD не может предотвратить.

Для снижения угроз от атакующих с меньшими возможностями в BFD имеется два механизма предотвращения подмены подмены пакетов BFD Control. Обобщенный механизм защиты (Generalized TTL Security Mechanism) [GTSM] использует поле TTL или Hop Count для предотвращения возможности подмены пакетов злоумышленником вне канала передачи. Раздел Authentication Section обеспечивает проверку подлинности пакетов BFD Control. Эти механизмы более подробно рассмотрены ниже.

Когда сессия BFD организована через один канал (физический канал или защищённый туннель, такой как IPsec), значение TTL или Hop Count при передаче **должно** устанавливаться на максимум при передаче с проверкой наличия максимального значения на приёмной стороне и отбрасыванием несоответствующих пакетов. Описание этого метода представлено в [GTSM]. При работе BFD через узлы пересылки (hop) или незащищённый туннель (например, GRE[1]) **следует** применять Authentication Section.

Уровень защиты, обеспечиваемый Authentication Section, зависит от выбранного типа аутентификации. Для простой парольной защиты (Simple Password) уровень определяется сложностью пароля и этот метод следует применять лишь в тех случаях, когда сессия BFD организуется в инфраструктуре, где перехват пакетов невозможен. Основным преимуществом этого метода является минимизация вычислительных издержек при проверке подлинности.

Аутентификация Keyed MD5 сильнее простой парольной защиты, поскольку ключи невозможно узнать из перехваченных пакетов. Однако метод уязвим для replay-атак в интервале, когда порядковый номер не меняется. Порядковые номера могут увеличиваться сколь угодно редко (или часто) для компромисса между строгостью проверки и вычислительными издержками.

[1]Generic Routing Encapsulation - базовая инкапсуляция маршрутных данных.

Механизм Meticulous Keyed MD5 ещё надёжней, поскольку требует увеличения порядкового номера в каждом пакете. Это снижает уровень уязвимости к replay-атакам, поскольку порядковый номер требуется увеличивать в каждом пакете, размер окна допустимых номеров мал, а начальный номер определяется случайно. Здесь сохраняется возможность атаки в начале сессии, пока определяется порядковый номер. Схема аутентификации требует расчёта MD5 для для каждого передаваемого и принимаемого пакета.

Считается, что применение SHA1 обеспечивает более сильную защиту, нежели MD5. Все замечания в части MD5 в этом разделе применимы и к SHA1.

Методы Keyed MD5/SHA1 и Meticulous Keyed MD5/SHA1 применяют конструкцию «секретного суффикса» (secret suffix или append only), в которой общий секретный ключ добавляется в конце данных при расчёте хэш-значения, вместо более распространённой конструкции с хэшированным кодом аутентификации сообщения (Hashed Message Authentication Code или HMAC) [HMAC]. Конструкция HMAC считается подходящей для BFD, но создателям дополнительных механизмов аутентификации для BFD рекомендуется прочесть [HMAC] и приведённые там ссылки.

Если обе системы выбирают случайные значения Local Discriminator в начале сеанса, это дополнительно ослабляет replay-атаки, независимо от типа применяемой аутентификации. Поскольку Local Discriminator можно поменять в любой момент в процессе работы сессии, это также помогает смягчить атаки.

Влияние использования пакетов BFD Echo на безопасность зависит от способа определения этих пакетов, поскольку их структура имеет локальную значимость для передающей системы и это выходит за рамки спецификации. Поскольку пакеты Echo задаёт и обрабатывает лишь передающая система, применение криптографической аутентификации не гарантирует, что другая система действительно жива. Атакующий может завернуть пакеты Echo назад, не зная секретного ключа, что может создать ложное представление о работоспособности канала. Это можно смягчить применением подходящего интервала для пакетов BFD Control. Для пакетов BFD Echo можно также применять [GTSM], хотя значение TTL/Hop Count будет уменьшаться на 1 при «отражении» пакета, что оставляет возможность для подмены.

# *10. Литература*

## 10.1. Нормативные документы

[GTSM] Gill, V., Heasley, J., Meyer, D., Savola, P., Ed., and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, October 2007.

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

[MD5] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[SHA1] Eastlake 3rd, D. and P. Jones, "US Secure Hash Algorithm 1 (SHA1)", RFC 3174, September 2001.

## 10.2. Дополнительная литература

[HMAC] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[IANA-CONSIDERATIONS] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

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

# *Приложение A. Совместимость с прежними версиями*

Хотя протокол версии 0 (задана в документах Internet-Draft, предшествовавших этому RFC) вряд ли получила широкое распространение, некоторые разработчики могут пожелать совместимости с ней. Для этого можно использовать любой механизм, не меняющий определение протокола, поэтому проблем совместимости возникать не должно.

Описанный здесь механизм будет сходиться к версии 1, если её реализуют обе системы, даже если одна из них будет обновлены с версии 0 в интервале Detection Time. Он будет совместим с системами, реализующими лишь одну версию (или настроенными лишь на одну). Очевидно, что эту функцию не следует применять, если система поддерживает лишь одну версию.

Сессия BFD будет удерживать согласование (negotiation holddown), если она настроена на автоматический выбор версии и сессия только что организована, или сессия будет очищена вручную. Для сессии устанавливается статус AdminDown и версия 1. В процессе удержания который длится до 1 интервала Detection Time, система может передавать пакеты BFD Control как обычно, но будет игнорировать полученные пакеты. По завершении интервала holddown состояние меняется на Down и возобновляется обычная работа.

Когда система не находится в режиме удержания (holddown), выполняется автоматическое определение версии и в настоящий момент установлена версия 1 при получении любого пакета версии 0 она сразу же переключится на версию 0. если в данный момент используется версия 0, получение пакета версии 1, указывающего, что сосед имеет состояние AdminDown, система переключается на версию 1. Если система с версией 0 получает пакет версии 1, указывающий отличное от AdminDown состояние соседа, пакет отбрасывается (в соответствии со спецификацией).

Если используемая версия меняется, сессия закрывается в соответствии с новой версией (состояние Down для версии 1 и Failing для версии 0).

# *Приложение B. Участники работы*

Kireeti Kompella и Yakov Rekhter из Juniper Networks также внесли значительный вклад в этот документ.

# *Приложение C. Благодарности*

Толчком для создания этого документа был документ Kireeti Kompella «Protocol Liveness Protocol», который эта спецификация заменила.

Источником для режима Demand послужил документ «A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers», созданный G. Huang и другими.

Авторы благодарны Mike Shand, John Scudder, Stewart Bryant, Pekka Savola, Richard Spencer, Pasi Eronen за их существенный вклад в работу.

Спасибо также Owen Wheeler за поддержку телеконференций между авторами и разными производителями для решения вопросов реализации и разъяснений.

### 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