Network Working Group
Request for Comments: 5440
Category: Standards Track

JP. Vasseur, Ed.
Cisco Systems
JL. Le Roux, Ed.
France Telecom

March 2009

Коммуникационный протокол PCE (PCEP)

Path Computation Element (PCE) Communication Protocol (PCEP)

Статус документа

Этот документ является проектом стандарта Internet (Internet Standards Track) и служит приглашением к дискуссии и внесению предложений с целью совершенствования протокола. Информацию о состоянии стандартизации и статусе протокола можно найти в текущей редакции документа «Internet Official Protocol Standards» (STD 1). Документ может распространяться свободно.

Авторские права

Тезисы

Этот документ задаёт протокол взаимодействия элементов расчёта пути PCE (PCEP) для коммуникаций между клиентами PCC и PCE или между парой PCE. Такие взаимодействия включают запросы расчёта пути и отклики на них, а также уведомления о конкретных состояниях, относящиеся к использованию PCE в контексте MPLS[1] и организации трафика GMPLS[2]. Протокол PCEP разработан с учётом обеспечения гибкости и расширяемости для того, чтобы можно было легко добавлять сообщения и объекты, поэтому в будущем возможно определение новых требований.

Оглавление

[1]Multiprotocol Label Switching - многопротокольная коммутация по меткам.
[2]Generalized MPLS Traffic Engineering - обобщенная многопротокольная коммутация по меткам.

1. Введение

В [RFC4655] описана мотивация и архитектура элемента расчёта пути (PCE) на основе моделей расчёта MPLS и GMPLS TE LSP. Модель позволяет отделить PCE от клиента расчёта путей (PCC), а также разрешает кооперацию PCE. Для этого нужен коммуникационный протокол между PCC PCE, а также между PCE. В [RFC4657] приведены базовые требования к такому протоколу, включая требование использования единого протокола для коммуникаций между PCC и PCE, а также между PCE. Зависящие от приложений требования (например, для коммуникаций внутри области или автономной системы и т. п.) не включены в [RFC4657], но указно требование простой расширяемости протокола для удовлетворения требований, задаваемых документами для конкретных приложений. Примерами таких документов служат [RFC4927], [RFC5376] и [INTER-LAYER].

В этом документе описан протокол расчёта элемента пути (PCEP[1]) для коммуникаций между PCC и PCE или между парой PCE в соответствии с [RFC4657]. Такие взаимодействия включают запросы расчёта пути и отклики на эти запросы, а также уведомления о конкретных состояниях, связанных с использованием PCE в контексте организации трафика MPLS и GMPLS.

Протокол PCEP разработан с учётом гибкости и расширяемости для обеспечения простого добавления в будущем сообщений и объектов.

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 интерпретируются в соответствии с RFC 2119 [RFC2119].

2. Терминология

Ниже приведены определения терминов, используемых в документе.

AS
: Автономная система.

Explicit path - явный путь
: Полностью явный путь от старта до получателя, состоящий из строгого списка интервалов (hop), которыми могут служить абстрактные узлы, такие как AS.

IGP area - область IGP
: Область OSPF или уровень IS-IS.

Inter-domain TE LSP - междоменный TE LSP
: TE LSP, проходящий по меньшей мере через два разных домена, где домены могут быть областями IGP, автономными системами или суб-AS (конфедерация BGP).

PCC
: Клиент расчёта пути - любое клиентское приложение, запрашивающее расчёт пути у PCE.

PCE
: Элемент расчёта пути - объект (компонента, приложение или узел сети), способный рассчитать сетевой путь или маршрут на основе графа сети с учётом дополнительных ограничений.

PCEP Peer
: Элемент, участвующий в сессии PCEP (т. е. PCC или PCE).

TED
: База данных организации трафика, содержащая информацию о топологии и ресурсах домена. TED может обеспечиваться расширениями IGP и, возможно, другими способами.

TE LSP
: Путь с коммутацией по меткам и организацией трафика.

Strict/loose path
: Сочетание строгих и свободных интервалов (hop), включающее хотя бы один свободный интервал, представляющий адресата (интервалом может быть абстрактный узел, такой как AS).

В этом документе при описании взаимодействия между PCE запрашивающий PCE играет роль PCC. Это позволяет сократить объем документа без потерь.

Формат сообщений в документе задаётся кодом Бэкуса-Наура (BNF), описанным в [RBNF].

3. Допущения

В [RFC4655] описаны разные типы PCE. Протокол PCEP не принимает каких-либо допущений и не вносит ограничений на природу PCE.

Кроме того, предполагается, что PCE имеет требуемую информацию (обычно включает топологию сети и данные о ресурсах) для расчёта пути TE LSP. Такая информация может собираться протоколами маршрутизации и некоторыми иными способами. Способы сбора информации выходят за рамки этого документа.

Не делается предположений о методах, применяемых PCC для обнаружения набора PCE (например, статическая настройка или динамическое детектирование), и алгоритмах выбора PCE. В [RFC4674] определён список требования для динамического обнаружения PCE, а основанные на IGP решения для этого указаны в [RFC5088] и [RFC5089].

4. Обзор архитектуры протокола (модель)

Целью этого раздела является описание модели PCEP в стиле [RFC4101]. Представлен архитектурный обзор (картина в целом) протокола, а детали описаны в последующих разделах.

[1]Path Computation Element Protocol.

4.1. Постановка задачи

Основанная на PCE архитектура расчёта путей для MPLS и GMPLS TE LSP описана в [RFC4655]. При раздельном размещении PCC и PCE нужен протокол для их взаимодействия. PCEP является протоколом, разработанным специально для связи между PCC и PCE или между парой PCE в соответствии с [RFC4657] - PCC может применять PCEP для отправки запросов расчёта пути одного или множества TE LSP элементу PCE, а тот может возвращать набор рассчитанных путей, если одни или несколько найденных путей отвечают заданному набору ограничений.

4.2. Обзор архитектуры протокола

PCEP работает на базе транспорта TCP, который удовлетворяет требованиям к надёжной доставке сообщений и управлению потоком данных без дополнительных усилий протокола.

Определено несколько типов сообщений PCEP, перечисленных ниже.

- Open и Keepalive служат для организации и поддержки сессии PCEP, соответственно.
- PCReq передаётся PCC элементу PCE для запроса расчёта пути.
- PCRep передаётся PCE клиенту PCC для ответа на запрос расчёта пути. Сообщение может содержать набор рассчитанных путей, если запрос был выполнен, или отрицательный отклик в противном случае (этот отклик может указывать причины, по которым путь не был найден).
- PCNtf передаётся от PCC к PCE или обратно для уведомления о конкретном событии.
- PCErr передаётся при возникновении протокольной ошибки.
- Close служит для завершения сессии PCEP.

Набор доступных PCE можно статически задать в конфигурации PCC или определять динамически. Механизмы обнаружения и выбора PCE выходят за рамки этого документа.

PCC может иметь сессии PCEP со множеством PCE, а PCE может иметь сессии PCEP со множеством PCC.

Каждое сообщение PCEP рассматривается как единый блок передачи и чередование частей сообщений **недопустимо**. Например, PCC, передавший сообщение PCReq и желающий закрыть сессию должен завершить отправку запроса и лишь после этого передавать сообщение Close.

4.2.1. Фаза инициализации

Фаза инициализации состоит из 2 последовательных этапов, схематически представленных на рисунке 1.

1) Организация соединения TCP (3-этапное согласование) между PCC и PCE.
2) Организация сессии PCEP через соединение TCP.

После организации соединения TCP клиент PCC и элемент PCE (партнёры PCEP) инициируют организацию сессии PCEP с согласованием различных параметров этой сессии. Параметры передаются в сообщениях Open и включают таймер Keepalive, DeadTimer, а также могут подробно указывать возможности и правила, задающие условия, при которых запросы расчёта пути могут передаваться элементу PCE. Если при организации сессии PCEP возник отказ по причине несогласия партнёров PCEP по части параметров сессии или отсутствия ответа одного из партнёров в течение времени на организацию сессии, соединение TCP незамедлительно разрывается. Последующие попытки разрешены, но реализации следует использовать экспоненциально нарастающие интервалы между такими попытками.

Сообщения о сохранении активности (Keepalive) служат для подтверждения сообщений Open, а также передаются после организации сессии PCEP. Между данной парой партнёров PCEP в каждый момент может существовать лишь одна сессия PCEP и лишь одно соединение TCP через порт PCEP.

Описания сообщений Open и Keepalive приведены в параграфах 6.2 и 6.3, соответственно.

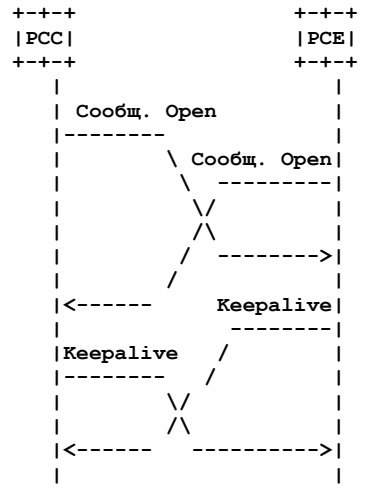

Рисунок 1. Фаза инициализации PCEP (запускается PCC).

Отметим, что после организации сессии PCEP обмен сообщениями Keepalive не является обязательным.

4.2.2. Сообщения о сохранении активности

После организации сессии PCE или PCC может захотеть получать информацию о доступности своего партнёра PCEP.

Можно использовать для этого TCP, но при этом возможны отказы удалённого партнёра PCEP без нарушения соединения TCP. Можно также полагаться на встроенные механизмы реализации TCP, но это может не давать

уведомлений об отказах достаточно быстро. Наконец, PCC может ждать отправки запроса на расчёт пути и использовать свои неудачные попытки передачи или отсутствие отклика в качестве сигнала об отказе сессии, но это явно не будет эффективно.

Для таких случаев в PCEP предусмотрен механизм информирования на основе таймеров Keepalive и DeadTimer, а также сообщений Keepalive.

На каждой стороне сессии PCEP запускается таймер Keepalive, который сбрасывается при отправке любого сообщения в сессии. По завершении отсчёта таймера передаётся сообщение Keepalive. Другой трафик также служит подтверждением активности (см. параграф 6.3).

Участники сессии PCEP запускают также DeadTimer, который сбрасывается при получении в сессии любого сообщения. Если в интервале DeadTimer не было получено ни одного сообщения, сессия считается «умершей».

Отметим, что на сообщения Keepalive не требуется отвечать и они не являются частью двухстороннего согласования, как в других протоколах. Этот механизм предназначен для минимизации трафика сохранения активности в сессии.

Трафик сохранения активности в сессии может оказаться несбалансированным по причине разных требований сторон. Каждый из партнёров может указать в сообщении Open значения таймеров Keepalive (интервал отправки сообщений Keepalive при отсутствии другого трафика) и DeadTimer (т. е. интервал отсутствия трафика, после которого сессия считается «мёртвой»), рекомендуемые для партнёра. Стороны могут применять разные значения таймера Keepalive.

Минимальное значение таймера Keepalive составляет 1 секунду и указывается в секундах. Рекомендуется интервал 30 секунд. Таймер можно отменить, установив для него нулевое значение.

Для таймера DeadTimer рекомендуется устанавливать по умолчанию значение, превышающее в 4 раза значение таймера Keepalive у партнёра. Это предотвращает перегрузку соединения TCP избыточными сообщениями Keepalive.

4.2.3. Запрос расчёта пути от PCC к PCE

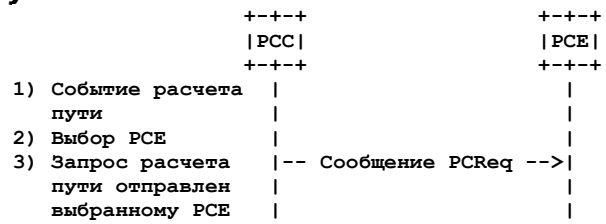

Рисунок 2. Запрос расчета пути.

После того, как PCC организовал сессию PCEP с одним или несколькими PCE, если инициируется событие, требующее расчёта набора путей, PCC сначала выбирает один или несколько PCE. Отметим, что выбор PCE может быть сделан до организации сессии PCEP.

Как только PCC выбрал элемент PCE, он передаёт тому запрос расчёта пути (сообщение PCReq), содержащий различные объекты, задающие набор ограничений на расчёт пути. Например, «рассчитать путь TE LSP от IP=x.y.z.t, до IP=x'.y'.z'.t' с пропускной способностью B Мбит/с, приоритетом Setup/Holding=P, ...» В дополнение PCC может указать важность запроса, задав для него приоритет. Каждый запрос однозначно указывается номером (request-id) и парой адресов PCC-PCE. Процесс схематически показан на рисунке 2. Отметим, что PCC может в любой момент отправить PCE множество запросов расчёта путей.

Описание сообщения PCReq дано в параграфе 6.4.

4.2.4. Отклик на запрос расчёта пути от PCE к PCC

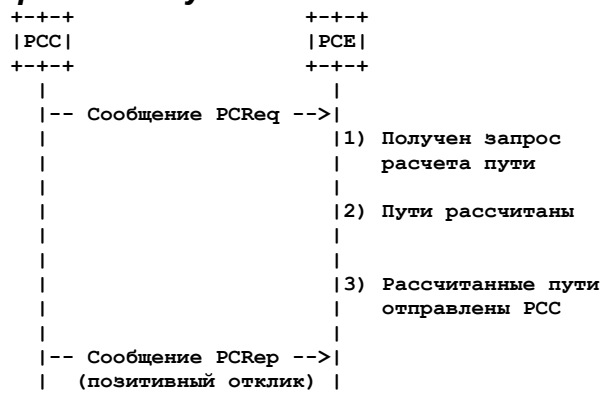

Рисунок 3a. Успешный расчет пути.

При получении запроса на расчёт пути от PCC элемент PCE запускает расчёт, который может дать 2 варианта.

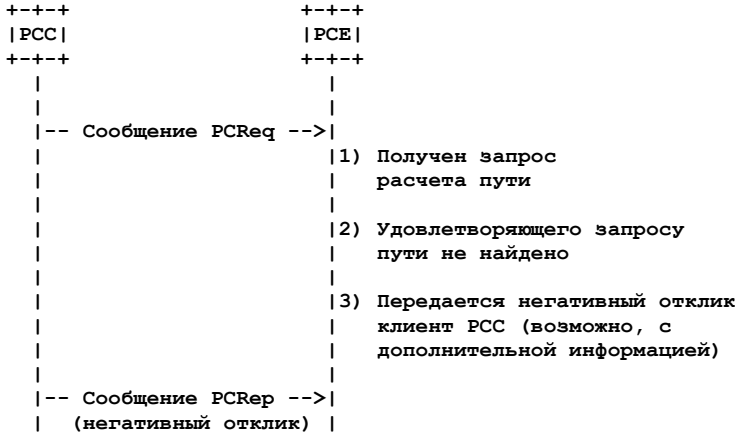

Рисунок 3b. Неудачный расчет пути.

- Позитивный (рисунок 3a) - PCE выполняет расчёт пути, удовлетворяющего заданному набору ограничений. В этом случае PCE возвращает набор рассчитанных путей запрашивающему PCC. Отметим, что PCEP поддерживает возможность передачи запросов, которые дают множество путей (например, расчёт набора путей с разными каналами).
- Негативный (рисунок 3b) - не найдено пути, соответствующего заданным ограничениям. В этом случае PCE может указать набор ограничений, которые не удалось выполнить. При получении негативного отклика PCC может передать обновленный запрос или предпринять иные действия.

Сообщение PCRep описано в параграфе 6.5.

4.2.5. Уведомление

При некоторых обстоятельствах PCE может захотеть уведомить PCC о конкретном событии. Предположим, например, что PCE внезапно перегружается, что может вести к неприемлемой задержке откликов. PCE может захотеть уведомить одного или множество клиентов PCC о том, что некоторые их запросы (указываются в уведомлении) не будут выполнены или могут столкнуться с неприемлемыми задержками. При получении такого уведомления PCC может направить запрос другому PCE (при наличии). Аналогично, PCC может уведомлять PCE о таких событиях, как отмена ожидающего запроса.

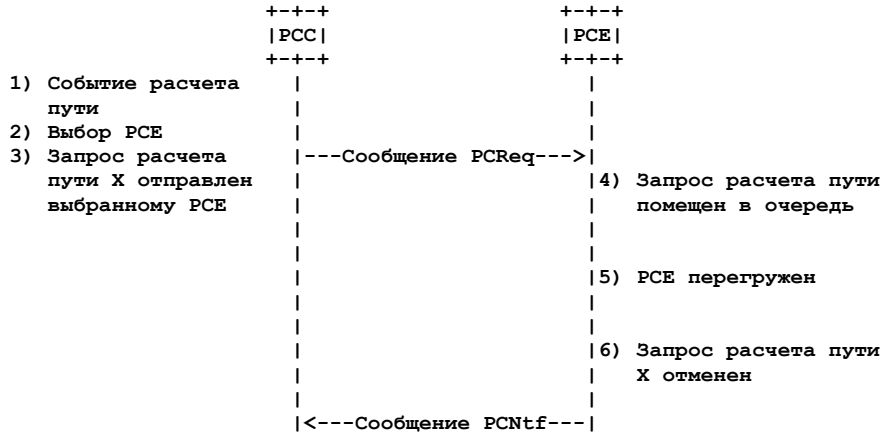

Рисунок 4. Пример передачи уведомления PCE (отказ) клиенту PCC.

Описание сообщения PCNtf дано в параграфе 6.6.

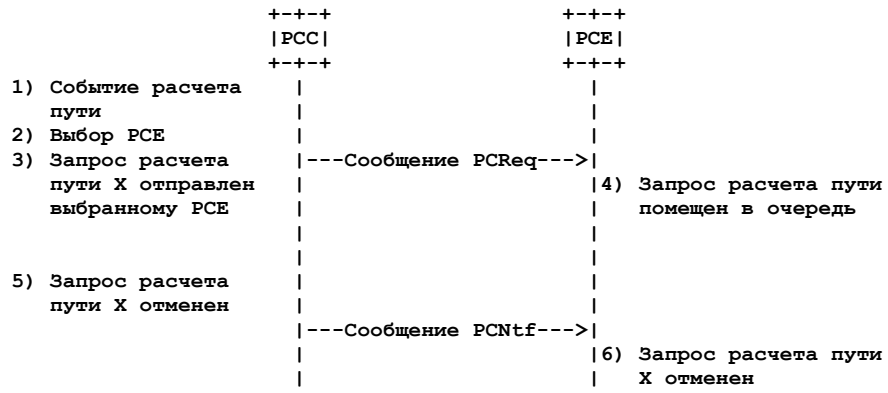

Рисунок 5. Пример передачи уведомления PCC (отмена) элементу PCE.

4.2.6. Сообщение об ошибке

Сообщение об ошибке PCEP (PCErr) передаётся при возникновении условий, считающихся ошибкой протокола, или при получении запроса, не соответствующего спецификации PCEP (например, не поддерживаемая возможность, отсутствие обязательного объекта, нарушение правил, неожиданное сообщение, ссылка на неизвестный запрос).

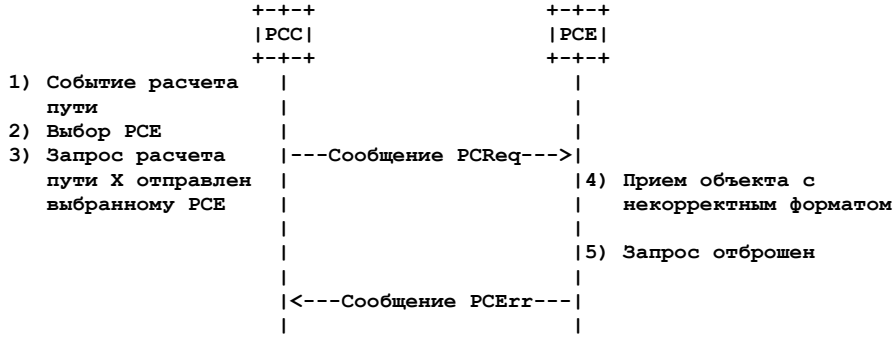

Рисунок 6. Пример сообщения об ошибке, переданного PCE клиенту PCC.

Сообщение PCErr описано в параграфе 6.7.

4.2.7. Разрыв сессии PCEP

Когда один из партнёров желает прервать сессию PCEP, он сначала передаёт сообщение Close, а затем разрывает соединение TCP. Если сессия прерывается PCE, клиент PCC сбрасывает все состояния, относящиеся к ожидающим запросам, которые были переданы этому PCE. Если сессию прерывает PCC, элемент PCE сбрасывает все ожидающие запросы от этого PCC, а также связанные с ним состояния. Сообщение Close может передаваться для завершения сессии PCEP лишь в том случае, когда такая сессия была организована. При разрыве соединения TCP сессия PCEP прерывается незамедлительно.

Сообщение Close описано в параграфе 6.8.

4.2.8. Краткосрочные и постоянные сессии PCEP

Реализация может принять решение о сохранении сессии PCEP (и соответствующего соединения TCP) на неограниченное время (например, это может быть целесообразно при частой отправке запросов на расчёт пути, чтобы снизить издержки, связанные с организацией соединения TCP для каждого запроса, которые ведут к дополнительным задержкам). И наоборот, в некоторых случаях может быть желательно создавать и завершать сессии PCEP для каждого запроса (например, при редких запросах на расчёт пути).

5. Транспортный протокол

PCEP работает на основе TCP, используя зарегистрированный порт TCP (4189). Это позволяет выполнить требования гарантированной доставки и управления потоком данных без участия протокола. Все сообщения PCEP **должны** передаваться с использованием зарегистрированного порта TCP у отправителя и получателя.

6. Сообщения PCEP

Сообщение PCEP состоит из базового заголовка, за которым следует тело переменного размера с набором объектов, среди которых могут быть обязательные и необязательные. В контексте этого документа объект называется обязательным в сообщении PCEP, когда он **должен** быть включён для того, чтобы объект стал действительным. Сообщение PCEP без обязательного объекта **должно** вызывать генерацию сообщения об ошибке (см. параграф 7.15). Необязательные объекты можно не включать в сообщение.

В базовом заголовке каждого объекта PCEP определён флаг P (см. параграф 7.2). При установленном флаге объекта в сообщении PCReq элемент PCE **должен** принять во внимание данные объекта при расчёте пути. Например, объект METRIC, определённый в параграфе 7.8, позволяет PCC задать границы приемлемой стоимости пути. Объект METRIC является необязательным, но PCC может установить флаг для принятия во внимание данных этого объекта. В этом случае, если ограничения не может быть учтено, PCE **должен** генерировать сообщение об ошибке.

Для каждого типа сообщений PCEP определены правила, указывающие набор объектов, которые могут быть включены в сообщение. Для задания правил применяется форма BNF [RBNF]. Квадратные скобки указывают необязательные последовательности. Реализация **должна** создавать сообщения PCEP с использованием указанного здесь порядка.

6.1. Базовый заголовок

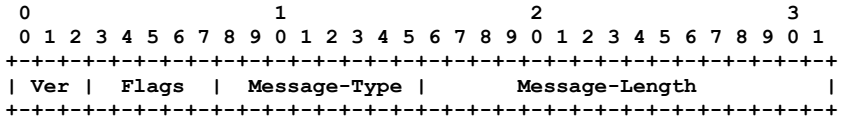

Рисунок 7. Базовый заголовок сообщения PCEP.

Ver (Version - 3 бита)

Номер версии PCEP, 1 для текущей версии.

Flags (5 битов)

Флаги в настоящее время не определены, невыделенные биты считаются резервными. Они **должны** сбрасываться (0) при передаче и **должны** игнорироваться на приёмной стороне.

Message-Type (8 битов)

Определённые в настоящее время типы сообщений указаны в таблице.

Номер	*Назначение*
1	Open
2	Keepalive
3	Path Computation Request
4	Path Computation Reply
5	Notification

6 Error
7 Close

Message-Length (16 битов)

Общий размер сообщения PCEP с учётом базового заголовка (в байтах).

6.2. Сообщение Open

Сообщения Open передаются от PCC к PCE и от PCE к PCC для организации сессии PCEP. Поле Message-Type в базовом заголовке PCEP сообщения Open имеет значение 1.

После организации соединения TCP первым сообщением от PCC к PCE или от PCE к PCC **должно** быть сообщение Open, как указано в приложении A.

Передача любого сообщения до Open **должна** вызывать протокольную ошибку с возвратом сообщения PCErr с Error-Type "PCEP session establishment failure" и Error-value "reception of an invalid Open message or a non Open message", а попытка организации сессии PCEP **должна** прерываться разрывом соединения TCP.

Сообщение Open служит для организации сессии между партнёрами PCEP. На этапе организации сессии партнёры обмениваются некоторыми характеристиками сеанса. Если обе стороны принимают такие характеристики, сессия PCEP организуется. Формат сообщения Open приведён ниже.

```
<Open Message>::= <Common Header>
                  <OPEN>
```

Сообщение Open **должно** включать в точности один объект (параграф 7.3).

В объекте OPEN задаются характеристики сессии. После организации соединения TCP отправитель **должен** запустить таймер инициализации OpenWait, по истечении которого при отсутствии сообщения Open передаётся сообщение PCErr и соединение TCP освобождается (приложение A).

После передачи партнёру сообщения Open отправитель **должен** запустить таймер инициализации KeepWait, по истечении которого при отсутствии сообщения Keepalive или PCErr в случае несогласия с характеристиками сессии **должно** передавать сообщение PCErr, а соединение TCP **должно** быть освобождено (приложение A).

Таймеры OpenWait и KeepWait имеют фиксированное значение (1 минута).

При получении сообщения Open приёмная сторона **должна** определить приемлемость предложенных характеристик сессии PCEP. Если признающий партнёр не готов воспринять хотя бы одну из предложенных характеристик, он **должен** передать сообщение об ошибке. В это сообщение **следует** включить объект OPEN и для каждого неприемлемого параметра сессии **следует** указать в нем подходящее значение. Партнёр PCEP **может** снова передать сообщение Open с другими характеристиками сессии. Если новое сообщение Open имеет такой же набор характеристик или новые характеристики остаются неприемлемыми, принимающая сторона **должна** передать сообщение об ошибке и незамедлительно разорвать соединение TCP. Описание сообщений об ошибках дано в параграфе 7.15. Последующие попытки разрешены, но реализации **следует** экспоненциально увеличивать интервал между повторами.

Если характеристики сессии PCEP устраивают, принимающий партнёр **должен** передать сообщение Keepalive (параграф 6.3), которое служит подтверждением.

Сессия PCEP считается организованной, когда обе стороны получили от партнёра PCEP сообщения Keepalive.

6.3. Сообщение Keepalive

Сообщения Keepalive передаёт PCC или PCE для подтверждения активности сессии. Keepalive служит также откликом на Open, подтверждающим приём сообщения и приемлемость характеристик сессии PCEP. Поле Message-Type в базовом заголовке PCEP сообщения Keepalive имеет значение 2. Сообщения Keepalive не включают объектов.

PCEP имеет свой механизм подтверждения активности сессии PCEP. Это требует задать частоту, с которой каждый из партнёров PCEP передаёт сообщения Keepalive. Партнёры могут выбрать разные значения частоты отправки сообщений. Значение DeadTimer определяет период, по истечении которого партнёр считает сессию PCEP неактивной, если он не получил ни одного сообщения PCEP (Keepalive или другие сообщения), поэтому любое сообщение служит подтверждением активности сессии. Для таймеров DeadTimers партнёры PCEP также не обязаны устанавливать одинаковые значения. Минимальное значение таймера Keepalive составляет 1 секунду. Реализациям **следует** внимательно учитывать влияние малых значений таймера Keepalive, поскольку они приводить к некорректной работе в периоды временной нестабильности сети.

Сообщения Keepalive передаются с частотой, заданной в объекте OPEN из сообщения Open, в соответствии с правилами, заданными в параграфе 7.3. Поскольку любое сообщение PCEP выступает в качестве Keepalive, реализация может передавать сообщения Keepalive с постоянными интервалами, независимо от других сообщений PCEP или определять время отправки Keepalive от последнего сообщения PCEP (не Keepalive).

Отметим, что передача сообщений Keepalive для сохранения сессии не обязательна и партнёры могут не передавать этих сообщений после организации сессии PCEP. В этом случае узел не ожидает сообщений Keepalive и **недопустимо** считать их отсутствие признаком неактивности сессии.

Формат сообщения Keepalive показан ниже.

```
<Keepalive Message>::= <Common Header>
```

6.4. Сообщение PCReq

Сообщение с запросом расчёта пути (Path Computation Request или PCReq) передаётся клиентом PCC элементу PCE для запроса расчёта пути. PCReq может содержать один или несколько запросов расчёта пути. Поле Message-Type в базовом заголовке PCEP для сообщений PCReq имеет значение 3.

В сообщение PCReq **должны** включаться два обязательных объекта - RP и END-POINTS (раздел 7). Если одного или обоих объектов нет, PCE **должен** передать сообщение об ошибке клиенту PCC. Другие объекты не обязательны.

Формат сообщения PCReq показан ниже.

```
<PCReq Message>::= <Common Header>
                   [<svec-list>]
                   <request-list>
```

где

```
<svec-list>::=<SVEC>[<svec-list>]
<request-list>::=<request>[<request-list>]

<request>::= <RP>
             <END-POINTS>
             [<LSPA>]
             [<BANDWIDTH>]
             [<metric-list>]
             [<RRO>[<BANDWIDTH>]]
             [<IRO>]
             [<LOAD-BALANCING>]
```

где

```
<metric-list>::=<METRIC>[<metric-list>]
```

Объекты SVEC, RP, END-POINTS, LSPA, BANDWIDTH, METRIC, RRO, IRO и LOAD-BALANCING определены в разделе 7. Особый случай с двумя объектами BANDWIDTH рассмотрен в параграфе 7.7.

Реализация PCEP вольна обрабатывать полученные запросы в любом порядке. Например, запросы могут обрабатываться в порядке получения, переупорядочиваться в соответствии с локальной политикой или приоритетом, заданным в объекте RP (параграф 7.4.1) или обрабатываться параллельно.

6.5. Сообщение PCRep

Отклик PCEP с расчётом пути (Path Computation Reply или PCRep) передаётся PCE клиенту PCC в ответ на предшествующий запрос PCReq. Поле Message-Type в базовом заголовке PCEP сообщения PCRep имеет значение 4.

Протокол поддерживает связывание откликов на множество запросов расчёта пути в одно сообщение PCRep. Если PCE получает несинхронизированные запросы расчёта пути в одном или нескольких сообщениях PCReq от клиента PCC, он **может** объединить рассчитанные пути в одном сообщении PCRep для снижения нагрузки на уровень управления. Отметим, что оборотной стороной такого подхода является дополнительная задержка откликов на некоторые из запросов. PCE, получивший множество запросов в одном сообщении PCReq, **может** возвратить каждый рассчитанный путь в отдельном сообщении PCRep или вернуть все пути в одном PCRep. Сообщение PCRep может включать позитивные и негативные отклики.

Сообщение PCRep может включать набор рассчитанных путей, соответствующих одному запросу с распределением нагрузки (см. параграф 7.16) или множеству запросов от клиента PCC. Сообщение PCRep может также включать множество доступных путей, соответствующих одному запросу.

Сообщение PCRep **должно** включать по меньшей мере один объект RP. Для каждого отклика, объединяемого в одно сообщение PCReq **должен** включаться объект RP, содержащий идентификатор запроса, идентичный одному из заданных в объекте RP соответствующего сообщения PCReq (см. определение объекта RP в параграфе 7.4).

Если запрос расчёта пути выполнен (т. е. элемент PCE нашёл путь, удовлетворяющий набору ограничений), в сообщение PCRep помещается набор рассчитанных путей, указанных объектами ERO[1], определёнными в параграфе 7.9. Ситуация с предоставлением множества путей в сообщении PCRep подробно рассмотрена в параграфе 7.13. Кроме того, при запросе PCC расчёта набора путей с суммарной пропускной способностью через объект LOAD-BALANCING в сообщении PCReq, за объектом ERO каждого из рассчитанных путей следует объект BANDWIDTH, как указано в параграфе 7.16.

Если расчёт пути не был выполнен, сообщение PCRep **должно** включать объект NO-PATH (параграф 7.5), который может содержать дополнительную информацию (например, причины отказа при расчёте пути).

Формат сообщения PCRep показан ниже

```
<PCRep Message> ::= <Common Header>
                    <response-list>
```

где

```
<response-list>::=<response>[<response-list>]
<response>::=<RP>
             [<NO-PATH>]
             [<attribute-list>]
             [<path-list>]
<path-list>::=<path>[<path-list>]

<path>::= <ERO><attribute-list>
```

где

```
<attribute-list>::=[<LSPA>]
                   [<BANDWIDTH>]
                   [<metric-list>]
                   [<IRO>]

<metric-list>::=<METRIC>[<metric-list>]
```

6.6. Сообщение PCNtf

Уведомление PCEP (Notification или PCNtf) может передавать PCE или PCC для информирования о конкретном событии. Поле Message-Type в базовом заголовке PCEP для сообщения PCNtf имеет значение 5.

[1]Explicit Route Object - объект с явным маршрутом.

В сообщении PCNtf **должен** присутствовать хотя бы один элемент NOTIFICATION и **может** быть несколько таких объектов для уведомления о нескольких событиях. Объект NOTIFICATION определён в параграфе 7.14. Сообщение PCNtf **может** включать объекты RP (параграф 7.4), когда уведомление относится к конкретному запросу расчёта пути.

Сообщение PCNtf может передавать PCC или PCE в ответ на запрос или без запроса.

Формат сообщения PCNtf показан ниже.

```
<PCNtf Message>::=<Common Header>
                  <notify-list>

<notify-list>::=<notify> [<notify-list>]

<notify>::= [<request-id-list>]
             <notification-list>

<request-id-list>::=<RP>[<request-id-list>]

<notification-list>::=<NOTIFICATION>[<notification-list>]
```

6.7. Сообщение PCErr

Сообщения PCEP об ошибках (Error или PCErr) передаются, если возникает протокольная ошибка или запрос не соответствует спецификации PCEP (например, получено сообщение с некорректным форматом, отсутствует обязательный объект, нарушены правила, получено неожиданное сообщение или указан неизвестный запрос). Поле Message-Type в базовом заголовке PCEP сообщения PCErr имеет значение 6.

Сообщение PCErr передаётся PCC или PCE в ответ на запрос или без запроса. При передаче сообщения PCErr в ответ на запрос, оно **должно** включать набор объектов RP, связанных с ожидающими запросами расчёта пути, вызвавшими ошибки. В остальных случаях (передача без запроса) объекты RP не включаются в PCErr. Например, объект RP не включается в сообщение PCErr, когда возникает ошибка на этапе инициализации. Сообщение PCErr **должно** включать объект PCEP-ERROR, заданный для ошибки PCEP. Объект PCEP-ERROR определён в параграфе 7.15.

Формат сообщения PCErr показан ниже.

```
<PCErr Message> ::= <Common Header>
                    ( <error-obj-list> [<Open>] ) | <error>
                    [<error-list>]

<error-obj-list>::=<PCEP-ERROR>[<error-obj-list>]

<error>::=[<request-id-list>]
           <error-obj-list>

<request-id-list>::=<RP>[<request-id-list>]

<error-list>::=<error>[<error-list>]
```

Процедура обработки сообщений PCErr описана в параграфе 7.15.

6.8. Сообщение Close

Сообщение Close передаётся PCC или PCE для завершения текущей сессии PCEP. Поле Message-Type в базовом заголовке PCEP для сообщения Close имеет значение 7. Формат сообщения Close показан ниже.

```
<Close Message>::= <Common Header>
                   <CLOSE>
```

Сообщение Close **должно** содержать в точности один объект CLOSE (параграф 7.17). При наличии нескольких объектов CLOSE обрабатываться **должен** первый, а остальные игнорируются.

При получении сообщения Close принимающий узел PCEP **должен** отменить все ожидающие запросы, он **должен** также закрыть соединение TCP и **недопустима** передача других сообщений PCEP в данной сессии PCEP.

6.9. Приём неизвестных сообщений

Получившая неизвестное сообщение реализация PCEP **должна** передать сообщение PCErr с Error-value=2 (возможность не поддерживается).

Если PCC или PCE получает нераспознанные сообщения число не менее MAX-UNKNOWN-MESSAGES в минуту, он **должен** передать сообщение Close со значением "Reception of an unacceptable number of unknown PCEP message". PCC/PCE **должен** закрывать соединение TCP и **недопустима** передача других сообщений PCEP в данной сессии PCEP. Для MAX-UNKNOWN-MESSAGES **рекомендуется** значение 5.

7. Форматы объектов

Объекты PCEP имеют общий формат и начинаются с базового заголовка (параграф 7.2), за которым следуют поля конкретного объекта, определённые независимо для каждого типа объектов. Объекты могут также включать блоки данных TLV[1], структура которых описана в параграфе 7.1.

7.1. Формат PCEP TLV

Объект PCEP может включать один или несколько необязательных блоков TLV.

Все PCEP TLV имеют показанный ниже формат.

```
Type -   2 байта
Length - 2 байта
```

[1]Type-length-value - тип, размер, значение.

```
   Value -  переменный размер
```

TLV объекта PCEP включает 2-байтовое поле типа, 2 байта размера TLV и поле значения.

Поле Length определяет размер поля значения в байтах. TLV дополняется для выравнивания по 4-байтовой границе. Заполнение не учитывается в поле Length (т. е. для 3-байтового значения поле размера будет содержать значение 3, а общий размер TLV составит 8 байтов).

Не распознанные TLV **должны** игнорироваться.

IANA оправляет пространством идентификаторов типа PCEP Object TLV, как описано в разделе 9.

7.2. Базовый заголовок объекта

Объект PCEP в сообщении PCEP содержит 1 или несколько 32-слов с базовым заголовком, показанным ниже.

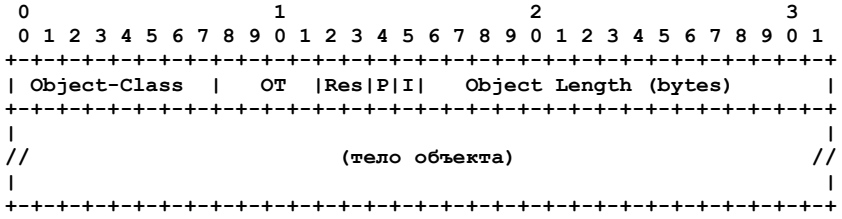

Рисунок 8. Базовый заголовок объекта PCEP.

Object-Class (8 битов)
Указывает класс объекта PCEP.

OT (Object-Type - 4 бита)
Указывает тип объекта PCEP.
Значения поле Object-Class и Object-Type назначаются IANA.
Поля Object-Class и Object-Type однозначно указывают каждый объект PCEP.

Res flags (2 бита)
Резервное поле. **Должно** устанавливаться в 0 при передаче, а на приёмной стороне **должно** игнорироваться.

P flag (Processing-Rule - 1 бит)
Флаг P позволяет PCC указать в сообщении PCReq, передаваемом PCE, нужно ли принимать объект во внимание при расчёте пути или этот объект не обязательно учитывать. При установленном флаге P объект **должен** учитываться PCE, а при сброшенном флаге P элемент PCE может игнорировать объект.

I flag (Ignore - 1 бит)
Флаг I применяется PCE в сообщении PCRep для указания клиенту PCC, был ли обработан необязательный объект. PCE **может** включить игнорируемый объект в отклик и установить флаг I для указания того, что он был проигнорирован. Сброшенный флаг указывает, что необязательный объект был обработан при расчёте пути. Установка флага I для необязательных объектов является лишь информационной и может не применяться. Флаг I не имеет значения в сообщениях PCRep, когда был установлен флаг P в соответствующем запросе PCReq.
Если PCE не понимает объект с флагом P или понимает объект, но решает игнорировать его, все сообщение PCEP **должно** быть отвергнуто, а PCE **должен** передать сообщение PCErr с Error-Type="Unknown Object" или "Not supported Object" вместе с соответствующим объектом RP. Отметим, что при включении в PCReq множества запросов отвергаться **должны** лишь неизвестные/нераспознанные объекты с установленным флагом P.

Object Length (16 битов)
Указывает общий размер объекта (с учётом заголовка) в байтах. Поле Object Length всегда **должно** быть кратно 4 и быть не меньше 4. Максимальный размер объекта составляет 65528 байтов.

7.3. Объект OPEN

Объект OPEN **должен** присутствовать в каждом сообщении Open и **может** включаться в сообщения PCErr. В сообщении **должен** быть лишь один объект OPEN.

Объект OPEN содержит набор полей для указания версии PCEP, частоты Keepalive, DeadTimer и идентификатора сессии PCEP, а также различных флагов. Объект может также включать набор TLV, служащих для передачи различных характеристик сессии, таких как возможности PCE, правила политики и т. п. TLV в настоящее время не определены.

OPEN Object-Class = 1.

OPEN Object-Type = 1.

Формат тела объекта OPEN показан на рисунке.

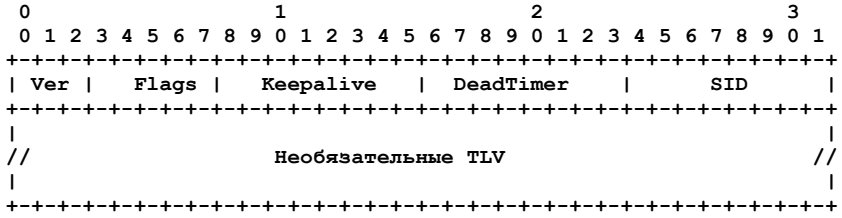

Рисунок 9. Формат объекта OPEN.

Ver (3 бита)
Версия PCEP (1 для текущей версии).

Flags (5 битов)
Флаги в настоящее время не определены. Нераспределенные биты являются резервными. Они **должны** устанавливаться в 0 при передаче, а на приёмной стороне **должны** игнорироваться.

Keepalive (8 битов)
Максимальный интервал времени (в секундах) между двумя последовательными сообщениями PCEP, переданными отправителем этого сообщения. Минимальное значение Keepalive составляет 1 секунду. При

установке значения 0 после организации сессии сообщения Keepalive больше не будут передаваться удалённому партнёру. **Рекомендуемый** интервал составляет 30 секунд.

DeadTimer (8 битов)

Интервал времени, по истечении которого узел PCEP будет считать сессию с отправителем сообщения Open бездействующей (down) при отсутствии сообщений PCEP. Для DeadTimer **следует** устанавливать значение 0 и оно **должно** игнорироваться при Keepalive = 0. **Рекомендуется** устанавливать для DeadTimer значение в 4 раза больше Keepalive.

Пример.

А передаёт партнёру В сообщение Open с Keepalive=10 и DeadTimer=40. Это означает, что А передаёт сообщения Keepalive (или иные сообщения PCEP) партнёру В каждые 10 секунд и В будет считать сессию PCEP с А бездействующей, если не получит от А сообщений в течение 40 секунд.

SID (PCEP session ID - 8 битов)

Целочисленный беззнаковый идентификатор сессии PCEP. Значение SID **должно** инкрементироваться для каждой создаваемой сессии PCEP, значение **следует** увеличивать на 1 и при достижении максимума возвращаться к 0. Это служит для протоколирования и поиска неполадок.

SID применяется для однозначного указания сессий с тем же партнёром. Реализация PCEP может использовать общую нумерацию SID для всех партнёров или поддерживать независимые номера для каждого. Первый вариант позволяет поддерживать до 256 одновременных сессий, второй - столько же сессий для каждого партнёра. В каждом направлении применяется один идентификатор SID.

В тело объекта OPEN могут включаться необязательные TLV для указания характеристик PCC или PCE. Спецификация таких TLV выходит за рамки документа.

В сообщении Open объект OPEN указывает предлагаемые характеристики сессии PCEP. При получении неприемлемых характеристик в фазе инициализации сессии принимающая сторона PCEP (PCE) **может** включить в сообщение PCErr объект OPEN с предложением приемлемых значений характеристик сессии.

7.4. Объект RP

Объект с параметрами запроса (RP - Request Parameters) **должен** включаться во все сообщения PCReq и PCRep и **может** присутствовать в сообщениях PCNtf и PCErr. Объект служит для задания характеристик запроса расчёта пути.

Флаг P **должен** устанавливаться для объектов RP в сообщениях PCReq и PCRep и **должен** сбрасываться в сообщениях PCNtf и PCErr. Если объект RP получен с некорректным флагом P (см. выше), приёмная сторона **должна** передать сообщение PCErr с Error-Type=10 и Error-value=1. Соответствующий запрос расчёта пути **должен** отменяться элементом PCE без дополнительного уведомления.

7.4.1. Определение объекта

RP Object-Class = 2.

RP Object-Type = 1.

Формат тела объекта RP показан на рисунке.

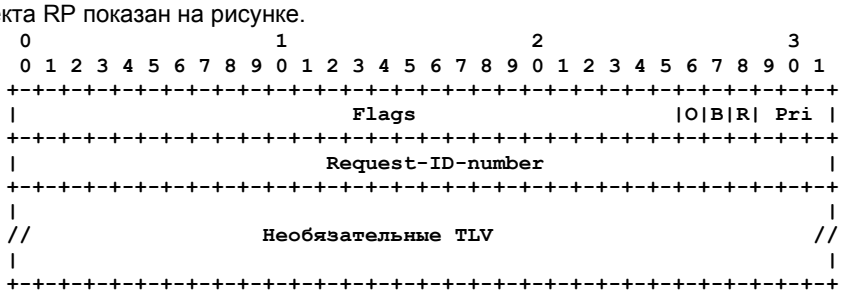

Рисунок 10. Формат тела объекта RP.

Тело объекта RP имеет переменный размер и может включать дополнительные TLV (в настоящее время не определены).

Flags (32 бита)

Ниже перечислены определённые в настоящее время флаги.

Pri (Priority - 3 бита)

Поле Priority может служить запрашивающему PCC для указания элементу PCE приоритета запроса (от 1 до 7). Решение о выборе приоритета принимается локально, значение 0 **должно** устанавливаться, если приоритет не задаётся. Трактовка приоритета для запроса расчета пути планировщиком PCE зависит от реализации и выходит за рамки этого документа. Отметим, что PCE не обязаны поддерживать поле приоритета, а для объектов RP клиентам PCC **рекомендуется** указывать приоритет 0. Если PCE не учитывает приоритет запросов, **рекомендуется** устанавливать приоритет 0 в объектах RP соответствующих сообщений PCRep, независимо от приоритета объекта RP в запросе PCReq. Большие значения соответствуют более высокому приоритету. Отметим, что согласованность значений приоритета для разных PCC должен обеспечивать администратор сети. Способность PCE поддерживать приоритизацию запросов **может** быть определена динамически клиентами PCC с помощью обнаружения возможностей PCE. Если это не анонсируется PCE, клиент PCC может установить желаемый приоритет и узнать о поддержке приоритизации запросов в PCE из поля Priority объекта RP в отклике PCRep. Если поле имеет значение 0, это указывает, что PCE не поддерживает приоритизацию запросов, т. е. указанные значения приоритета не учитываются.

R (Reoptimization - 1 бит)

Установленный флаг показывает, что запрашивающий PCC указывает, что запрос PCReq относится к повторной оптимизации имеющегося TE LSP. Для всех TE LSP кроме путей с нулевой пропускной способностью при установленном бите R объект RRO (параграф 7.10) **должен** включаться в сообщение PCReq для указания пути имеющегося TE LSP. Для этих TE LSP при установленном бите R имеющаяся пропускная способность TE LSP для повторной оптимизации **должна** быть представлена в объекте BANDWIDTH

(параграф 7.7). Этот объект BANDWIDTH является дополнением к экземпляру объекта, служащего для описания желаемой полос LSP после повторной оптимизации. Для LSP с нулевой пропускной способностью объекты RRO и BANDWIDTH, указывающие характеристики имеющегося TE LSP, являются необязательными.

B (Bi-directional - 1 бит)

Установленный флаг показывает, что запрос PCC для расчёта пути относится к двухстороннему TE LSP с одинаковыми требованиями по организации трафика для обоих направлений, включая «общую судьбу», защиту и восстановление, LSR, каналы TE и требования к ресурсам (например, к задержке и её вариациям). При сброшенном флаге TE LSP является односторонним.

O (strict/loose - 1 бит)

Установленный флаг в сообщении PCReq указывает приемлемость нестрогих путей. Сброшенный флаг указывает PCE, что путь должен включать лишь строгие интервалы. В сообщении PCRep установленный бит O показывает, что возвращённый путь является нестрогим, а сброшенный говорит о наличии в пути исключительно строгих интервалов.

Нераспределенные биты являются резервными. Они **должны** устанавливаться в 0 при передаче, а на приёмной стороне **должны** игнорироваться.

Request-ID-number (32 бита)

Значение Request-ID-number комбинируется с IP-адресом отправителя aPCC и адресом PCE для однозначного указания контекста запроса на расчёт пути. Request-ID-number позволяет однозначно различать ожидающие запросы, поэтому **должно** меняться (например, увеличиваться) при отправке каждого нового запроса PCE.

При достижении максимума отсчёт продолжается с минимума, значение 0x00000000 считается недействительным.

Если от PCE не получено отклика с расчётом пути (например, запрос был отброшен PCE по причине нехватки памяти), а PCC хочет повторить запрос, **должно** сохраняться прежнее значение Request-ID-number. При получении запроса на расчёт пути от PCC с таким же значением Request-ID-number элементу PCE **следует** считать этот запрос новым. Реализация **может** кэшировать отклики на запросы расчёта путей для ускоренной обработки повторов без двойного расчёта (в случае отбрасывания или потери первого запроса). При получении отклика с расчётом пути от PCE с тем же Request-ID-number клиенту PCC **следует** отбрасывать отклик без уведомления.

Для разных запросов, отправляемых PCE, **должны** указываться разные Request-ID-number.

Одинаковые значения Request-ID-number **могут** применяться для запросов, передаваемых разным PCE. Отклик с расчётом пути будет однозначно определяться IP-адресом отвечающего PCE.

7.4.2. Обработка объекта RP

При получении PCReq без объекта RP элемент PCE **должен** передать сообщение PCErr запрашивающему PCC с Error-Type = "Required Object missing" и Error-value = "RP Object missing".

Если бит O объекта RP в сообщении PCReq сброшен и локальная политика PCE не предоставляет явных путей (например, из соображений конфиденциальности), **должно** передаваться сообщение PCErr с Error-Type = "Policy Violation" и Error-value = "O bit cleared" запрашивающему клиенту PCC, а ожидающий запрос расчёта пути **должен** отбрасываться.

Когда установлен бит R объекта RP в сообщении PCReq, это указывает, что запрос расчёта пути связан с повторной оптимизацией имеющегося TE LSP. В этом случает PCC **должен** также предоставить строгий/нестрогий путь включением объекта RRO в сообщение PCReq, чтобы избежать/ограничить двойной учёт пропускной способности, если TE LSP имеет отличную от 0 полосу. Если PCC не запрашивает строгий путь (бит O установлен), повторная оптимизация по-прежнему может быть запрошена PCC, но это требует от PCE учёта состояний (отслеживания ранее рассчитанного пути со связанным списком строгих интервалов) или возможности находить полный сегмент нужного пути. В дополнение к этому PCC **должен** информировать PCE о работающем пути и связанном с ним списке строгих интервалов в PCReq. Отсутствие RRO в PCReq для TE LSP с отличной от 0 полосой (установлен бит R в объекте RP) **должно** вызывать отправку сообщения PCErr с Error-Type = "Required Object Missing" и Error-value = "RRO Object missing for reoptimization".

Если PCC/PCE принимает сообщение PCRep/PCReq с объектом RP, указывающим неизвестный Request-ID-number, PCC/PCE **должен** передать сообщение PCErr с Error-Type="Unknown request reference". Это служит для отладки. Если PCC/PCE получает сообщения PCRep/PCReq с неизвестными запросами со скоростью не меньше MAX-UNKNOWN-REQUESTS в минуту, PCC/PCE **должен** передать сообщение PCEP CLOSE со значением "Reception of an unacceptable number of unknown requests/replies". **Рекомендуемое** значение MAX-UNKNOWN-REQUESTS составляет 5. PCC/PCE **должен** закрыть соединение TCP и **недопустима** передача других сообщений PCEP в сессии PCEP.

Приём сообщения PCEP с объектом RP, имеющим Request-ID-number=0x00000000, **должен** трактоваться как неизвестный запрос.

7.5. Объект NO-PATH

Объект NO-PATH используется в сообщениях PCRep с откликом о неудаче запроса на расчёт пути (PCE не удалось найти путь, соответствующий набору ограничений). Когда PCE не может найти соответствующий заданному в запросе набору ограничений, он **должен** включить объект NO-PATH в сообщение PCRep.

Имеется несколько категорий проблем, которые могут вести к негативному отклику. Например, цепочка PCE может быть разорвана (если в расчёте участвует более одного PCE) или не найден соответствующий ограничениям путь. Поле NI (Nature of Issue - природа проблемы) в объекте NO-PATH служит для указания категории ошибки.

Если PCE поддерживает такую возможность, объект NO-PATH **может** включать NO-PATH-VECTOR TLV, определённый ниже, для предоставления дополнительной информации о причинах негативного отклика. Сообщение PCRep **может** также включать список объектов, которые указывают невыполненные ограничения. PCE **может** просто ответить набором объектов, вызвавших неудачу при расчёте, а **может** указать предлагаемые значения, для которых путь можно найти (т. е. значения, отличающиеся от указанных в запросе).

NO-PATH Object-Class = 3.

NO-PATH Object-Type = 1.

Формат тела объекта NO-PATH показан ниже.

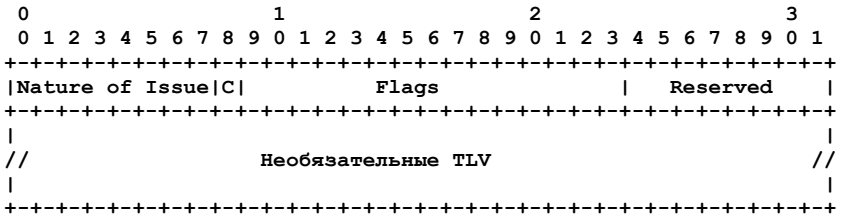

Рисунок 11. Формат объекта NO-PATH.

NI - Nature of Issue (8 битов)

Поле NI служит для указания проблемы, вызвавшей негативный отклик. Определены два значения:

0 - не найдено пути, соответствующего заданному набору ограничений;

1 - цепочка PCE разорвана.

Поле NI может применяться PCC с различными целями:

- корректировка ограничений перед отправкой нового запроса на расчёт пути;
- явный выбор новой цепочки PCE;
- регистрация типа ошибки для последующих действий администратора сети.

Агентство IANA управляет пространством кодов NI, как описано в разделе 9.

Flags (16 битов)

В настоящее время определён один флаг:

C (1 бит)

При установленном флаге PCE указывает набор невыполненных ограничений (причины, по которым путь не был найден) в сообщении PCRep путём включения соответствующих объектов PCEP. При сброшенном флаге невыполненные ограничения не указываются. Флаг C не имеет значения и игнорируется, если NI отличается от 0x00.

Нераспределенные биты являются резервными. Они **должны** устанавливаться в 0 при передаче, а на приёмной стороне **должны** игнорироваться.

Reserved (8 битов)

Резервное поле. **Должно** устанавливаться в 0 при передаче, а на приёмной стороне **должно** игнорироваться.

Тело объекта NO-PATH имеет переменный размер и может содержать дополнительные TLV. Единственным определенным в настоящее время TLV является NO-PATH-VECTOR TLV, описанный ниже.

Пример

Рассмотрим PCC, передающий элементу PCE запрос расчёта пути для TE LSP с пропускной способностью X Мбит/с. Предположим, что PCE не может найти пути, обеспечивающего X Мбит/с. В этом случае PCE должен включить в сообщение PCRep объект NO-PATH. Дополнительно PCE может включить исходный объект BANDWIDTH для указания причины отказа при расчёте пути (в этом случае поле NI имеет значение 0x00 и флаг C установлен). Если элемент PCE поддерживает такую возможность, он может дополнительно включить объект BANDWIDTH и указать значение Y в поле bandwidth объекта BANDWIDTH (в этом случае устанавливается флаг C), где Y определяет возможную пропускную способность для TE LSP с заданными характеристиками (такими как приоритет организации и удержания, атрибут TE LSP, локальная защита и т. п.), который может быть рассчитан. Когда объект NO-PATH отсутствует в сообщении PCRep, это говорит о полном выполнении запроса расчёта пути и соответствующие пути представляются в сообщении PCRep.

В объект NO-PATH **можно** включать необязательный NO-PATH-VECTOR TLV для предоставления дополнительной информации о причинах негативного отклика.

NO-PATH-VECTOR TLV соответствует формату PCEP TLV, определённому в параграфе 7.1, и содержит двухбайтовые поля типа и размера (значения), за которыми следует 32-битовое поле флагов.

```
Тип - 1
Размер - 4 байта
Значение - 32-битовое поле флагов
```

Агентство IANA поддерживает пространство флагов для NO-PATH-VECTOR TLV (см. параграф 9).

В настоящее время определены 3 флага:

- бит 31 - PCE в данный момент не доступен;
- бит 30 - неизвестный адресат;
- бит 29 - неизвестный источник.

7.6. Объект END-POINTS

Объект END-POINTS применяется в сообщении PCReq для указания IP-адресов отправителя и получателя в пути, для которого запрашивается расчёт. Флаг P в объекте END-POINTS **должен** быть установлен. Если объект END-POINTS получен со сброшенным флагом P, принимающая сторона **должна** отправить сообщение PCErr с Error-Type=10 и Error-value=1. Соответствующий запрос расчёта пути **должен** быть отвергнут PCE без дополнительного уведомления.

Отметим, что адреса отправителя и получателя в объекте END-POINTS могут соответствовать IP-адресам отправителя и получателя в TE LSP или сегменте пути. Определены два объекта END-POINTS (для IPv4 и IPv6).

END-POINTS Object-Class = 4.

END-POINTS Object-Type = 1 для IPv4 и 2 для IPv6.

Формат тела объекта END-POINTS для IPv4 (Object-Type=1) показан на рисунке 12.

Формат тела объекта END-POINTS для IPv6 (Object-Type=2) показан на рисунке 13.

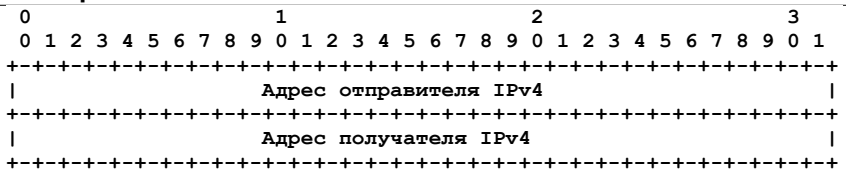

Рисунок 12. Формат тела объекта END-POINTS для IPv4.

Тело объекта END-POINTS имеет фиксированный размер 8 байтов для IPv4 и 32 байта для IPv6.

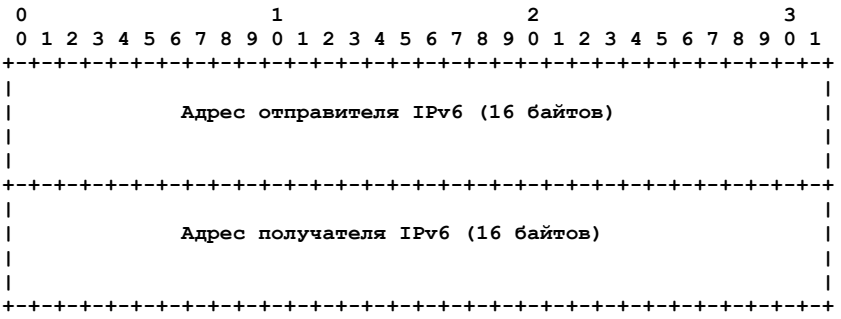

Рисунок 13. Формат тела объекта END-POINTS для IPv6.

При наличии нескольких объектов END-POINTS первый **должен** обрабатываться, а остальные игнорироваться.

7.7. Объект BANDWIDTH

Объект BANDWIDTH служит для указания запрашиваемой пропускной способности TE LSP. Обозначение пропускной способности аналогично применяемому в сигнализации RSVP [RFC2205], [RFC3209], [RFC3473].

Если запрашиваемая пропускная способность равна 0, объект BANDWIDTH можно не включать, но при запросе отличной от 0 пропускной способности сообщение PCReq **должно** содержать объект BANDWIDTH.

В случае повторной оптимизации TE LSP пропускная способность имеющегося TE LSP также **должна** указываться, если запрашиваемое значение отличается от имеющейся пропускной способности. Поэтому определены два значения Object-Type, относящиеся к запросу пропускной способности и повторной оптимизации TE LSP.

Объект BANDWIDTH может передаваться в сообщениях PCReq и PCRep.

BANDWIDTH Object-Class = 5.

Определены два значения Object-Type для объекта BANDWIDTH:

- запрос пропускной способности - BANDWIDTH Object-Type = 1.
- пропускная способность для имеющегося TE LSP при повторной оптимизации - BANDWIDTH Object-Type = 2.

Формат тела объекта BANDWIDTH показан ниже.

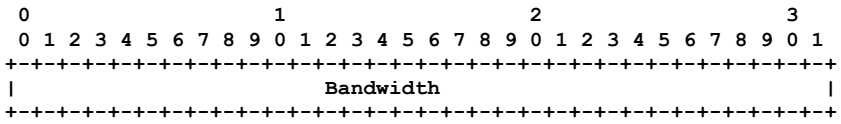

Рисунок 14. Формат тела объекта BANDWIDTH.

Bandwidth (32 бита)

Запрашиваемая пропускная способность, указанная 32-битовым значением с плавающей точкой в формате IEEE [IEEE.754.1985] в бит/с. Таблица часто применяемых значений приведена в параграфе 3.1.2 [RFC3471].

Тело объекта BANDWIDTH имеет фиксированный размер 4 байта.

7.8. Объект METRIC

Необязательный объект METRIC может применяться с разными целями.

В сообщение PCReq клиент PCC **может** включить один или несколько объектов METRIC.

- Для указания метрики, которая **должна** быть оптимизирована алгоритмом расчёта пути (метрика IGP или TE, число интервалов). В настоящее время определены три типа метрики - стоимость IGP, метрика TE (см. [RFC3785]) и число интервалов пересылки в TE LSP.
- Для указания границ стоимости пути, которые **недопустимо** переходить для пути, считающегося приемлемым клиентом PCC.

В сообщение PCRep объект METRIC **можно** включать для указания стоимости рассчитанного пути. Он **может** также помещаться в PCRep с объектом NO-PATH для указания ограничения метрики, которые не были выполнены.

Аспекты алгоритма расчёта пути, используемые PCE для оптимизации конкретной метрики, выходят за рамки документа.

Следует понимать, что такая метрика имеет смысл лишь при согласованном применении - например, при передаче значения задержки между двумя PCE в разных доменах должны использоваться согласованные методы определения задержки.

Отсутствие объекта METRIC **должно** восприниматься элементом PCE как запрос расчёта пути без ограничения метрики.

METRIC Object-Class = 6.

METRIC Object-Type = 1.

Формат тела объекта METRIC показан ниже.

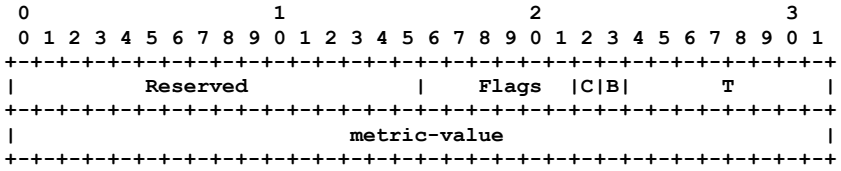

Рисунок 15. Формат тела объекта METRIC.

Тело объекта METRIC имеет фиксированный размер 8 байтов.

Reserved (16 битов)

Резервное поле. **Должно** устанавливаться в 0 при передаче, а на приёмной стороне **должно** игнорироваться.

T (Type - 8 битов)

Задаёт тип метрики. В настоящее время определены 3 типа:

T=1 - метрика IGP;

T=2 - метрика TE;

T=3 - число интервалов пересылки (Hop Count).

Flags (8 битов)

В настоящее время определены 2 флага.

B (Bound - 1 бит) - при установке флага в сообщении PCReq поле metric-value указывает границу (максимум) для метрики пути, которая не должна превышаться, чтобы клиент PCC счёл путь приемлемым. Метрика пути должна быть не больше значения, указанного полем metric-value. При сброшенном флаге B значение metric-value не используется в качестве ограничения метрики.

C (Computed Metric - 1 бит) - установка флага в сообщении PCReq указывает, что элемент PCE **должен** обеспечить указанное значение метрики рассчитанного пути (если найден соответствующий ограничениям путь)в сообщении PCRep.

Нераспределенные биты являются резервными. Они **должны** устанавливаться в 0 при передаче, а на приёмной стороне **должны** игнорироваться.

Metric-value (32 бита)

Значение метрики представляется 32-битовым числом с плавающей точкой в формате IEEE (см. [IEEE.754.1985]).

В сообщении PCRep или PCReq для данного запроса (т. е. для данного RP) **может** присутствовать множество объектов METRIC. Для данного запроса **должно** присутствовать не более одного экземпляра объекта METRIC каждого типа с одинаковым флагом B. Если присутствует несколько экземпляров объекта METRIC с одинаковым флагом B, **должен** рассматриваться лишь первый из них, а прочие **должны** игнорироваться.

Для данного запроса разрешается наличие двух объектов METRIC с разными флагами B. Кроме того, разрешается помещать для данного запроса два объекта METRIC разных типов со сброшенными флагами B - в этом случае элемент PCE должен применять соответствующую функцию для оптимизации по множеству параметров.

Объект METRIC, служащий для указания метрики, оптимизируемой при расчёте пути, **должен** иметь сброшенный флаг B и подходящее значение флага C. Когда расчёт пути связан с повторной оптимизацией имеющегося TE LSP (в этом случае установлен флаг R объекта RP), реализация **может** установить в поле metric-value рассчитанное значение метрики для повторной оптимизации TE LSP по конкретному типу метрики.

Объект METRIC, служащий для ограничения, **должен** иметь установленный флаг B и подходящие значения флага C и поля metric-value.

В сообщении PCRep, если это разрешено политикой PCE, **должно** присутствовать не более 1 элемента METRIC, который указывает метрику рассчитанного пути, если флаг C был установлен в соответствующем объекте METRIC запроса на расчёт пути (флаг B **должен** быть сброшен). Флаг C не имеет значения в сообщениях PCRep. Сообщение PCRep **может** включать дополнительные объекты METRIC, соответствующие границам, - в этом случае поле metric-value **должно** совпадать с соответствующей метрикой рассчитанного пути (флаг B **должен** быть установлен). Если не найдено пути, соответствующего ограничениям, объект METRIC **может** также включаться в сообщение PCRep с объектом NO-PATH для указания значения метрики, которое может быть обеспечено.

Пример

Если PCC передаёт элементу PCE запрос расчёта пути, где оптимизируется метрика IGP, а метрика TE не должна превышать M, в сообщение PCReq помещаются 2 объекта METRIC:

- первый объект METRIC с B=0, T=1, C=1, metric-value=0x0000;
- второй объект METRIC с B=1, T=2, metric-value=M

Если путь, соответствующий заданным ограничениям, найден и политика не запрещает возвращать рассчитанную метрику, PCE помещает в отклик один объект METRIC с B=0, T=1 и metric-value с рассчитанной стоимостью пути IGP. В дополнение PCE может включить объект METRIC с B=1, T=2 и metric-value с рассчитанной стоимостью пути TE.

7.9. Объект ERO

Объект ERO служит для кодирования пути TE LSP через сеть. ERO передаётся в сообщении PCRep для представления TE LSP, если расчёт пути был успешным.

Содержимое объекта идентично содержимому объекта RSVP-TE[1] ERO[2], определённого в [RFC3209], [RFC3473], [RFC3477]. Т. е. объект представляется последовательностью субобъектов и каждый из субобъектов RSVP-TE ERO,

[1]Resource Reservation Protocol Traffic Engineering Extensions - расширения организации трафика RSVP.
[2]Explicit Route Object - объект с явным маршрутом.

который уже определён или может быть определён в будущем для использования в RSVP-TE ERO, подходит для этого объекта.

Типы субобъектов PCEP ERO соответствуют типам RSVP-TE ERO.

Поскольку явный путь доступен для сигнализации с помощью уровня управления MPLS или GMPLS, смысл всех субобъектов и полей в данном объекте идентичен определённому для ERO.

ERO Object-Class = 7.

ERO Object-Type = 1.

7.10. Объект RRO

Объекты RRO передаются исключительно в сообщениях PCReq для информирования о маршруте TE LSP, для которого желательна повторная оптимизация.

Содержимое объекта идентично представлению объекта Route Record Object, определённого в [RFC3209], [RFC3473], [RFC3477]. Т. е. объект представляется последовательностью субобъектов и каждый из субобъектов RSVP-TE RRO, который уже определён или может быть определён в будущем для использования в RSVP-TE RRO, подходит для этого объекта.

Смысл всех субобъектов и полей данного объекта идентичен определённому для RSVP-TE RRO.

Типы субобъектов PCEP RRO соответствуют типам RSVP-TE RRO.

RRO Object-Class = 8.

RRO Object-Type = 1.

7.11. Объект LSPA

Необязательный объект LSPA (атрибуты LSP) задаёт различные атрибуты TE LSP, принимаемые во внимание PCE при расчёте пути. Объект LSPA может включаться в сообщение PCReq или PCRep при неудаче расчёта пути (в этом случае PCRep содержит также объект NO-PATH, а LSPA указывает ограничения, которые не выполнены). Большинство полей LSPA идентичны полям атрибута SESSION-ATTRIBUTE (C-Type = 7), определённого в [RFC3209] и [RFC4090]. Отсутствие объекта в сообщении PCReq говорит о том, что приоритет Setup и Holding имеет значение 0 и нет ограничений на сходство (affinity). Подробное описание сходства ресурсов приведено в параграфе 4.7.4 [RFC3209] .

LSPA Object-Class = 9.

LSPA Object-Types = 1.

Формат тела объекта LSPA показан ниже.

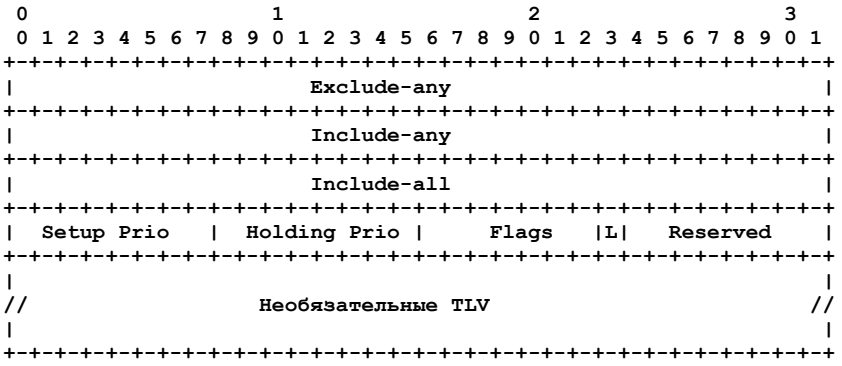

Рисунок 16. Формат тела объекта LSPA.

Setup Prio (Setup Priority - 8 битов)

Приоритет получения ресурсов для TE LSP от 0 до 7 (0 указывает высший приоритет). Setup Priority служит для определения возможности вытеснения одной сессии другой.

Holding Prio (Holding Priority - 8 битов)

Приоритет удержания ресурсов для TE LSP от 0 до 7 (0 указывает высший приоритет). Holding Priority служит для определения возможности вытеснения одной сессии другой.

Flags (8 битов)

Флаг L

Соответствует биту Local Protection Desired [RFC3209] объекта SESSION-ATTRIBUTE. Установка флага означает, что рассчитанный путь должен включать каналы, защищённые Fast Reroute, как определено в [RFC4090].

Не распределенные биты являются резервными. Они **должны** устанавливаться в 0 при передаче, а на приёмной стороне **должны** игнорироваться.

Reserved (8 битов)

Резервное поле. **Должно** устанавливаться в 0 при передаче, а на приёмной стороне **должно** игнорироваться.

Отметим, что в будущем могут быть определены необязательные TLV для передачи дополнительных атрибутов TE LSP, таких как определённые в [RFC5420].

7.12. Объект IRO

Необязательный объект IRO[1] может служить для указания того, что рассчитанный маршрут **должен** проходить через заданный набор сетевых элементов. IRO **может** включаться в сообщения PCReq и PCRep. При передаче в сообщении PCRep вместе с NO-PATH объект IRO указывает набор элементов, которые не позволили PCE найти путь.

[1]Include Route Object - объект со включенным маршрутом.

IRO Object-Class = 10.

IRO Object-Type = 1.

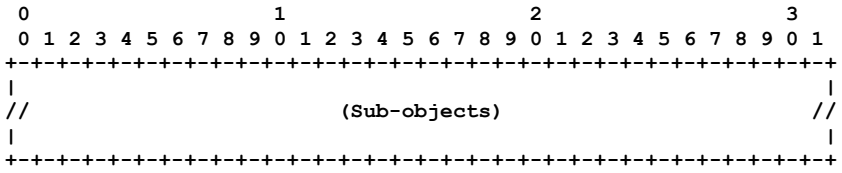

Рисунок 17. Формат тела объекта IRO.

Sub-objects

IRO образуется из субобъектов, идентичных определенным в [RFC3209], [RFC3473], [RFC3477].
Ниже перечислены поддерживаемые типы субобъектов.

Тип	***Субобъект***
1	Префикс IPv4.
2	Префикс IPv6.
4	Идентификатор безадресного интерфейса.
32	Номер автономной системы.

Бит L таких субобъектов не имеет значения в IRO.

7.13. Объект SVEC

7.13.1. Понятие зависимых и синхронизированных запросов расчёта пути

Запросы расчёта пути называются независимыми, если они не связаны между собой. И наоборот, запросы, которые не могут быть выполнены независимо один от другого (например, запрос на расчёт набора разных путей), называются зависимыми.

Набор запросов называют несинхронизированным, если PCE может выполнить их последовательно и независимо.

При некоторых обстоятельствах синхронизация запросов может давать преимущества или быть обязательной.

Рассмотрим набор из N путей TE LSP, для которого PCC нужно передать запросы расчёта пути элементу PCE. Первым решением является отправка N сообщений PCReq выбранному PCE. В этом случае запросы расчёта пути не синхронизированы. Отметим, что PCC может передать N запросов в K элементов PCE для распределения нагрузки. Учитывая, что M запросов (M<N) передаётся определённому PCEi, как описано выше, эти M запросов могут быть переданы в форме последовательных сообщений PCReq, направленных PCEi, или собраны в одно сообщение PCReq (PCEP позволяет объединить несколько запросов расчёта пути в одном сообщении PCReq). Тем не менее, даже при независимых запросах может оказаться желательным запросить у PCE синхронизированный расчёт, что может вести к более оптимальным вычислениям и/или снижать вероятность блокировки в PCE без учёта состояния. Иными словами, PCE не нужно вычислять соответствующие пути последовательно и независимо, а следует рассчитывать их «одновременно». Например, попытка «одновременно» рассчитать M путей TE LSP может позволить PCE повысить вероятность выполнения множества ограничений.

Рассмотрим 2 TE LSP, для которых запрашивается пропускная способность N1 и N2 Мбит/с с максимальной сквозной задержкой для каждого TE LSP в X мсек. Могут возникать ситуации, в которых расчёт первого TE LSP независимо от второго TE LSP может приводить к невозможности выполнения ограничений задержки для второго TE LSP.

Второй пример связан с ограничением пропускной способности. Довольно просто представить ситуацию, когда последовательный, независимый расчёт путей приведёт к невозможности выполнить оба запроса (в результате фрагментирования пропускной способности), тогда как синхронизированный расчёт позволит выполнить оба запроса.

Последний пример относится к способности избегать выделения одного и того же ресурса для множества запросов, что помогает снизить вероятность отказа при организации пути по сравнению с последовательным расчётом для отдельных запросов.

Зависимые расчёты пути обычно синхронизированы. Например, при запросе M разных путей несинхронизированные расчёты существенно повышают вероятность невыполнения всех запросов (иногда это называют «проблемой захвата»).

Кроме того, это не позволяет PCE реализовать специализированные функции, такие как минимизация суммарной стоимости TE LSP. В таких случаях запросы расчёта путей должны синхронизироваться, они не могут выполняться независимо один от другого.

Набор независимых запросов расчёта пути может быть синхронизированным или несинхронизированным.

Синхронизация набора запросов расчёта путей достигается с помощью объекта SVEC, который задаёт список синхронизированных запросов 9зависимых или независимых).

PCEP поддерживает три режима:

- пакет из набора независимых и несинхронизированных запросов расчёта пути;
- пакет из набора независимых и синхронизированных запросов расчёта пути (требуется объект SVEC);
- пакет из набора зависимых и синхронизированных запросов расчёта пути (требуется объект SVEC).

7.13.2. Объект SVEC

В параграфе 7.13.1 рассмотрены обстоятельства, при которых может быть желательна или требоваться синхронизация набора запросов на расчёт путей. Объект SVEC (Synchronization VECtor - вектор синхронизации) позволяет PCC запросить синхронизацию набора зависимых или независимых запросов. Необязательный объект SVEC может включаться в сообщения PCReq.

Целью объекта SVEC в сообщении PCReq является запрос синхронизации M запросов расчёта путей. Объект SVEC имеет переменный размер и включает M запросов расчёта пути, которые нужно синхронизировать. Каждый запрос однозначно указывается полем Request-ID-number в соответствующем объекте RP. Объект SVEC также включает набор флагов, задающих тип синхронизации.

SVEC Object-Class = 11.

SVEC Object-Type = 1.

Формат тела объекта SVEC показан на рисунке.

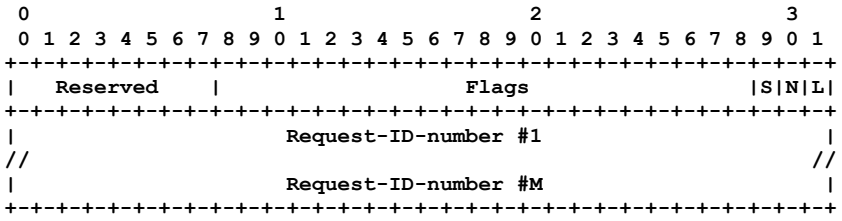

Рисунок 18. Формат тела объекта SVEC.

Reserved (8 битов)

Резервное поле. **Должно** устанавливаться в 0 при передаче, а на приёмной стороне **должно** игнорироваться.

Flags (24 бита)

Определяют потенциальные зависимости между запросами расчёта путей.

L (Link diverse)

Установленный флаг указывает, что рассчитываемым путям, соответствующим запросам, заданным последующими объектами RP, **недопустимо** иметь общие каналы.

N (Node diverse)

Установленный флаг указывает, что рассчитываемым путям, соответствующим запросам, заданным последующими объектами RP, **недопустимо** иметь общие узлы.

S (SRLG diverse)

Установленный флаг указывает, что рассчитываемым путям, соответствующим запросам, заданным последующими объектами RP, **недопустимо** иметь общие SRLG[1].

В случае набора из M синхронизированных независимых запросов биты L, N и S сбрасываются.

Не распределенные биты **должны** устанавливаться в 0 при передаче, а на приёмной стороне **должны** игнорироваться.

Определённые выше флаги не исключают один другого.

7.13.3. Обработка объекта SVEC

Объект SVEC позволяет PCC задать список M запросов на расчёт путей, которые **должны** быть синхронизированы с потенциальной зависимостью. Набор из M запросов расчёта путей может быть передан в одном или множестве сообщений PCReq. В последнем случае **рекомендуется** реализовать в PCE локальный таймер (SyncTimer), активируемый при получении первого сообщения PCReq с объектом SVEC и по завершении отсчёта таймера фиксировать протокольную ошибку, если не были получены все M запросов. Если PCE получает запрос расчёта пути, который не может быть выполнен (например, в результате наличия в PCReq неподдерживаемого объекта с установленным битом P), PCE передаёт сообщение PCErr для этого запроса (см. параграф 7.2) и **должен** отменить весь набор связанных запросов, а также **должен** передать сообщение PCErr с Error-Type="Synchronized path computation request missing".

Отметим, что такие сообщения PCReq могут также включать несинхронизированные запросы расчёта пути. Например, сообщение PCReq может включать N синхронизированных запросов, относящихся к RP 1, ..., RP N и указанных в объекте SVEC, а также другие запросы расчёта путей, которые обрабатываются как обычно.

7.14. Объект NOTIFICATION

Объекты NOTIFICATION передаются только в сообщениях PCNtf от PCC к PCE или от PCE к PCC для информирования о событии.

NOTIFICATION Object-Class = 12.

NOTIFICATION Object-Type = 1.

Формат тела объекта NOTIFICATION показан на рисунке.

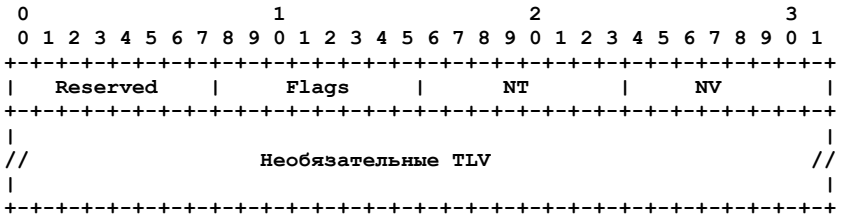

Рисунок 19. Формат тела объекта NOTIFICATION.

Reserved (8 битов)

Резервное поле. **Должно** устанавливаться в 0 при передаче, а на приёмной стороне **должно** игнорироваться.

Flags (8 битов)

Флаги в настоящее время не определены. Не распределенные биты **должны** устанавливаться в 0 при передаче, а на приёмной стороне **должны** игнорироваться.

[1]Shared Risk Link Group - группа каналов с общими рисками.

NT (Notification Type - 8 битов)

Notification-type указывает тип уведомления.

NV (Notification Value - 8 битов)

Notification-value обеспечивает дополнительную информацию, связанную с конкретным типом уведомления.

Значения Notification-type и Notification-value поддерживаются IANA.

Ниже перечислены определённые в настоящее время Notification-type и Notification-value.

- Notification-type=1 - ожидающий запрос отменен
  - Notification-value=1 - клиент PCC отменил набор ожидающих запросов. Notification-type=1 с Notification-value=1 указывает, что PCC хочет информировать PCE об отмене набора ожидающих запросов. Такое событие может возникать в результате внешних условий, таких как получение отклика от другого PCE (если PCC передал нескольким PCE одни и те же запросы расчёта путей), отказ сети, делающий запрос устаревшим, или иное локальное событие в PCC. Объект NOTIFICATION с Notification-type=1 и Notification-value=1 передаётся в сообщении PCNtf, передаваемом от PCC к PCE. Объект RP, соответствующий отменённому запросу, также **должен** включаться в сообщение PCNtf. Множество объектов RP может включаться в одно сообщение PCNtf и в этом случае уведомления применяется ко всем этим объектам. Если такое уведомление получено клиентом PCC от PCE, PCC **должен** отбросить уведомление без генерации ошибки.
  - Notification-value=2 - элемент PCE отменил набор ожидающих запросов. Notification-type=1 с Notification-value=2 указывает, что PCE хочет информировать PCC об отмене набора ожидающих запросов. Объект NOTIFICATION с Notification-type=1 и Notification-value=2 передаётся в сообщении PCNtf от PCE к PCC. Объект RP, соответствующий отменённому запросу, также **должен** включаться в сообщение PCNtf. Множество объектов RP может включаться в одно сообщение PCNtf и в этом случае уведомления применяется ко всем этим объектам. Если такое уведомление получено элементом PCE от PCC, PCE **должен** отбросить уведомление без генерации ошибки.
- Notification-type=2 - перегрузка PCE
  - Notification-value=1 - Notification-type=2 с Notification-value=1 указывает клиенту PCC, что PCE в настоящее время перегружен. Если объекты RP не включены в сообщение PCNtf, **не следует** передавать PCE другие запросы, пока состояние перегрузки не закончится - остающиеся запросы будут обработаны. Если некоторые из ожидающих запросов не могут быть обработаны в результате перегрузки, PCE **должен** также включить в уведомление объекты RP, указывающие запросы, отменённые PCE. В таких случаях PCE не передаёт дополнительного сообщения PCNtf с Notification-type=1 и Notification-value=2, поскольку список отменённых запросов указывается соответствующими объектами RP. Если такое уведомление получено элементом PCE от PCC, PCE **должен** отбросить уведомление без генерации ошибки.
  - Реализации PCE **следует** использовать механизм с двумя порогами для определения состояния перегрузки применительно к конкретному ресурсу (например, CPU, память). Применение двух порогов позволит обеспечить гистерезис смены состояний «перегружен - не перегружен».
  - Дополнительно в объект NOTIFICATION может включаться OVERLOADED-DURATION TLV с указанием периода, в течение которого не следует передавать дополнительных запросов элементу PCE. По истечении этого периода PCE не считается перегруженным.

    OVERLOADED-DURATION TLV соответствует формату PCEP TLV, определённому в параграфе 7.1 и включает 2-байтовые поля типа и размера, за которыми следует 32-битовое поле значения.

    Type - 2

    Length - 4 байта

    Value - 32-битовое поле, указывающее оценку периода перегрузки PCE в секундах.
  - Notification-value=2 - Notification-type=2 с Notification-value=2 указывает, что PCE больше не перегружен и доступен для обработки новых запросов расчёта пути. Реализации **следует** обеспечивать передачу элементом PCE такого уведомления каждому клиенту PCC, которому было отправлено сообщение Notification (с Notification-type=2, Notification-value=1), за исключением случаев, когда эти сообщения включали OVERLOADED-DURATION TLV и PCE хочет дождаться завершения указанного периода перед получением новых запросов. Если такое уведомление получено элементом PCE от PCC, PCE **должен** отбросить уведомление без генерации ошибки. **Рекомендуется** поддерживать в PCE ту или иную процедуру демпфирования уведомлений для предотвращения слишком частой смены состояний «перегрузка - отсутствие перегрузки». Например, реализация может использовать гистерезис с двухпороговым механизмом, вызывающим отправку сообщений о состоянии перегрузки. Кроме того, при значительной нестабильности ресурсов PCE **следует** применять дополнительные механизмы демпфирования (линейные или экспоненциальные) для снижения частоты уведомлений и предотвращения ненужных осцилляций.

Когда PCC получает индикацию перегрузки от PCE, ему следует учитывать влияние на всю сеть. Клиент должен помнить о том, что другие PCC также могли получить уведомление и в результате этого многие запросы расчёта путей могут быть перенаправлены другим PCE. Это может привести к дополнительной перегрузке элементов PCE в сети. Поэтому приложению на PCC, получившем уведомление о перегрузке, следует применить ту или иную (например, экспоненциальную) форму снижения скорости отправки запросов на расчёт пути. Это особенно актуально при росте числа PCE, сообщающих о перегрузке.

7.15. Объект PCEP-ERROR

Объекты PCEP-ERROR передаются лишь в сообщениях PCErr для уведомления об ошибках PCEP.

PCEP-ERROR Object-Class = 13.

PCEP-ERROR Object-Type = 1.

Формат тела объекта PCEP-ERROR показан ниже.

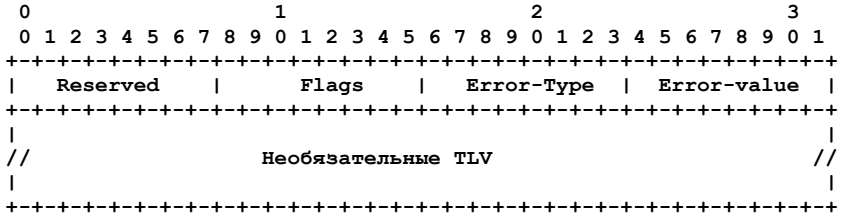

Рисунок 20. Формат тела объекта PCEP-ERROR.

Объект PCEP-ERROR используется для уведомления об ошибке PCEP и характеризуется полем Error-Type, указывающим тип ошибки, и полем Error-value с дополнительной информацией об ошибке конкретного типа. Значения Error-Type и Error-value поддерживаются IANA (см. раздел 9. Взаимодействие с IANA).

Reserved (8 битов)
Резервное поле. **Должно** устанавливаться в 0 при передаче, а на приёмной стороне **должно** игнорироваться.

Flags (8 битов)
Флаги в настоящее время не определены. Не распределенные биты **должны** устанавливаться в 0 при передаче, а на приёмной стороне **должны** игнорироваться.

Error-Type (8 битов)
Указывает класс ошибки.

Error-value (8 битов)
Обеспечивает дополнительную информацию об ошибке.

Объект PCEP-ERROR может включать дополнительные TLV с более подробной информацией об ошибке.

Сообщение PCErr может включать множество объектов PCEP-ERROR.

Для каждой ошибки PCEP определены значения Error-Type и Error-value, показанные ниже.

Error-Type	*Error-value*	*Значение*
1		Отказ при организации сессии PCEP.
	1	Получение недействительного сообщения Open или отсутствие Open.
	2	Сообщение Open не получено до завершения отсчёта таймера OpenWait.
	3	Неприемлемые или несогласуемые характеристики сессии.
	4	Неприемлемые или несогласуемые характеристики сессии.
	5	Получение второго сообщения Open с остающимися неприемлемыми характеристиками.
	6	Получение сообщения PCErr, предлагающего неприемлемые характеристики сессии.
	7	Не получено сообщения Keepalive или PCErr к моменту завершения отсчёта KeepWait.
2		Возможность не поддерживается.
3		Неизвестный объект.
	1	Нераспознанный класс объекта.
	2	Нераспознанный тип объекта.
4		Неподдерживаемый объект.
	1	Неподдерживаемый класс объекта.
	2	Неподдерживаемый тип объекта.
5		Нарушение политики.
	1	Установлен бит C в объекте METRIC (запрос отвергнут).
	2	Установлен бит O в объекте RP (запрос отвергнут).
6		Отсутствует обязательный объект.
	1	Отсутствует объект RP.
	2	Отсутствует объект RRO для запроса повторной оптимизации (бит R в объекте RP установлен) при отличной от нуля пропускной способности.
	3	Отсутствует объект END-POINTS.
7		Отсутствует запрос расчёта синхронизированного пути.
8		Ссылка на неизвестный запрос.
9		Попытка организовать вторую сессию PCEP.
10		Получение недействительного объекта.
	1	Получение объекта со сброшенным флагом P, когда спецификация требует установки флага.

Ниже приведены более подробные описания ошибок.

Error-Type=1
Отказ при организации сессии PCEP.
При получении некорректно сформированного сообщения принимающий узел PCEP **должен** передать сообщение PCErr с Error-Type=1 и Error-value=1.
Если не было принято сообщения Open до завершения отсчёта таймера OpenWait, принимающий узел PCEP **должен** передать сообщение PCErr с Error-Type=1 и Error-value=2 (см. Приложение A).
Если одна или несколько характеристик сессии PCEP не приемлемы для получившего узла и не согласуемы, принимающий узел **должен** передать сообщение PCErr с Error-Type=1 и Error-value=3.
Если получено сообщение Open с неприемлемыми, но согласуемыми характеристиками сессии, принимающий узел PCEP **должен** передать сообщение PCErr с Error-Type-1 и Error-value=4 (см. параграф 6.2).
Если получено второе сообщение Open в сессии PCEP и характеристики сессии остаются неприемлемыми, принимающий узел PCEP **должен** передать сообщение PCErr с Error-Type-1 и Error-value=5 (см. параграф 6.2).
Если получено сообщение PCErr в фазе организации сессии PCEP, которая включает сообщение Open, предлагающее неприемлемые характеристики сессии, принимающий узел PCEP **должен** передать сообщение PCErr с Error-Type=1 и Error-value=6.
Если не было принято сообщения Keepalive или PCErr до завершения отсчёта таймера KeepWait в фазе организации сессии PCEP, принимающий узел PCEP **должен** передать сообщение PCErr с Error-Type=1 и Error-value=7.

Error-Type=2

PCE указывает, что запрос расчёта пути не может быть выполнен, поскольку не поддерживается одна или несколько требуемых возможностей. Соответствующий запрос расчёта пути **должен** быть отменен.

Error-Type=3 или Error-Type=4

Если получено сообщение PCEP с объектом PCEP (с флагом P), не распознанным или не поддерживаемым PCE, элемент PCE **должен** передать PCErr с объектом PCEP-ERROR (Error-Type=3 или 4, соответственно). Дополнительно PCE **может** включить в сообщение PCErr неизвестный или неподдерживаемый объект. Запрос расчёта пути **должен** быть отменен PCE без дополнительного уведомления.

Error-Type=5

Если получен запрос расчёта пути, не соответствующий политике, согласованной между PCC и PCE, элемент PCE **должен** передать сообщение PCErr с объектом PCEP-ERROR (Error-Type=5). Соответствующий запрос расчёта пути **должен** быть отменен. Зависящие от политики TLV в объекте PCEP-ERROR могут быть определены в других документах для указания природы нарушения политики.

Error-Type=6

Если полученный запрос расчёта пути не включает обязательного объекта, элемент PCE **должен** передать сообщение PCErr с объектом PCEP-ERROR (Error-Type=6). При отсутствии нескольких обязательных объектов сообщение PCErr **должно** включать объект PCEP-ERROR для каждого пропущенного объекта. Соответствующий запрос расчёта пути **должен** быть отменен.

Error-Type=7

Если PCC передаёт синхронизированный запрос расчёта путей элементу PCE, а PCE не получает всех запросов, указанных в соответствующем объекте SVEC по завершении отсчёта SyncTimer (параграф 7.13.3), PCE **должен** передать сообщение PCErr с объектом PCEP-ERROR (Error-Type=7). Соответствующий синхронизированный расчёт путей **должен** быть отменен. PCE **рекомендуется** включать в сообщение REQ-MISSING TLV (см. ниже), указывающие пропущенные запросы.

REQ-MISSING TLV соответствует формату PCEP TLV, определённому в параграфе 7.1 и включает 2-байтовые поля типа и размера, за которыми следует 4-байтовое поле значения.

Type - 3

Length - 4 байта

Value - 4 байта, указывающих Request-ID-number, соответствующий пропущенному запросу.

Error-Type=8

Если PCC получает сообщение PCRep, относящееся к неизвестному запросу расчёта пути, PCC **должен** передать сообщение PCErr с объектом PCEP-ERROR (Error-Type=8). Дополнительно PCC **должен** включить в сообщение PCErr неизвестный объект RP.

Error-Type=9

Если узел PCEP обнаруживает попытку партнёра PCEP организовать вторую сессию PCEP, он **должен** передать сообщение PCErr с Error-Type=9 и Error-value=1. Существующая сессия PCEP **должна** сохраняться, а все последующие сообщения, относящиеся к попытке организации второй сессии PCEP, **должны** игнорироваться.

Error-Type=10

Если узел PCEP получает объект без флага P, но этот флаг должен быть установлен в соответствии с данной спецификацией, он **должен** передать сообщение PCErr с Error-Type=10 и Error-value=1.

7.16. Объект LOAD-BALANCING

Возникают ситуации, когда PCE не может найти TE LSP с пропускной способностью X, хотя такие требования могут быть выполнены путём организации набора TE LSP с суммарной пропускной способностью X. Таким образом, для PCC может быть полезным запрос набора TE LSP с суммарной пропускной способностью X Мбит/с, возможно с некоторым ограничением числа TE LSP и минимальной пропускной способности каждого TE LSP. Такие запросы выполняются с помощью вставки объекта LOAD-BALANCING в сообщение PCReq, передаваемое PCE.

Объект LOAD-BALANCING является необязательным.

LOAD-BALANCING Object-Class = 14.

LOAD-BALANCING Object-Type = 1.

Формат тела объекта LOAD-BALANCING показан на рисунке.

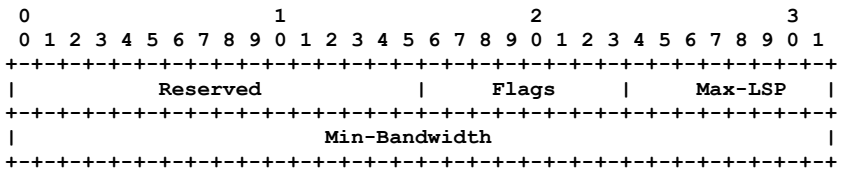

Рисунок 21. Формат тела объекта LOAD-BALANCING.

Reserved (16 битов)

Резервное поле. **Должно** устанавливаться в 0 при передаче, а на приёмной стороне **должно** игнорироваться.

Flags (8 битов)

Флаги в настоящее время не определены. Поле Flags **должно** устанавливаться в 0 при передаче, а на приёмной стороне **должно** игнорироваться.

Max-LSP (8 битов)

Максимальное число TE LSP в наборе.

Min-Bandwidth (32 бита)

Минимальная пропускная способность каждого TE LSP, представленная 32-битовым числом с плавающей точкой в формате IEEE (см. [IEEE.754.1985]), указывающим число битов в секунду.

Тело объекта LOAD-BALANCING имеет фиксированный размер 8 байтов.

Если PCC запрашивает расчёт набора TE LSP с суммарной пропускной способностью X, максимальным числом TE LSP равным N и минимальной пропускной способностью каждого TE LSP равной B, он включает объект BANDWIDTH с пропускной способностью X и объект LOAD-BALANCING с Max-LSP и Min-Bandwidth со значениями N и B, соответственно.

7.17. Объект CLOSE

Объект CLOSE **должен** включаться в каждое сообщение Close и **должен** быть единственным объектом этого типа в сообщении Close. При получении сообщения Close с несколькими объектами CLOSE должен обрабатываться лишь первый такой объект, а остальные объекты CLOSE **должны** игнорироваться.

CLOSE Object-Class = 15.

CLOSE Object-Type = 1.

Формат тела объекта CLOSE показан на рисунке.

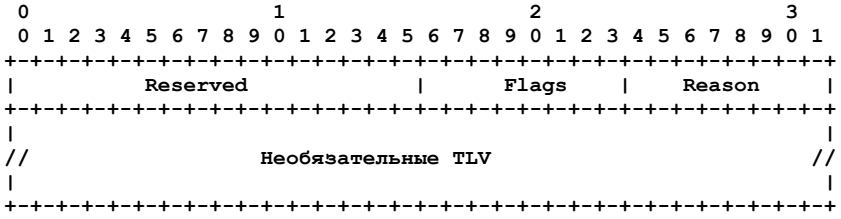

Рисунок 22. Формат объекта CLOSE.

Reserved (16 битов)

Резервное поле. **Должно** устанавливаться в 0 при передаче, а на приёмной стороне **должно** игнорироваться.

Flags (8 битов)

Флаги в настоящее время не определены. Поле Flags **должно** устанавливаться в 0 при передаче, а на приёмной стороне **должно** игнорироваться.

Reason (8 битов)

Указывает причину закрытия сессии PCEP. Установка этого поля не обязательна. Пространством значений поля Reason управляет IANA. Определённые в настоящее время коды причин указаны ниже.

Код причины	*Описание*
1	Не представлено объяснения.
2	Завершён отсчёт таймера DeadTimer.
3	Получено сообщение PCEP с некорректным форматом.
4	Получен неприемлемый номер неизвестного запроса или отклика.
5	Получен неприемлемый номер нераспознанного сообщения PCEP.

В тело объекта CLOSE могут включаться необязательные TLV. Спецификация таких TLV выходит за рамки документа.

8. Вопросы управляемости

Этот раздел следует рекомендациям [PCE-MANAGE].

8.1. Управление функциями и политикой

Реализациям PCEP **следует** разрешать настройку перечисленных ниже параметров сессии PCEP.

- Локальные значения таймеров Keepalive и DeadTimer (т. е. параметры, передаваемые узлом PCEP в сообщении Open).
- Максимальные приемлемые значения удалённых таймеров Keepalive и DeadTimer (т. е. параметры, получаемые от партнёра в сообщении Open).
- Разрешение и запрет согласования.
- При разрешённом согласовании минимальные приемлемые значения таймеров Keepalive и DeadTimer, получаемые от партнёра PCEP.
- таймер SyncTimer.
- Максимальное число сессий, которые могут быть организованы.
- Таймер запросов - время, в течение которого PCC ждёт отклика перед отправкой повторного запроса на расчёт пути (возможно другому PCE).
- MAX-UNKNOWN-REQUESTS.
- MAX-UNKNOWN-MESSAGES.

Эти параметры могут быть заданы по умолчанию для любой сессии PCEP, в которой участвует узел PCEP, или могут применяться к конкретной сессии с данным партнёром. PCEP или определённой группе сессий с конкретной группой партнёров PCEP. Реализациям PCEP **следует** разрешать инициирование сессий PCEP с выбранным подмножеством обнаруженных PCE. Отметим, что выбор PCE определяется локальными потребностями. Реализациям PCEP **следует** разрешать настройку конкретной сессии PCEP с данным партнёром. PCEP, включая установку приведённых ниже параметров.

- IP-адрес партнёра PCEP.
- Роль узла PCEP - PCC, PCE или оба сразу.
- Следует ли узлу PCEP организовывать сессию PCEP сразу или ждать инициативы партнёра.
- Параметры сессии PCEP, указанные выше, если они отличаются от принятых по умолчанию.
- Набор правил PCEP, включая тип операций, разрешённых партнёру PCEP (например, расчёт различных путей, синхронизация и т. п.).

Реализация PCEP **должна** разрешать ограничение набора партнёров PCEP, которые могут инициировать сессию PCEP с узлом PCEP (например, список уполномоченных узлов PCEP, все узлы PCEP в области, все узлы PCEP в AS).

8.2. Модели информации и данных

Модуль PCEP MIB определённый в [PCEP-MIB], описывает управляемые объекты для моделирования коммуникаций PCEP, включая:

- конфигурацию и статус клиента PCEP;
- конфигурацию и информацию узла PCEP;
- конфигурацию и информацию сессии PCEP;
- уведомления о смене состояния сессии PCEP.

8.3. Детектирование и мониторинг живучести

PCEP включает механизм keepalive для проверки живучести партнёра PCEP и процедуру уведомления, позволяющую PCE сообщить о своей перегрузке клиентам PCC. Определены также процедуры мониторинга живучести и производительности данной цепочки PCE (в случае расчёта пути несколькими PCE) [PCE-MONITOR].

8.4. Проверка корректности работы

Проверка корректности работы PCEP может быть выполнена путём мониторинга различных параметров. Реализациям PCEP **следует** поддерживать проверку перечисленных ниже параметров.

- Время отклика (минимальное, среднее, максимальное) на уровне PCE.
- Отказы сессий PCEP.
- Продолжительность активного состояния сессии.
- Число повреждённых сообщений.
- Число отказов при расчётах.
- Число запросов, на которые не было получено отклика в течение настраиваемого времени после проверки наличия хотя бы одного пути, соответствующего заданному набору ограничений.

Реализации PCEP **следует** записывать ошибки (например, повреждённые сообщения, нераспознанные объекты) в системный журнал.

8.5. Требования к другим протоколам и функциональным компонентам

PCEP не предъявляет новых требований к другим протоколам. Поскольку PCEP работает на базе транспорта TCP, для управления PCEP можно применять механизма управления TCP (такие как TCP MIB [RFC4022]).

Механизмы PCE Discovery ([RFC5088], [RFC5089]) могут влиять на работу PCEP. Для предотвращения частой организации и удаления сессий PCEP в результате высокой частоты обнаружения и потери PCE (Discoveries/Disappearances) **рекомендуется** применять то или иное демпфирование организации сессий PCEP.

8.6. Влияние на работу сети

Для предотвращения неприемлемого влияния на работу сети реализациям **следует** разрешать ограничивать число сессий, которые может организовать узел PCEP, и **можно** разрешать ограничивать частоту сообщений, передаваемых узлом PCEP и получаемых от партнёра. **Можно** также разрешать передачу уведомления о превышении порога частоты передачи.

9. Взаимодействие с IANA

Агентство IANA выделило значения протокольных параметров PCEP (сообщения, объекты, TLV).

Агентство IANA организовало новый реестр верхнего уровня для всех кодов и субреестров PCEP.

Для каждого из новых реестров задана процедура IETF Consensus - новые значения выделяются с согласия IETF (см. [RFC5226]). В частности, новые значения выделяются через RFC, одобренные IESG. Обычно IESG запрашивает информацию о предполагаемых назначениях у соответствующи лиц (например, в рабочей группе, если она имеется).

9.1. Порт TCP

Для PCEP зарегистрирован порт TCP 4189.

9.2. Сообщения PCEP

Агентство IANA создало реестр для сообщений PCEP. Каждое сообщение PCEP имеет определённый тип.

Значение	*Смысл*	*Документ*
1	Open	Данный документ
2	Keepalive	Данный документ
3	Path Computation Request	Данный документ
4	Path Computation Reply	Данный документ
5	Notification	Данный документ
6	Error	Данный документ
7	Close	Данный документ

9.3. Объекты PCEP

Агентство IANA создало реестр объектов PCEP. Каждый объект PCEP имеет класс Object-Class и тип Object-Type.

Object-Class	*Имя*	*Документ*
1	OPEN, Object-Type = 1	Данный документ

2	RP, Object-Type = 1	Данный документ
3	NO-PATH, Object-Type = 1	Данный документ
4	END-POINTS, Object-Type = 1 - адрес IPv4, = 2 - адрес IPv6	Данный документ
5	BANDWIDTH, Object-Type = 1 - запрошенная полоса, = 2 - полоса имеющегося IPv6 TE LSP, для которого выполняется повторная оптимизация.	Данный документ
6	METRIC, Object-Type = 1	Данный документ
7	ERO, Object-Type = 1	Данный документ
8	RRO, Object-Type = 1	Данный документ
9	LSPA, Object-Type = 1	Данный документ
10	IRO, Object-Type = 1	Данный документ
11	SVEC, Object-Type = 1	Данный документ
12	NOTIFICATION, Object-Type = 1	Данный документ
13	PCEP-ERROR, Object-Type = 1	Данный документ
14	LOAD-BALANCING, Object-Type = 1	Данный документ
15	CLOSE, Object-Type = 1	Данный документ

9.4. Базовый заголовок сообщения PCEP

Агентство IANA создало реестр для поддержки поля Flag в базовом заголовке сообщений PCEP.

Номера битов могут выделяться только по процедуре IETF Consensus. Для каждого бита отслеживается:

- номер бита (отсчёт с 0 для старшего бита);
- описание возможности;
- RFC с определением.

В настоящее время битов для базового заголовка сообщений PCEP не определено.

9.5. Поле флагов объекта OPEN

Агентство IANA создало реестр для поля Flag объекта OPEN.

Номера битов могут выделяться только по процедуре IETF Consensus. Для каждого бита отслеживается:

- номер бита (отсчёт с 0 для старшего бита);
- описание возможности;
- RFC с определением.

В настоящее время биты флагов объекта OPEN не определены.

9.6. Объект RP

Номера битов могут выделяться только по процедуре IETF Consensus. Для каждого бита отслеживается:

- номер бита (отсчёт с 0 для старшего бита);
- описание возможности;
- RFC с определением.

Определены несколько битов для поля флагов объекта RP, показанных ниже.

Бит	***Описание***	***Документ***
26	Строгий/нестрогий	Данный документ
27	Двухсторонний	Данный документ
28	Повторная оптимизация	Данный документ
29-31	Приоритет	Данный документ

9.7. Поле флагов объекта NO-PATH

Агентство IANA создало реестр для полей NI и Flag объекта NO-PATH.

Значение	***Описание***	***Документ***
0	Не найдено пути, соответствующего заданным ограничениям.	Данный документ
1	Цепочка PCE разорвана.	Данный документ

Номера битов могут выделяться только по процедуре IETF Consensus. Для каждого бита отслеживается:

- номер бита (отсчёт с 0 для старшего бита);
- описание возможности;
- RFC с определением.

Определён один бит для поля флагов объекта NO-PATH.

Значение	***Описание***	***Документ***
0	Указаны невыполненные ограничения.	Данный документ

9.8. Объект METRIC

Агентство IANA создало реестр для полей T и Flag объекта METRIC.

Ниже приведены значения поля T.

Значение	***Описание***	***Документ***
1	Метрика IGP.	Данный документ
2	Метрика TE.	Данный документ

3	Число интервалов.	Данный документ

Номера битов могут выделяться только по процедуре IETF Consensus. Для каждого бита отслеживается:

- номер бита (отсчёт с 0 для старшего бита);
- описание возможности;
- RFC с определением.

Определены несколько битов для поля флагов объекта METRIC, показанных ниже.

Бит	*Описание*	*Документ*
6	Рассчитанная метрика	Данный документ
7	Привязка (Bound)	Данный документ

9.9. Поле флагов объекта LSPA

Агентство IANA создало реестр для поля Flag объекта LSPA.

Номера битов могут выделяться только по процедуре IETF Consensus. Для каждого бита отслеживается:

- номер бита (отсчёт с 0 для старшего бита);
- описание возможности;
- RFC с определением.

Определён один бит для поля флагов объекта LSPA.

Бит	*Описание*	*Документ*
7	Желательна локальная защита	Данный документ

9.10. Поле флагов объекта SVEC

Агентство IANA создало реестр для поля Flag объекта SVEC.

Номера битов могут выделяться только по процедуре IETF Consensus. Для каждого бита отслеживается:

- номер бита (отсчёт с 0 для старшего бита);
- описание возможности;
- RFC с определением.

Определены 3 бита для поля флагов объекта SVEC, показанных ниже.

Бит	*Описание*	*Документ*
21	Различие SRLG	Данный документ
22	Различие узлов	Данный документ
23	Различие каналов	Данный документ

9.11. Объект NOTIFICATION

Агентство IANA создало реестр для полей Notification-type и Notification-value объекта NOTIFICATION и управляет пространством кодов.

Notification-type	*Notification-value*	*Описание*	*Документ*
1		Отмена ожидающего запроса	Данный документ
	1	PCC отменил набор ожидающих запросов	
	2	PCE отменил набор ожидающих запросов	
2		Перегрузка PCE	Данный документ
	1	Насыщение PCE	
	2	PCE вышел из насыщения	

Агентство IANA создало реестр для поля Flag объекта NOTIFICATION.

Номера битов могут выделяться только по процедуре IETF Consensus. Для каждого бита отслеживается:

- номер бита (отсчёт с 0 для старшего бита);
- описание возможности;
- RFC с определением.

В настоящее время не определено битов поля Flag в объекте NOTIFICATION.

9.12. Объект PCEP-ERROR

Агентство IANA создало реестр для полей Error-Type и Error-value объекта PCEP-ERROR и поддерживает его.

Для каждой ошибки PCEP определены значения Error-Type и Error-value приведённые ниже.

Error-Type	*Error-value*	*Описание*	*Документ*
1		Отказ при организации сессии PCEP.	Данный документ
	1	Приём недействительного сообщения Open или отсутствие Open.	
	2	Не получено сообщения Open до завершения отсчёта таймера OpenWait.	
	3	Неприемлемые или несогласуемые характеристики сессии.	
	4	Неприемлемые, но согласуемые характеристики сессии.	
	5	Получение второго сообщения Open с неприемлемыми характеристиками.	
	6	Получение сообщения PCErr с неприемлемыми характеристиками сессии.	
	7	Не принято сообщений Keepalive или PCErr до завершения отсчёта KeepWait.	
	8	Версия PCEP не поддерживается.	
2		Возможность не поддерживается.	Данный документ

3		Неизвестный объект.	Данный документ
	1	Нераспознанный класс объекта.	
	2	Нераспознанный тип объекта.	
4		Неподдерживаемый объект.	Данный документ
	1	Неподдерживаемый класс объекта.	
	2	Неподдерживаемый тип объекта.	
5		Нарушение политики.	Данный документ
	1	Установлен бит C в объекте METRIC (запрос отвергнут.	
	2	Сброшен бит O в объекте RP (запрос отвергнут.	
6		Отсутствует обязательный объект.	Данный документ
	1	Нет объекта RP.	
	2	Нет объекта RRO для запроса повторной оптимизации (бит R в RP установлен).	
	3	Нет объекта END-POINTS.	
7		Отсутствует запрос синхронизированного расчёта пути.	Данный документ
8		Ссылка на неизвестный запрос.	Данный документ
9		Попытка организовать вторую сессию PCEP.	Данный документ
10		Получение недействительного объекта.	Данный документ
	1	Получение объекта со сброшенным флагом P, хотя данная спецификация требует его установки.	

Агентство IANA создало реестр для поля Flag объекта PCEP-ERROR.

Номера битов могут выделяться только по процедуре IETF Consensus. Для каждого бита отслеживается:

- номер бита (отсчёт с 0 для старшего бита);
- описание возможности;
- RFC с определением.

В настоящее время не определено битов поля Flag в объекте PCEP-ERROR.

9.13. Поле флагов объекта LOAD-BALANCING

Агентство IANA создало реестр для поля Flag объекта LOAD-BALANCING.

Номера битов могут выделяться только по процедуре IETF Consensus. Для каждого бита отслеживается:

- номер бита (отсчёт с 0 для старшего бита);
- описание возможности;
- RFC с определением.

В настоящее время не определено битов поля Flag в объекте LOAD-BALANCING.

9.14. Объект CLOSE

Объект CLOSE **должен** присутствовать в каждом сообщении Close для указания причины разрыва сессии PCEP. Поле reason объекта CLOSE указывает причину разрыва сессии PCEP. Значения поля reason в объекте CLOSE поддерживаются IANA.

Код причины	***Описание***
1	Без объяснения.
2	Таймер DeadTimer.
3	Приём сообщения PCEP с некорректным форматом.
4	Получение неприемлемого числа неизвестных запросов или откликов.
5	Получение неприемлемого числа нераспознанных сообщений PCEP.

Агентство IANA создало реестр для поля Flag объекта CLOSE.

Номера битов могут выделяться только по процедуре IETF Consensus. Для каждого бита отслеживается:

- номер бита (отсчёт с 0 для старшего бита);
- описание возможности;
- RFC с определением.

В настоящее время не определено битов поля Flag в объекте CLOSE.

9.15. Индикаторы типов PCEP TLV

Агентство IANA создало реестр для PCEP TLV.

Код	***Описание***	***Документ***
1	NO-PATH-VECTOR TLV	Данный документ
2	OVERLOAD-DURATION TLV	Данный документ
3	REQ-MISSING TLV	Данный документ

9.16. NO-PATH-VECTOR TLV

IANA управляет пространством флагов, передаваемых в NO-PATH-VECTOR TLV, определённом в этом документе. Биты флагов нумеруются с 0, начиная со старшего бита.

Номера битов могут выделяться только по процедуре IETF Consensus. Для каждого бита отслеживается:

- номер бита (отсчёт с 0 для старшего бита);

- флаг имени;
- ссылка на документ.

Бит	*Описание*	*Документ*
31	PCE в данный момент недоступен	Данный документ
20	Неизвестный адресат	Данный документ
29	Неизвестный отправитель	Данный документ

10. Вопросы безопасности

10.1. Уязвимости

Атаки на PCEP могут нарушать работу активных сетей. Если отклики с рассчитанными путями изменить, это может привести к выбору неприемлемых LSP клиентами PCC. Эти LSP могут проходить через участки сети, где трафик подвергается исследованию или могут попадать в перегруженные или резервные каналы. Отклики с расчётами пути можно атаковать путём изменения сообщений PCRep, подмены PCE или изменения запросов PCReq, вынуждающего PCE выполнить расчёт, отличающийся от изначально запрошенного.

Можно нарушить работу PCE с использованием различных DoS-атак[1], которые могут вызвать перегрузку PCE, ведущую к замедлению откликов клиентам PCC до неприемлемых величин. Это может также приводить к неприемлемо долгому времени восстановления или задержкам организации LSP. В экстремальных случаях это может приводить даже к отказам от выполнения запросов.

Ниже перечислены некоторые атаки, которые могут быть организованы на протокол PCEP:

- обман (подмена PCC или PCE);
- слежка (перехват сообщений);
- фальсификация;
- отказ в обслуживании.

При работе через несколько AS, где требуется взаимодействие между PCE, атаки могут стать особо значимыми с точки зрения бизнеса и качества обслуживания.

Кроме того, отслеживание запросов и откликов PCEP может дать атакующему возможность сбора информации о работе сети. Просматривая сообщения PCEP, злоумышленник может определить схему организации обслуживания в сети и узнать, куда направляется трафик, что делает сеть уязвимой к целевым атакам, а также подвергает риску данные конкретных LSP.

В следующих параграфах рассматриваются механизмы защиты протокола PCEP от атак.

10.2. Методы защиты TCP

Во время написания этого документа TCP-MD5 [RFC2385] был единственным доступным механизмом защиты соединений TCP, на основе которых организуются сессии PCEP.

Как разъяснено в [RFC2385], применение MD5 связано с некоторыми ограничениями и не обеспечивает высокого уровня защиты, как считалось ранее. Реализации PCEP с поддержкой TCP-MD5 **следует** проектировать так, чтобы в будущих версиях можно было легко интегрировать более надёжные методы и алгоритмы для защиты TCP.

Опция аутентификации TCP [TCP-AUTH] (TCP-AO[2]) задаёт новые процедуры защиты для TCP, но ещё не разработана до конца. Поскольку считается, что [TCP-AUTH] будет обеспечивать существенно более надёжную защиту для приложений, использующих TCP, разработчикам следует обновить свои реализации после выхода RFC для TCP-AO.

Реализации **должны** поддерживать TCP-MD5 и следует делать функцию защиты конфигурационной опцией.

Операторы должны учитывать, что некоторые развёрнутые реализации PCEP могут применять предварительные варианты [TCP-AUTH] и потребуется настройка политики защиты для безопасных коммуникаций между узлами PCEP с поддержкой TCP-AO и без неё.

Другим вариантом защиты транспорта TCP является протокол TLS[3] [RFC5246]. Этот протокол обеспечивает защиту от прослушивания, искажения и подделки. Однако TLS не защищает сами соединения TCP, поскольку не применяется аутентификация заголовков TCP. Поэтому сохраняется уязвимость к атакам со сбросом TCP (от которых защищает TCP-MD5). Однако применение TLS требует спецификации инициирования протоколом PCEP согласования TLS и способа интерпретации сертификатов, передаваемых в TLS. Такая спецификация выходит за рамки документа, но может стать предметом будущей работы.

10.3. Аутентификация и защита целостности PCEP

Проверка подлинности и целостности позволяет получателю PCEP убедиться, что сообщение действительно исходит из указанного узла и не было изменено в процессе передачи.

Механизм TCP-MD5 [RFC2385], упомянутый выше, обеспечивает такую защиту с учётом проблем, отмеченных в [RFC2385] и [RFC4278]. Эти проблемы будут решаться с помощью [TCP-AUTH].

10.4. Конфиденциальность PCEP

Обеспечение конфиденциальности коммуникаций PCEP очень важно, особенно при работе в нескольких AS, где конечные точки PCEP размещаются в разных AS, поскольку злоумышленник может получить важную информацию о рассчитанных путях и ресурсах, перехватив сообщения PCE.

[1]Denial-of-service - отказ в обслуживании.
[2]TCP Authentication Option.
[3]Transport Layer Security - защита транспортного уровня.

Конфиденциальность PCEP можно обеспечить путём шифрования. TCP **может** работать через туннели IPsec [RFC4303] для обеспечения нужного шифрования. Отметим, что IPsec может также обеспечить контроль подлинности и целостности и тогда применение TCP-MD5 или TCP-AO не потребуется. Однако есть опасения по части сложности настройки и применения IPsec. Использование IPSec с PCEP выходит за рамки документа и может быть рассмотрено в отдельной работе.

10.5. Настройка ключей и обмен ими

Для аутентификации, защиты от от несанкционированного доступа и шифрования требуется использование ключей отправителем и получателем.

Возможна настройка сеансовых ключей, но это может быть обременительно для операторов (а также для протокола BGP, как указано в [BGP-SEC]). При небольшом числе PCC и PCE в сети **можно** задать ключи вручную, но нужно учитывать уязвимости таких механизмов (например, конфигурационные ошибки, социальная психология и невнимательность оператора могут приводить к нарушению защиты). Кроме того, настроенные вручную ключи скорей всего будут обновляться более редко, что также повышает риск. При большом числе PCC и PCE на оператора ложится значительная нагрузка по настройке и поддержке, поскольку требуется настроить каждый PCC и PCE.

Другим вариантом является применение групповых ключей. Такой ключ известен всем членам домена доверия. Поскольку маршрутизаторы в области IGP или AS являются частью домена доверия [MPLS-SEC], групповой ключ PCEP **может** быть известен всем PCC и PCE в области IGP или AS. Использование групповых ключей существенно снижает нагрузку на оператора по настройке, обеспечивая защиту PCEP от атак извне. Однако следует отметить, что с ростом числа элементов возрастает и риск раскрытия (утечки) ключа.

При использовании групповых ключей нужно задавать отдельные ключи для коммуникаций между PCE в разных домена (например, AS), но число таких взаимодействий обычно будет очень мало.

Обнаружение PCE ([RFC5088] и [RFC5089]) является важным механизмом для развёртывания PCEP в больших сетях. Этот механизм позволяет PCC обнаруживать наличие PCE в сети без необходимости настройки. Очевидно, что обнаруженные PCE не будут настроены и PCC не будет знать нужного для работы ключа. Решить эту проблему можно тремя способами, указанными ниже, с сохранением некоторых аспектов безопасности.

- PCC могут применять групповой ключ, как указано выше.
- PCC могут использовать тот или иной протокол защищённого обмена ключами с PCE (например, протокол Internet Key Exchange версии 2 - IKE [RFC4306]). Недостатком этого подхода является то, что не на всех маршрутизаторах поддерживается IKE и это может создать преграду для развёртывания PCEP. Детали такого подхода выходят за рамки документа и могут быть рассмотрены в отдельной работе.
- PCC могут использовать сервер ключей для получения ключа, позволяющего работать с PCE. Это лишь переносит проблему, поскольку связь PCC с сервером ключей также требуется защищать (например, с помощью Kerberos [RFC4120]), но обеспечивает некоторое (незначительное) преимущество в плане расширяемости, если PCC нужно знать ключи нескольких PCE, поскольку здесь достаточно знать лишь ключ сервера ключей. Отметим, что реализации серверов ключей в настоящее время очень ограничены. Детали такого подхода выходят за рамки документа и могут быть рассмотрены в отдельной работе.

Связи PCEP вероятно будут продолжительными даже при неоднократном закрытии и восстановлении сессий PCEP. Если протокольные отношения сохраняются долго или для большого числа взаимодействий, **рекомендуется** регулярная замена ключей, используемых партнёрами [RFC4107]. Отметим, что TCP-MD5 не позволяет менять ключи без разрыва соединений TCP что будет приводить к необходимости перезапуска сессий PCEP. Это может создать проблему для PCEP. Отметим также, что планы внедрения TCP-AO [TCP-AUTH] позволяют менять ключи динамически для активного соединения TCP.

При использовании обмена ключами (например, IKE), сравнительно просто поддерживать динамический обмен ключами и применять его для PCEP.

Отметим, что управление по основному каналу для ключей TCP-AO [TCP-AUTH] в настоящее время не решено.

В [RFC3562] указаны некоторые проблемы управления ключами для защищённых соединений TCP.

10.6. Политика доступа

Несанкционированный доступ к функциям PCE является одним из вариантов атак. Это может быть не просто атака, нацеленная на отказ в обслуживании (см. параграф 10.7), но и механизм получения злоумышленником важной информации о сети и правилах её работы просто путём отправки обманных запросов на расчёт путей. Кроме того, ложные запросы расчётов могут использоваться для прогнозирования размещения трафика в сети при выполнении реальных запросов, что позволяет злоумышленнику нацелить атаку на конкретные ресурсы сети.

Элементам PCE **следует** поддерживать настраиваемую политику доступа. При использовании аутентификации контроль доступа можно обеспечить за счёт настройки или обмена ключами, как описано в параграфе 10.5. Более простые правила **можно** настроить на PCE в форме списков доступа, содержащих IP-адреса легитимных PCC. Правила **следует** делать настраиваемыми, чтобы ограничить типы запросов, поддерживаемые для разных PCC.

Рекомендуется регистрировать нарушения политики доступа в системном журнале PCE и проверять этот журнал для обнаружения попыток атаковать PCE. Такие механизмы **должны** быть облегчёнными, чтобы нельзя было организовать с их помощью DoS-атаки (см. параграф 10.7).

10.7. Защита от DoS-атак

DoS-атаки могут быть организованы на уровне TCP или PCEP, т. е. PCE можно атаковать через TCP или в созданной сессии PCEP.

10.7.1. Защита от DoS-атак на TCP

PCEP может быть целью DoS-атак на TCP, таких как SYN-атаки, как и все протоколы, работающие на основе TCP. В спецификациях других протоколов этот вопрос достаточно изучен и PCEP может воспользоваться набранным опытом. Читателям рекомендуется обратиться, например, к спецификации протокола распространения меток LDP[1] [RFC5036]. Для защиты от DoS-атак на TCP реализации PCEP могут поддерживать приведённые ниже методы.

- PCEP использует один зарегистрированный порт для всех коммуникаций. PCE **следует** слушать соединения TCP только на портах, где ожидаются соединения.
- PCE **может** реализовать список доступа для прямого отбрасывания (reject или discard) попыток соединения TCP от неуполномоченных PCC.
- PCE **не следует** разрешать множество соединений TCP с одним PCC через зарегистрированный порт PCEP.
- PCE **может** потребовать использования опции MD5 для всех соединений TCP и **может** отвергать (или отбрасывать) все попытки организации соединения без MD5. PCE **недопустимо** воспринимать SYN-пакеты с недействительной суммой MD5 для сегмента. Отметим однако, что применение MD5 требует от получателя использовать ресурсы CPU для расчёта контрольной суммы до того, как он сможет принять решение об отбрасывании приемлемого по остальным параметрам сегмента SYN.

10.7.2. Формирование и правила на входе

Реализация PCEP может подвергаться DoS-атакам в легитимной сессии PCEP. Например, PCC может передавать очень большое число сообщений PCReq, перегружающих PCE или вызывающих постановку в очередь запросов от других PCC.

Отметим, что прямое использование поля Priority в объекте RP для приоритизации полученных запросов не обеспечивает какой-либо защиты, поскольку атакующий может установить для всех запросов высший приоритет.

Поэтому **рекомендуется** включать в реализации PCE механизмы формирования и применения правил на входе, которые ограничивают запросы от одного PCC или используют методы постановки в очередь и/или снижения приоритета для слишком активных PCC.

Такие механизмы **могут** быть установлены по умолчанию, но их **следует** делать настраиваемыми. Эти механизмы особенно важны в средах с множеством сервис-провайдеров для защиты ресурсов одного провайдера от необоснованного, чрезмерного или злонамеренного использования элементами PCE другого провайдера.

11. Благодарности

Авторы благодарны Dave Oran, Dean Cheng, Jerry Ash, Igor Bryskin, Carol Iturrade, Siva Sivabalan, Rich Bradford, Richard Douville, Jon Parker, Martin German и Dennis Aristow за полезные предложения. Спасибо также Fabien Verhaeghe за полезные дискуссии и предложения. David McGrew и Brian Weis внесли важный вклад в раздел «Вопросы безопасности».

Ross Callon, Magnus Westerlund, Lars Eggert, Pasi Eronen, Tim Polk, Chris Newman и Russ Housley внесли существенные предложения в процессе обзора IESG.

12. Литература

12.1. Нормативные документы

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

[RFC2205] Braden, B., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification", RFC 2205, September 1997.

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

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

[RFC3473] Berger, L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003.

[RFC3477] Kompella, K. and Y. Rekhter, "Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)", RFC 3477, January 2003.

[RFC4090] Pan, P., Swallow, G., and A. Atlas, "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, May 2005.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

12.2. Дополнительная литература

[BGP-SEC] Christian, B. and T. Tauber, "BGP Security Requirements", Work in Progress, November 2008.

[IEEE.754.1985] IEEE Standard 754, "Standard for Binary Floating-Point Arithmetic", August 1985.

[INTER-LAYER] Oki, E., Roux, J., Kumaki, K., Farrel, A., and T. Takeda, "PCC-PCE Communication and PCE Discovery Requirements for Inter-Layer Traffic Engineering", Work in Progress, January 2009.

[MPLS-SEC] Fang, L. and M. Behringer, "Security Framework for MPLS and GMPLS Networks", Work in Progress, November 2008.

[1]Label Distribution Protocol.

[PCE-MANAGE] Farrel, A., "Inclusion of Manageability Sections in PCE Working Group Drafts", Work in Progress[1], January 2009.

[PCE-MONITOR] Vasseur, J., Roux, J., and Y. Ikejiri, "A set of monitoring tools for Path Computation Element based Architecture", Work in Progress[2], November 2008.

[PCEP-MIB] Stephan, E. and K. Koushik, "PCE communication protocol (PCEP) Management Information Base", Work in Progress[3], November 2008.

[RBNF] Farrel, A., "Reduced Backus-Naur Form (RBNF) A Syntax Used in Various Protocol Specifications", Work in Progress[4], November 2008.

[RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[RFC3471] Berger, L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description", RFC 3471, January 2003.

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

[RFC3785] Le Faucheur, F., Uppili, R., Vedrenne, A., Merckx, P., and T. Telkamp, "Use of Interior Gateway Protocol (IGP) Metric as a second MPLS Traffic Engineering (TE) Metric", BCP 87, RFC 3785, May 2004.

[RFC4022] Raghunarayan, R., "Management Information Base for the Transmission Control Protocol (TCP)", RFC 4022, March 2005.

[RFC4101] Rescorla, E. and IAB, "Writing Protocol Models", RFC 4101, June 2005.

[RFC4107] Bellovin, S. and R. Housley, "Guidelines for Cryptographic Key Management", BCP 107, RFC 4107, June 2005.

[RFC4120] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005.

[RFC4278] Bellovin, S. and A. Zinin, "Standards Maturity Variance Regarding the TCP MD5 Signature Option (RFC 2385) and the BGP-4 Specification", RFC 4278, January 2006.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[RFC4306] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

[RFC5420] Farrel, A., Ed., Papadimitriou, D., Vasseur, JP., and A. Ayyangarps, "Encoding of Attributes for MPLS LSP Establishment Using Resource Reservation Protocol Traffic Engineering (RSVP-TE)", RFC 5420, February 2009.

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.

[RFC4657] Ash, J. and J. Le Roux, "Path Computation Element (PCE) Communication Protocol Generic Requirements", RFC 4657, September 2006.

[RFC4674] Le Roux, J., "Requirements for Path Computation Element (PCE) Discovery", RFC 4674, October 2006.

[RFC4927] Le Roux, J., "Path Computation Element Communication Protocol (PCECP) Specific Requirements for Inter-Area MPLS and GMPLS Traffic Engineering", RFC 4927, June 2007.

[RFC5036] Andersson, L., Minei, I., and B. Thomas, "LDP Specification", RFC 5036, October 2007.

[RFC5088] Le Roux, JL., Vasseur, JP., Ikejiri, Y., and R. Zhang, "OSPF Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5088, January 2008.

[RFC5089] Le Roux, JL., Vasseur, JP., Ikejiri, Y., and R. Zhang, "IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5089, January 2008.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5376] Bitar, N., Zhang, R., and K. Kumaki, "Inter-AS Requirements for the Path Computation Element Communication Protocol (PCECP)", RFC 5376, November 2008.

[TCP-AUTH] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", Work in Progress[5], November 2008.

Приложение A. Конечный автомат PCEP

В этом разделе описан конечный автомат PCEP FSM (finite state machine).

Ниже перечислены переменные PCEP.

[1]Работа опубликована в RFC 6123. *Прим. перев.*
[2]Работа опубликована в RFC 5886. *Прим. перев.*
[3]Работа опубликована в RFC 7420. *Прим. перев.*
[4]Работа опубликована в RFC 5511. *Прим. перев.*
[5]Работа опубликована в RFC 5925. *Прим. перев.*

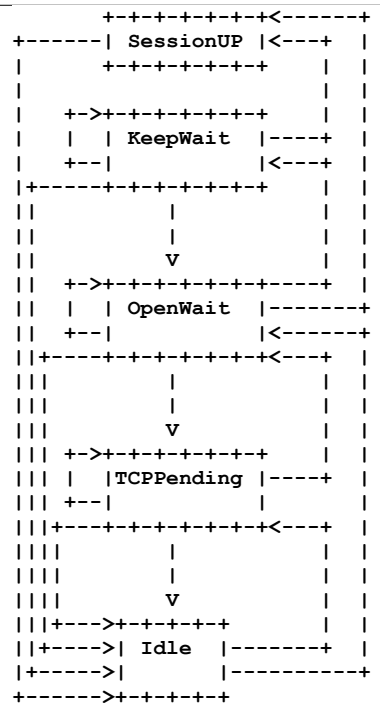

Рисунок 23. Конечный автомат PCEP для клиента PCC.

Connect

Таймер (в секундах) запускаемый после инициализации соединения TCP через зарегистрированный для PCEP порт TCP. Значение таймера Connect составляет 60 секунд.

ConnectRetry

Число попыток организации соединения TCP с партнёром PCEP при возникновении отказов.

ConnectMaxRetry

Максимальное число попыток системы организовать соединение TCP через порт PCEP до перехода в состояние Idle. ConnectMaxRetry имеет значение 5.

OpenWait

Таймер, соответствующий времени, в течение которого узел PCEP будет ждать сообщения Open от партнёра перед тем, как система освободит ресурсы PCEP и вернётся в состояние Idle. Таймер OpenWait имеет фиксированное значение 60 секунд.

KeepWait

Таймер, соответствующий времени, в течение которого узел PCEP будет ждать приёма Keepalive или PCErr от партнёра PCEP и по истечении которого система будет освобождать ресурс PCEP и возвращаться в состояние Idle. Таймер KeepWait имеет фиксированное значение 60 секунд.

OpenRetry

Число повторов сообщений Open с неприемлемыми характеристиками сессии PCEP, воспринимаемых системой.

Ниже приведены две определённые протоколом переменные состояния.

RemoteOK

Логическая переменная, устанавливаемая в 1 при получении системой приемлемого сообщения Open.

LocalOK

Логическая переменная, устанавливаемая в 1 при получении системой сообщения Keepalive, подтверждающего восприятие партнёром отправленного ему сообщения Open.

Состояние Idle

Исходное состояние PCEP, когда PCEP (система) ждёт инициализации, включённой пользователем вручную (конфигурацию) или автоматически выполненной по тому или иному событию. В состоянии Idle ресурсы PCEP (память, возможные процессы и т. п.) выделяются, но сообщения PCEP ещё не приняты от партнёра. Система слушает зарегистрированный для PCEP порт TCP.

Переменные инициализируются приведёнными ниже значениями.

TCPRetry=0,
LocalOK=0,
RemoteOK=0,
OpenRetry=0.

При детектировании локального инициализационного события (например, пользовательская настройка для организации сессии PCEP с определенным партнёром PCEP, локальное событие, организующее сессию PCEP с партнёром PCEP, такое как автоматическое обнаружение партнёра. PCEP) система:

- инициирует соединение TCP с партнёром PCEP;
- запускает таймер Connect;
- переходит в состояние TCPPending.

Если соединение TCP организовано, система при получении соединения TCP через зарегистрированный порт PCEP TCP:

- передаёт сообщение Open;
- запускает таймер OpenWait;
- переходит в состояние OpenWait.

Если организация соединения завершилась отказом, система остаётся в состоянии Idle. Все другие события в состоянии Idle игнорируются.

Предполагается, что реализации будут применять экспоненциально увеличиваемый таймер между автоматически генерируемыми событиями Initialization и попытками повторить организацию соединения TCP.

Состояние TCPPending

При успешной организации соединения TCP система:

- передаёт сообщение Open;
- запускает таймер OpenWait;
- переходит в состояние OpenWait.

При отказе организации соединения TCP (ошибка на этапе организации соединения TCP) или завершении отсчёта таймера Connect:

- если ConnectRetry = ConnectMaxRetry, система переходит в состояние Idle;
- если ConnectRetry < ConnectMaxRetry, система:
  1. инициирует соединение TCP с партнёром PCEP;
  2. инкрементирует переменную ConnectRetry;
  3. перезапускает таймер Connect;
  4. остаётся в состоянии TCPPending.

В ответ на остальные события система освобождает ресурсы PCEP, выделенные для данного партнёра., и переходит в состояние Idle.

Состояние OpenWait

В состоянии OpenWait система ждёт сообщения Open от партнёра. PCEP.

Если система получает сообщение Open от партнёра. PCEP до завершения отсчёта таймера OpenWait, она сначала проверяет все свои сессии, находящиеся в состоянии OpenWait или KeepWait. Если сессия с этим партнёром PCEP (тот же адрес IP) уже имеется, система выполняет процедуру устранения конфликтов:

- если система инициировала текущую сессию и имеет меньший адрес IP, нежели партнёр PCEP, система закрывает соединение TCP, освобождает ресурсы PCEP, выделенные для ожидающей сессии, и переходит в состояние Idle;
- если система инициировала текущую сессию и имеет больший адрес IP, нежели партнёр PCEP, система закрывает соединение TCP, освобождает ресурсы PCEP, выделенные для ожидающей сессии, и переходит в состояние Idle;
- в остальных случаях система проверяет атрибуты сессии PCEP (частота Keepalive, DeadTimer и т. п.).

При обнаружении ошибки (например, некорректное сообщение Open, приём сообщения, отличного от Open, наличие двух объектов OPEN) PCEP генерирует уведомление об ошибке, а узел PCEP передаёт сообщение PCErr с Error-Type=1 и Error-value=1. Система освобождает ресурсы PCEP, выделенные для партнёра. PCEP, закрывает соединение TCP и переходит в состояние Idle.

Если ошибок не обнаружено, OpenRetry=1, но параметры сессии не приемлемы, узел PCEP передаёт сообщение PCErr с Error-Type=1 и Error-value=5, а система освобождает ресурсы PCEP, выделенные для этого партнёра., и возвращается в состояние Idle.

Если ошибок не обнаружено и параметры сессии приемлемы для локальной системы, она:

- передаёт сообщение Keepalive партнёру PCEP;
- запускает таймер Keepalive;
- устанавливает RemoteOK = 1.

Если LocalOK=1, система сбрасывает таймер OpenWait и переходит в состояние UP.

Если LocalOK=0, система сбрасывает таймер OpenWait, запускает таймер KeepWait и переходит в состояние KeepWait.

Если ошибок не обнаружено, но параметры сессии не приемлемы и не согласуемы, узел PCEP передаёт PCErr с Error-Type=1 и Error-value=3, а система освобождает ресурсы PCEP, выделенные для этого партнёра., и возвращается в состояние Idle.

Если ошибок не обнаружено, OpenRetry=0, а параметры сессии (такие как период Keepalive или таймер DeadTimer) не приемлемы, но согласуемы, система:

- инкрементирует OpenRetry;
- передаёт сообщение PCErr с Error-Type=1 и Error-value=4, содержащее подходящие характеристики сессии;
- если LocalOK=1, система перезапускает таймер OpenWait и остаётся в состоянии OpenWait;
- если LocalOK=0, система сбрасывает таймер OpenWait, запускает таймер KeepWait и переходит в состояние KeepWait.

Если не было получено сообщения Open до завершения отсчёта таймера OpenWait, узел PCEP передаёт сообщение PCErr с Error-Type=1 и Error-value=2, система освобождает ресурсы PCEP, выделенные для этого партнёра., закрывает соединение TCP и переходит в состояние Idle.

В ответ на все прочие события система освобождает ресурсы PCEP, выделенные для этого партнёра., и переходит в состояние Idle.

Состояние KeepWait

В состоянии Keepwait система ждёт от партнёра. PCEP сообщения Keepalive, подтверждающего сообщение Open, или сообщения PCErr в ответ на неприемлемые характеристики сессии PCEP, предложенные в сообщении Open.

При обнаружении ошибки (например, некорректное сообщение Keepalive) PCEP генерирует уведомление и узел PCEP передаёт сообщение PCErr с Error-Type=1 и Error-value=1. Система освобождает ресурсы PCEP, выделенные для этого партнёра., закрывает соединение TCP и переходит в состояние Idle.

Если сообщение Keepalive принято до завершения отсчёта таймера KeepWait, система устанавливает LocalOK=1 и

- если RemoteOK=1, система сбрасывает таймер KeepWait и переходит в состояние UP;
- если RemoteOK=0, система сбрасывает таймер KeepWait, запускает таймер OpenWait и переходит в состояние OpenWait.

Если получено сообщение PCErr до завершения отсчёта таймера KeepWait:

1. если предложенные значения не приемлемы, узел PCEP передаёт сообщение PCErr с Error-Type=1 и Error-value=6, а система освобождает ресурсы PCEP, выделенные для этого партнёра., закрывает соединение TCP и переходит в состояние Idle;
2. если предложенные значения приемлемы, система подстраивает характеристики сессии PCEP в соответствии с предложенными в PCErr значениями, перезапускает таймер KeepWait и передаёт сообщение Open. Если RemoteOK=1, система перезапускает таймер KeepWait и остаётся в состоянии KeepWait. Если RemoteOK=0, система сбрасывает таймер KeepWait, запускает таймер OpenWait и переходит в состояние OpenWait.

Если не было получено сообщения Keepalive или PCErr к моменту завершения отсчёта таймера KeepWait, узел PCEP передаёт сообщение PCErr с Error-Type=1 и Error-value=7, а система освобождает ресурсы PCEP, выделенные для этого партнёра., закрывает соединение TCP и переходит в состояние Idle.

В ответ на все прочие события система освобождает ресурсы PCEP, выделенные для этого партнёра., и переходит в состояние Idle.

Состояние UP

В состоянии UP узел PCEP начинает обмен сообщениями PCEP в соответствии с характеристиками сессии.

По завершении отсчёта таймера Keepalive система перезапускает таймер и передаёт сообщение Keepalive.

Если не было получено сообщений PCEP (Keepalive, PCReq, PCRep, PCNtf) от партнёра. PCEP до завершения отсчёта DeadTimer, система прерывает сессию PCEP в соответствии с процедурой, определённой в параграфе 6.8, освобождает ресурсы PCEP, выделенные для партнёра., закрывает соединение TCP и переходит в состояние Idle.

При получении некорректно сформированного сообщения система прерывает сессию PCEP в соответствии с процедурой, определённой в параграфе 6.8, освобождает ресурсы PCEP, выделенные для партнёра., закрывает соединение TCP и переходит в состояние Idle.

Если система обнаруживает попытку партнёра. PCEP организовать второе соединение TCP, она останавливает организацию этого соединения и передаёт сообщение PCErr с Error-Type=9.

При отказе соединения TCP система освобождает ресурсы PCEP, выделенные для партнёра., закрывает соединение TCP и переходит в состояние Idle.

Приложение B. Переменные PCEP

Ниже перечислены конфигурационные переменные PCEP.

Таймер Keepalive

Минимальный интервал между последовательными сообщениями PCEP (Keepalive, PCReq, PCRep, PCNtf), передаваемыми партнёру PCEP. Рекомендуемое значение таймера Keepalive составляет 30 секунд.

DeadTimer

Интервал, по истечении которого узел PCEP считает сессию неработающей (down), если не было получено ни одного сообщения PCEP.

SyncTimer

Таймер, применяемый в случае синхронизированных запросов расчёта путей с использованием объектов SVEC, определённых в параграфе 7.13.3. Рассмотрим случай, когда сообщение PCReq, принятое PCE, содержит объект SVEC, указывающий M синхронизированных запросов расчёта путей. Если после завершения отсчёта SyncTimer все M запросов расчёта не были получены, возникает протокольная ошибка и элемент PCE **должен** отвергнуть все запросы расчёта путей. Назначение SyncTimer состоит в предотвращении хранения не использованных синхронизированных запросов, один из которых был по той или иной причине потерян (например, в результате некорректного поведения PCC). Таким образом, значение SyncTimer должно быть достаточно велико, чтобы отсчёт таймера не завершался при нормальных условиях. **Рекомендуемое** значение SyncTimer составляет 60 секунд.

MAX-UNKNOWN-REQUESTS

Рекомендуемое значение - 5.

MAX-UNKNOWN-MESSAGES

Рекомендуемое значение - 5.

Приложение C. Участники работы

Этот документ является результатом работы перечисленных ниже людей и редакторов, указанных в конце документа.

Arthi Ayyangar
Juniper Networks
1194 N. Mathilda Ave
Sunnyvale, CA 94089
USA
EMail: arthi@juniper.net

Adrian Farrel
Old Dog Consulting
Phone: +44 (0) 1978 860944
EMail: adrian@olddog.co.uk

Eiji Oki
NTT
Midori 3-9-11
Musashino, Tokyo, 180-8585
JAPAN
EMail: oki.eiji@lab.ntt.co.jp

Alia Atlas
British Telecom
EMail: akatlas@alum.mit.edu

Andrew Dolganow
Alcatel
600 March Road
Ottawa, ON K2K 2E6
CANADA
EMail: andrew.dolganow@alcatel.com

Yuichi Ikejiri
NTT Communications Corporation
1-1-6 Uchisaiwai-cho, Chiyoda-ku

Tokyo, 100-819
JAPAN
EMail: y.ikejiri@ntt.com

Kenji Kumaki
KDDI Corporation
Garden Air Tower Iidabashi, Chiyoda-ku,
Tokyo, 102-8460
JAPAN
EMail: ke-kumaki@kddi.com

Адреса авторов
JP Vasseur (редактор)
Cisco Systems
1414 Massachusetts Avenue
Boxborough, MA 01719
USA
EMail: jpv@cisco.com

JL Le Roux (редактор)
France Telecom
2, Avenue Pierre-Marzin
Lannion 22307
FRANCE
EMail: jeanlouis.leroux@orange-ftgroup.com

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru