Internet Engineering Task Force (IETF)
Request for Comments: 6550
Category: Standards Track
ISSN: 2070-1721

T. Winter, Ed.

P. Thubert, Ed.
Cisco Systems

A. Brandt
Sigma Designs

J. Hui
Arch Rock Corporation

R. Kelsey
Ember Corporation

P. Levis
Stanford University

K. Pister
Dust Networks

R. Struik
Struik Security Consultancy

JP. Vasseur
Cisco Systems

R. Alexander
Cooper Power Systems

March 2012

RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks

RPL - протокол маршрутизации IPv6 для сетей с недостаточным электропитанием и высокими потерями

Аннотация

Сети LLN[1] являются классом сетей, где маршрутизаторы и соединения между ними имеют определённые ограничения. Маршрутизаторы LLN обычно имеют ограниченную производительность, память и питание (батареи). Для соединений характерны высокие потери, малая скорость и нестабильность. При этом сети LLN могут включать от десятков до тысяч маршрутизаторов. Поддерживаемые потоки трафика включают соединения «точка-точка» (между устройствами внутри LLN), «один со многими» (point-to-multipoint - от центральной точки управления к подмножеству устройств внутри LLN) и «многие с одним» (multipoint-to-point - от устройств внутри LLN к центральной точке управления). Этот документ задаёт протокол маршрутизации IPv6 для сетей LLN (RPL[2]), обеспечивающий механизм поддержки многоточечных соединений между устройствами LLN и центральной точкой управления в обоих направлениях. Доступна также поддержка соединений «точка-точка».

Статус документа

Этот документ содержит проект стандарта Internet (Internet Standards Track).

Документ является результатом работы IETF[3] и представляет собой согласованное мнение сообщества IETF. Документ был вынесен на публичное рассмотрение и одобрен для публикации IESG[4]. Дополнительная информация о стандартах Internet доступна в разделе 2 RFC 5741.

Информацию о текущем статусе документа, обнаруженных ошибках и способах обратной связи можно получить, воспользовавшись ссылкой http://www.rfc-editor.org/info/rfc6550.

[1] Low-Power and Lossy Network - сеть с низким электропитанием и потерями.
[2] Routing Protocol for Low-Power and Lossy Networks.
[3] Internet Engineering Task Force - комиссия по решению инженерных задач Internet.
[4] Internet Engineering Steering Group - комиссия по инженерным разработкам Internet.

Оглавление

1. Введение

Сети LLN в основном состоят из узлов с ограничениями (производительность обработки, память, а иногда батарейное питание). Эти маршрутизаторы соединены каналами с потерями, которые обычно имеют малую скорость и часто нестабильны. Другой характеристикой таких сетей является картина трафика - соединения в сетях часто организуются по схеме «один со многими» (point-to-multipoint) или «множество с одним» (multipoint-to-point). Такие сети могут включать тысячи узлов. Указанные характеристики предъявляют серьёзные требования к маршрутизации, которые рабочая группа IETF ROLL определила для протокола маршрутизации в сетях LLN в документах [RFC5867], [RFC5826], [RFC5673], [RFC5548]. Этот документ задаёт протокол маршрутизации RPL[1]. Следует подчеркнуть, что протокол RPL, разработанный в соответствии с указанными документами, может иметь более широкое применение.

1.1. Принципы работы

Протокол RPL был разработан с целью выполнения требований [RFC5867], [RFC5826], [RFC5673], [RFC5548].

В сети может одновременно работать несколько экземпляров RPL, каждый из которых может иметь свои ограничения или критерии производительности. При этом требования разных экземпляров могут противостоять друг другу.

[1]Произносится как ripple. См. также https://www.rfc-editor.org/errata/eid4219. *Прим. перев.*

В разных приложениях LLN протокол RPL отделяет обработку и пересылку пакетов от целей оптимальной маршрутизации. Примерами таких целей служат минимизация энергопотребления и задержки, а также соблюдение ограничений. В этом документе описан режим работы RPL, а сопровождающие документы задают предметные функции маршрутизации (Objective Function). Реализация RPL для поддержки конкретных приложений LLN будет включать нужные приложению предметные функции.

Для работы RPL нужны двухсторонние каналы. В некоторых вариантах LLN эти каналы могут быть асимметричными. Нужна проверка доступности маршрутизатора до того, как его можно будет применять в качестве родителя. В RPL предполагается включение внешнего механизма в фазе выбора родителя для проверки свойств канала и доступности соседа. Детектирование недоступности соседей (Neighbor Unreachability Detection или NUD) обеспечивает такой механизм, но возможны и другие варианты, включая обнаружение двухсторонней пересылки (Bidirectional Forwarding Detection или BFD) [RFC5881], а также рекомендации нижележащих уровней через триггеры L2, подобные описанным в [RFC5184]. В общем случае механизм, реагирующий на трафик, является более предпочтительным за счёт минимизации издержек на отслеживание неиспользуемых каналов.

RPL также предполагает внешний механизм для доступа к управляющей информации и её доставки (называется RPL Packet Information - информация пакета RPL) в пакетах данных. Определение RPL Packet Information дано в параграфе 11.2 и эти сведения позволяют связать пакет данных с RPL Instance и проверить состояние маршрутизации RPL. Примером такого механизма является опция RPL [RFC6553]. Механизм нужен для всех пакетов, за исключением случая использования строгой маршрутизации, заданной отправителем (т. е. пакетов нисходящего направления в режиме Non-Storing, как описано в разделе 9), когда сама природа маршрутизации предотвращает петли и смягчает необходимость передачи RPL Packet Information. В сопровождающих спецификациях могут быть заданы дополнительные способы передачи RPL Packet Information в пакетах IPv6, а также расширен состав RPL Packet Information для поддержки дополнительных функций.

RPL поддерживает механизм распространения информации по формируемой динамически топологии сети. Это распространение обеспечивает минимальную конфигурацию узлов, позволяющую им работать по большей части автономно. Механизм использует Trickle [RFC6206] для оптимизации распространения, как описано в параграфе 8.3.

В некоторых вариантах применения RPL собирает топологии маршрутизаторов, владеющих независимыми префиксами, которые могут (не обязательно) быть агрегируемыми в зависимости от их происхождения. Принадлежащий маршрутизатору префикс анонсируется как относящийся к каналу (on-link).

RPL также предоставляет возможность связать подсеть с общим префиксом и маршрутизировать внутри этой подсети. Отправитель (источник) может вводить информацию о подсети для распространения протоколом RPL и будет полномочным для данной подсети. Поскольку многие каналы LLN являются непереходными, общий префикс распространяемый RPL в подсети, недопустимо анонсировать как относящийся к каналу.

В частности, RPL может распространять информацию IPv6 ND[1], такую как PIO[2] [RFC4861] и RIO[3] [RFC4191]. Данные ND, распространяемые RPL, сохраняют свою исходную семантику при передаче от маршрутизатора к хосту с ограниченными расширениями при передаче между маршрутизаторами, хотя их не следует путать с маршрутными анонсами и никогда не следует распространять напрямую в другие протоколы маршрутизации. Узлы RPL часто объединяют в себе характеристики поведения хоста и маршрутизатора. Как хост, узел обрабатывает опции в соответствии с [RFC4191], [RFC4861], [RFC4862] и [RFC6275], а как маршрутизатор, может анонсировать информацию из опций, нужную для конкретного канала, например в сообщениях RA[4]. Описание этого выходит за рамки документа.

Набор сопровождающих эту спецификацию документов содержит рекомендации в форме заявлений о применимости, определяющих вопросы функционирования для автоматизации зданий (Building Automation), домашней автоматизации (Home Automation), а также промышленных и городских приложений.

1.2. Взаимодействие с канальным уровнем

В соответствии с многоуровневой архитектурой IP протокол RPL не полагается на конкретные свойства той или иной технологии канального уровня. Протокол RPL рассчитан на работу с разными протоколами канального уровня, включая протоколы с ограничениями и потерями, а также предназначенные для работы с сильно ограниченными по возможностям маршрутизаторами и хостами, такими как беспроводные устройства с ограниченным электропитанием или устройства PLC[5].

Разработчики могут найти в [RFC3819] полезные ссылки для создания интерфейсов канального уровня между RPL и конкретными технологиями канального уровня.

2. Терминология

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

В документе применяется терминология [ROLL-TERMS], а также перечисленные ниже термины.

DAG (Directed Acyclic Graph) - направленный ациклический граф
: Направленный граф, все ребра которого ориентированы (имеют направление) так, что не возникает петель. Ребра направлены к корневым узлам и завершаются в них.

DAG root - корень DAG
: Узел в DAG, не имеющий выходных рёбер. Поскольку DAG по определению является ациклическим графом, каждый DAG должен иметь по крайней мере один корень и все пути завершаются в DAG root.

Destination-Oriented DAG (DODAG) - ориентированный на адресата граф DAG
: DAG с корнем у одного адресата, т. е. в одном корне DAG (DODAG root) без выходных рёбер.

---

[1] Neighbor Discovery - обнаружение соседей.
[2] Prefix Information Option - опция сведений о префиксе.
[3] Route Information Option - опция сведений о маршруте.
[4] ND Router Advertisement - анонс маршрутизатора.
[5] Power Line Communication - коммуникации по сети электропитания.

DODAG root - корень DODAG

Корнем DODAG является корень DAG для графа DODAG. Корень DODAG может служить граничным маршрутизатором для DODAG и агрегировать маршруты в DODAG или распространять маршруты DODAG в другие протоколы маршрутизации.

Virtual DODAG root - виртуальный корень DODAG

Виртуальный корень DODAG создаётся в результате скоординированной работы двух и более маршрутизаторов RPL (например, 6LBR[1]) с синхронизацией состояний DODAG, как единого корня DODAG (со множеством интерфейсов) в сети LLN. Координация скорей всего происходит между устройствами по надёжному транзитному каналу, а детали этой схемы выходят за рамки спецификации (определены в сопроводительных документах).

Up - вверх

Указывает направление от узлов-листьев к корню DODAG по рёбрам дерева DODAG. Это общепринято при рассмотрении графов, где более удалённые от корня вершины графа считаются более глубокими или низкими.

Down - вниз

Указывает направление от корня DODAG к узлам-листьям по рёбрам DODAG.

Rank - ранг

Ранг узла определяет его позицию относительно корня DODAG. Ранг строго возрастает в направлении Down и уменьшается в направлении Up. Точное значение Rank зависит от применяемой предметной функции DAG (OF). Ранг может просто соответствовать топологической дистанции, быть функцией метрики каналов, а также учитывать иные свойства (например, ограничения).

Objective Function (OF) - предметная функция

OF определяет использование метрики маршрутов, целей оптимизации и связанных функций для расчёта ранга (Rank). Кроме того, OF задаёт способ выбора родителей DODAG и, следовательно, формирование DODAG.

Objective Code Point (OCP) - код предметной функции

Идентификатор, указывающий предметную функцию OF, используемую DODAG.

RPLInstanceID - идентификатор экземпляра RPL

Уникальный в масштабе сети идентификатор экземпляра протокола. Графы DODAG с общим RPLInstanceID используют одну предметную функцию OF.

RPL Instance - экземпляр RPL

RPL Instance - это один или множество графов DODAG с общим RPLInstanceID. В большинстве случаев узел RPL может относиться к одному графу DODAG экземпляра RPL. Каждый экземпляр RPL работает независимо от других RPL Instance. Данный документ описывает операции в рамках одного RPL Instance.

DODAGID

Идентификатор корня DODAG, который **должен** быть достижимым адресом IPv6 для корневого узла[2]. Граф DODAG **должен** быть уникальным в рамках RPL Instance в сети LLN. Для указания DODAG служит пара (RPLInstanceID, DODAGID).

DODAG Version - версия DODAG

Конкретная итерация (версия) DODAG с данным DODAGID.

DODAGVersionNumber - номер версии DODAG

Счётчик, инкрементируемый корнем для формирования новой версии графа DODAG, однозначно указываемой триплетом (RPLInstanceID, DODAGID, DODAGVersionNumber).

Goal - цель

Зависящая от приложения цель, определённая вне RPL. Любой узел, являющийся корнем DODAG, должен знать об этой цели для решения вопроса о её достижимости. Типичной целью является создание графа DODAG в соответствии с конкретной OF и сохранение связности с набором хостов (например, для использования OF, минимизирующей метрику и подключённой к узлу с базой данных для хранения собранной информации).

Grounded - приземленный (граф)

Граф DODAG считается «приземленным» (grounded), если корень DODAG может удовлетворять цели (Goal).

Floating - плавайющий (граф)

Не приземлённый граф DODAG является «плавающим» (floating) и для него не предполагается наличие свойств, требуемых для достижения цели. Однако он может обеспечивать связность с другими узлами внутри DODAG.

DODAG parent - родитель DODAG

Родителем узла в DODAG является один из смежных узлов на пути к корню DODAG. Ранг родителя DODAG всегда меньше ранга данного узла (3.5.1. Сравнение ранга).

Sub-DODAG - субграф DODAG

Субграфом DODAG для узла является множество других узлов, чьи пути к корню DODAG проходят через данный узел. Ранг узлов в суб-DODAG больше значения Rank данного узла (3.5.1. Сравнение ранга).

Local DODAG

Локальные графы DODAG содержат единственный корневой узел и позволяют ему выделять RPL Instance, указываемый локальным RPLInstanceID, и управлять этим экземпляром без координации с другими узлами. Обычно это делается для оптимизации маршрутов к получателю внутри LLN (5. Экземпляр RPL).

Global DODAG

Global DODAG использует глобальный идентификатор RPLInstanceID, который может координироваться с другими узлами (5. Экземпляр RPL).

DIO

DODAG Information Object (6.3. Информационный объект DODAG)

DAO

Destination Advertisement Object (6.4. Сообщение DAO)

DIS

DODAG Information Solicitation (6.2. Сообщение DIS)

CC

Consistency Check (6.6. Проверка согласованности)

При формировании сети устройства LLN могут выступать в качестве хостов и маршрутизаторов, в отличие от традиционных сетей IP. В этом документе термин «хост» относится к устройству LLN, способному генерировать трафик, но не пересылающему трафик RPL, а маршрутизатором считается устройство LLN, способное пересылать и генерировать трафик RPL. Термин «узел» обозначает любое устройство RPL - хост или маршрутизатор.

---

[1] 6LoWPAN Border Router - граничный маршрутизатор 6LoWPAN (IPv6 Low-Power Wireless Personal Area Network).

[2] См. https://www.rfc-editor.org/errata/eid4654. *Прим. перев.*

3. Обзор протокола

В этом разделе описывается протокол RPL в стиле [RFC4101]. Детали протокола описаны в других разделах.

3.1. Топология

Здесь описаны базовые варианты топологии RPL и правила формирования топологии, т. е. графа DODAG.

3.1.1. Создание топологии

Сети LLN, такие как радиосети (Radio Network), обычно не имеют предопределённой топологии, обеспечиваемой, например, кабельными соединениями «точка-точка». Поэтому протоколу RPL нужно обнаруживать каналы, а затем экономно выбирать партнёров.

Во многих случаях в результате лишь частичного перекрытия областей L2 протокол RPL формирует непереходную топологию или сеть NBMA[1], на основе которой рассчитываются маршруты.

Маршруты RPL оптимизируются для доставки трафика к одному или нескольким корневым узлам, выступающим приёмниками трафика, а также трафика от этих узлов. В результате RPL организует топологию в форме направленного ациклического графа (DAG) с одним или несколькими ориентированными на получателя DAG (DODAG), по одному графу DODAG на адресата. Если DAG имеет несколько корней, предполагается их объединение той или иной общей магистралью, например, транзитным каналом.

3.1.2. Идентификаторы RPL

В RPL используется 4 значения для идентификации и поддержки топологии.

RPLInstanceID

RPLInstanceID указывает один или несколько графов DODAG. Сеть может иметь множество RPLInstanceID, каждый из которых определяет независимый набор DODAG, который можно оптимизировать для разных предметных функций (OF) и/или приложений. Набор DODAG, указанный RPLInstanceID, называется RPL Instance. Все DODAG одного экземпляра RPL используют одну функцию OF.

DODAGID

Областью действия DODAGID является экземпляр RPL. Комбинация RPLInstanceID и DODAGID однозначно указывает граф DODAG в сети. RPL Instance может включать множество DODAG с уникальными DODAGID.

DODAGVersionNumber

Областью действия DODAGVersionNumber является DODAG. Графы DODAG иногда реконструируются из корня DODAG путём инкрементирования DODAGVersionNumber. Комбинация RPLInstanceID, DODAGID и DODAGVersionNumber однозначно указывает DODAG Version.

Rank

Областью действия Rank является DODAG Version. Ранг задаёт частичное упорядочение в DODAG Version, задавая позиции отдельных узлов относительно корня DODAG.

3.1.3. Экземпляры RPL, графы DODAG и версии DODAG

Экземпляр RPL содержит один или несколько корней DODAG. RPL Instance может обеспечивать маршруты к некоторым целевым префиксам, достижимым через корни DODAG или по другим путям внутри графа DODAG. Корни могут работать независимо или координироваться через сеть, которая не обязательно является LLN.

Ниже перечислены возможные варианты RPL Instance.

- Один граф DODAG с одним корнем.

  Например, граф DODAG, оптимизированный для минимизации задержки, с корнем в виде централизованного контроллера освещения в системе домашней автоматизации.

- Множество некоординированных DODAG с независимыми корнями (разные DODAGID).

  Например, множество точек сбора данных в городском приложении сбора информации, которые не имеют подходящей связи для координирования работы или используют разные DODAG для динамического и автономного разделения сети.

- Один граф DODAG с виртуальным корнем, координирующим приёмники LLN (с одним DODAGID) через опорную сеть.

  Например, множество граничных маршрутизаторов, соединённых надёжным каналом для поддержки приложений 6LoWPAN и способных служить эквивалентными интерфейсами к приёмнику в одном DODAG.

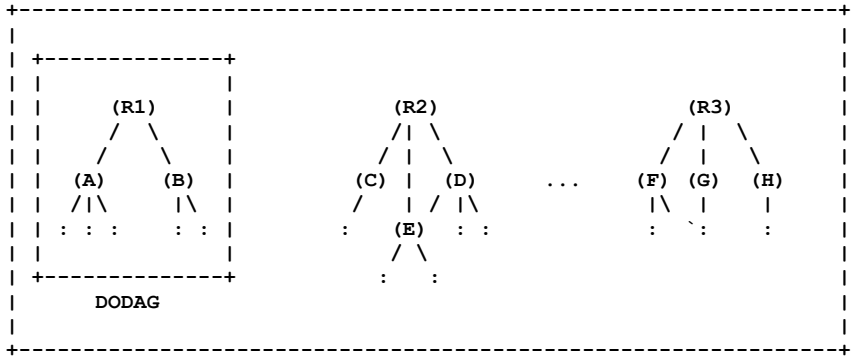


Рисунок 1. Экземпляр RPL.

[1]Non-Broadcast Multi-Access - множественный доступ без широковещания.

- Комбинация перечисленных ниже вариантов, подходящая для определённого приложения.

Каждый пакет RPL связывается с конкретным RPLInstanceID (см. параграф 11.2) и, следовательно, RPL Instance (раздел 5). Предоставление или автоматическое обнаружение сопоставлений RPLInstanceID с типом или службой трафика приложений выходит за рамки спецификации (определяется в сопровождающих документах).

На рисунке 1 показан пример экземпляра RPL, содержащего три DODAG с корнями R1, R2, R3, каждый из которых анонсирует одно значение RPLInstanceID. Линии указывают соединения между родителями и потомками.

На рисунке 2 показано, как инкрементирование DODAGVersionNumber ведёт к появлению новой версии DODAG и новой топологии DODAG. Отметим, что новое значение DODAG Version не всегда предполагает другую топологию DODAG. Для восприятия некоторых топологических изменении требуется новая версия DODAG, как описано ниже.

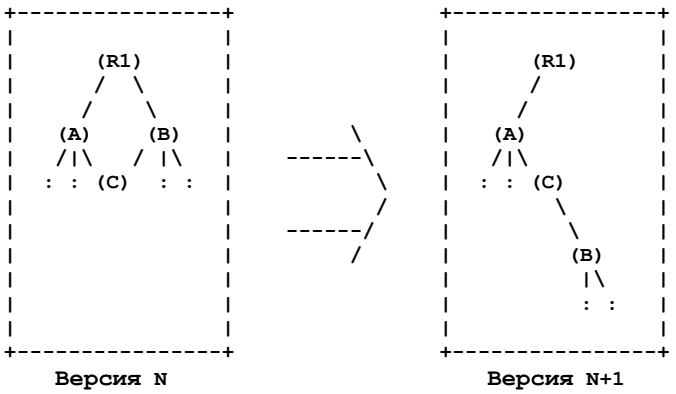


Рисунок 2. Версия DODAG.

В последующих примерах для простоты приведены древовидные структуры, хотя в реальных DODAG каждый узел может иметь несколько родителей, если связность позволяет это.

3.2. Восходящие маршруты и создание DODAG

RPL предоставляет восходящие (Up) маршруты в направлении корней DODAG, формирующие оптимизированный граф DODAG в соответствии с предметной функцией (OF). Узлы RPL поддерживают графы DODAG с помощью сообщений DIO (6.3. Информационный объект DODAG).

3.2.1. Предметная функция

Предметная функция (Objective Function или OF) определяет для узлов RPL способ выбора и оптимизации маршрутов в рамках экземпляра RPL. OF указывается целевым кодом OCP в опции DIO Configuration и определяет, как узлы транслируют метрику и ограничения, определённые в [RFC6551], в ранг (Rank), аппроксимирующий удалённость от корня DODAG. OF также определяет способ выбора родителей. Дополнительные сведения о предметных функциях приведены в разделе 14, а также в [RFC6551], [RFC6552] и сопроводительных спецификациях.

3.2.2. Восстановление DODAG

Корень DODAG запускает глобальную операцию восстановления, увеличивая DODAGVersionNumber, что ведёт к созданию новой версии DODAG. Узлы в новой DODAG Version могут выбирать новую позицию, ранг которой не привязан к значению Rank в прежней DODAG Version.

RPL также поддерживает механизмы, которые могут применяться для локального восстановления в рамках DODAG Version. Сообщение DIO задаёт параметры, настраиваемые и контролируемые политикой в корне DODAG.

3.2.3. Защита

RPL поддерживает целостность и конфиденциальность сообщений. Устройство протокола позволяет использовать механизмы защиты канального уровня, когда они доступны и пригодны, а в отсутствие таких механизмов RPL обеспечивает свои механизмы защиты. В RPL поддерживается три базовых механизма защиты.

При работе без защиты (unsecured) управляющие сообщения RPL передаются без использования дополнительных механизмов защиты, однако этот режим не оставляет сеть RPL без защиты и позволяет использовать внешние примитивы защиты (например, средства канального уровня) для выполнения требований приложения.

В режиме preinstalled узлы, подключающиеся к экземпляру RPL Instance, имеют заранее установленные ключи, позволяющие генерировать и обрабатывать защищённые сообщения RPL.

Третий режим называется authenticated и использует заранее установленные ключи, как в режиме preinstalled, но узлы могут подключаться к RPL Instance лишь в качестве листьев. Для подключения аутентифицированного экземпляра RPL Instance как маршрутизатора требуется получение ключа от удостоверяющего центра, но спецификация не задаёт процесс получения ключа. Отметим, что сама спецификация не задаёт деталей реализации RPL для защищённой работы в режиме authenticated. Для безопасной работы реализации RPL в аутентифицированном режиме нужны дополнительные спецификации, задающие детали запроса и получения аутентификационного материала (ключи, сертификаты) и его источники (см. 10.3. Установка ключей).

3.2.4. «Приземлённые» и «плавающие» DODAG

Графы DODAG могут быть приземлёнными или плавающими, роль анонсирует корень DODAG. Приземлённый граф DODAG предлагает связность с хостами, требуемую для выполнения заданной приложением задачи. От плавающего DODAG не ожидается достижение цели и в большинстве случаев он обеспечивает лишь маршруты к узлам внутри DODAG. Плавающие DODAG могут применяться, например, для сохранения связности в процессе восстановления.

3.2.5. Локальные DODAG

Узлы RPL могут оптимизировать маршруты к адресатам внутри LLN, формируя Local DODAG, где корнем DODAG является желаемый получатель. В отличие от глобальных DAG, которые могут содержать множество DODAG, локальные DAG имеют лишь один граф DODAG и, следовательно, - один корень DODAG. Локальные DODAG создаются по запросу.

3.2.6. Административные предпочтения

Реализация или развёртывание могут задавать предпочтительность отдельных корней DODAG административными средствами. Административные предпочтения обеспечивают возможность контролировать трафик и формирование DODAG для более эффективной поддержки потребностей приложений.

3.2.7. Проверка пути данных и обнаружение петель

Слабое электропитание и потери в сетях LLN вынуждают применять в RPL детектирование петель по запросу с использованием пакетов данных. Поскольку трафик данных может быть нечастым, поддержка постоянного соответствия топологии маршрутизации и топологии физических соединений может потреблять излишнюю энергию. В типичных сетях LLN наблюдаются изменения физической связности, которые являются временными и безопасными для трафика, но могут требовать затрат энергии на их отслеживание плоскостью управления. Такие изменения не следует обрабатывать в RPL, пока нет реальных данных для передачи. Этот аспект устройства RPL основан на опыте применения широко распространённых протоколов LLN, а также обширных экспериментальных данных.

Опция RPL Packet Information в пакетах данных включает ранг передающего узла. Несоответствие маршрута для пакета (Upward или Downward) и значений Rank для двух узлов говорит о возможной петле. При получении такого пакета узел инициирует локальную операцию восстановления. Например, если узел получает пакет, помеченный для передачи в восходящем направлении, а запись для пакета указывает, что передающий узел имеет меньший ранг, нежели принимающий, тогда принимающий узел может сделать вывод о том, что пакет не нужно передавать «наверх» (Upward) и граф DODAG не согласован.

3.2.8. Работа распределенного алгоритма

Высокоуровневый алгоритм построения графа DODAG приведён ниже.

- Некоторые узлы настраиваются как корни DODAG с соответствующей конфигурацией DODAG.
- Узлы анонсируют своё присутствие, принадлежность к DODAG, стоимость маршрутизации и метрику, передавая групповые для локального канала сообщения DIO всем узлам RPL.
- Узлы прослушивают сообщения DIO и используют сведения из них для присоединения к новому графу DODAG (выбор родителей DODAG) или поддержки имеющегося DODAG в соответствии с заданной предметной функцией (OF) и рангом их соседей.
- Узлы предоставляют записи таблицы маршрутов для адресатов из сообщения DIO через своих родителей DODAG в DODAG Version. Узел, решивший присоединиться к DODAG, может предоставить одного или нескольких родителей DODAG в качестве следующего интервала для принятого по умолчанию маршрута и ряда других внешних маршрутов для связанного с ним экземпляра.

3.3. Нисходящие маршруты и анонсирование адресатов

RPL использует сообщения DAO для организации нисходящих (Downward) маршрутов. Сообщения DAO являются необязательным свойством приложений, которым нужен трафик point-to-multipoint (P2MP) или point-to-point (P2P). RPL поддерживает для трафика Downward два режима - Storing (с поддержкой состояний) и Non-Storing (маршрутизация, заданная отправителем - source route), описанные в разделе 9, и каждый RPL Instance работает в одном из этих режимов. В обоих случаях пакеты P2P передаются в восходящем направлении к корню DODAG, затем в нисходящем к конечному получателю (если получатель не находится на маршруте Upward). В режиме Non-Storing пакет будет проходить весь путь до корня DODAG и только потом пойдёт вниз (Down). В режиме Storing пакет может быть развернут вниз общим предком отправителя и получателя, не дойдя до корня DODAG.

На момент создания этой спецификации от реализаций не ожидалась поддержка обоих режимов Storing и Non-Storing. Предполагалось, что большинство реализаций будут работать без маршрутов Downward или лишь в одном из режимов Non-Storing и Storing. Другие режимы, такие как комбинация Storing и Non-Storing, выходят за рамки спецификации и могут быть описаны в сопровождающих документах.

Данная спецификация описывает базовый режим работы для поддержки трафика P2P. Дополнительные спецификации могут предлагать разные режимы оптимизации трафика P2P.

3.4. Обнаружение маршрутов локальных DODAG

Сеть RPL может также поддерживать по запросу обнаружение DODAG для конкретных адресатов в LLN. Такие Local DODAG ведут себя несколько иначе, чем Global DODAG, однако они однозначно определяются парой DODAGID и RPLInstanceID. Идентификатор RPLInstanceID указывает, является ли DODAG локальным графом DODAG.

3.5. Свойства Rank

Ранг узла является скалярным представлением расположения этого узла в DODAG Version. Ранг применяется для обнаружения и предотвращения петель, поэтому от него требуются определённые свойства. Точный расчёт ранга выполняет предметная функция OF, однако от Rank требуется наличие базовых свойств, независимых от OF. В частности, значение Rank для узлов должно монотонно снижаться по мере перемещения по DODAG Version в направлении адресата DODAG. В этом отношении ранг можно считать скалярным представлением местоположения или радиусом узла внутри DODAG Version.

Детали расчёта Rank предметной функцией OF выходят за рамки спецификации. Расчёт может зависеть, например, от родителей, метрики узлов, а также их конфигурации и правил (14. Рекомендации для предметных функций).

Rank не является стоимостью пути, хотя значение может выводиться из метрики пути и зависеть от неё. Rank имеет свойства, которые могут не относиться к метрике.

Тип
Rank является абстрактным числовым значением.

Назначение
Rank указывает позицию в DODAG Version относительно соседей и не обязательно служит надёжной индикацией или подходящим выражением дистанции от корня или стоимости пути к нему.

Стабильность
Стабильность Rank определяет стабильность топологии маршрутов. **Рекомендуется** применять то или иное демпфирование и фильтрацию для сохранения стабильной топологии, поэтому не требуется менять Rank так же быстро, как метрику каналов или узлов. Новая версия DODAG служит хорошей возможностью согласовать расхождения, которые могли возникнуть между метрикой и Rank в DODAG Version.

Свойства
Значение Rank инкрементируется строго монотонно и может служить для проверки движения к корню или от него. Метрика, подобно пропускной способности или вариациям задержки, может не иметь такого свойства.

Абстракция
Rank не имеет физических единиц, а имеет скорее диапазон инкрементирования на интервал пересылки (hop), где каждое приращение определяется целевой функцией.

Значение Rank передаётся в процедуру выбора родителей DODAG в соответствии со стратегией предотвращения петель в RPL. Когда родитель добавлен и значение Rank для узла в DODAG анонсировано, дальнейший выбор узлом родителей DODAG и перемещение внутри DODAG ограничиваются в пользу предотвращения петель.

3.5.1. Сравнение ранга

Ранг можно рассматривать как число с фиксированной запятой, положение которой определяется значением MinHopRankIncrease, задающим максимальное различие рангов между узлом и любым из его родителей DODAG. Значение предоставляет корень DODAG, обеспечивая компромисс между точностью указания стоимости интервала пересылки и максимальным числом таких интервалов в сети. Например, большое значение MinHopRankIncrease позволяет точнее учесть влияние интервала пересылки на ранг, но снижает возможное число интервалов.

Когда предметная функция OF вычисляет ранг, она работает с полным (16 битов) значением Rank. При сравнении рангов, например, для определения родительских отношений или обнаружения петель, применяется лишь целая часть Rank, которую определяет макрос DAGRank(), где floor(x) возвращает наибольшее целое число, не превышающее x.

```
DAGRank(rank) = floor(rank/MinHopRankIncrease)
```

Например, если 16-битовое значение Rank задаёт десятичное число 27, а MinHopRankIncrease имеет десятичное значение 16, DAGRank(27) = floor(1.6875) = 1. Целая часть Rank будет иметь значение 1, а дробная - 11/16.

В соответствии с соглашениями этого документа использование макроса DAGRank(node) можно интерпретировать как DAGRank(node.rank), где node.rank указывает значение Rank, поддерживаемое узлом.

Узел A имеет Rank меньше ранга узла B, если DAGRank(A) < DAGRank(B), ранг узлов одинаков, если DAGRank(A) = DAGRank(B) и ранг узла A больше ранга узла B, если DAGRank(A) > DAGRank(B).

3.5.2. Отношения между рангами

При расчёте рангов для соседних узлов M и N в LLN поддерживаются указанные ниже свойства.

DAGRank(M) < DAGRank(N)
Узел M ближе к корню DODAG, нежели узел N. Узел M может быть родителем DODAG для узла N без риска возникновения петли. Кроме того, все родители N в наборе родителей DODAG должны иметь ранг меньше DAGRank(N). Иными словами, ранг, представленный узлом Node N, **должен** быть больше ранга, представленного любым из его родителей.

DAGRank(M) = DAGRank(N)
Позиции узлов M и N относительно корня DODAG похожи или идентичны. Маршрутизация через узел с тем же Rank может создавать петлю (если этот узел выберет маршрут с таким же Rank).

DAGRank(M) > DAGRank(N)
Узел M расположен дальше от корня DODAG, чем узел N. Кроме того, узел M может фактически входить в суб-DODAG узла N. Если N выберет M в качестве родителя DODAG, возникает риск создания петли.

Например, значение Rank можно рассчитать так, чтобы точно отслеживать ETX[1], когда минимизируемой функцией OF метрикой является ETX, задержка или иной параметр в зависимости от функции OF, применяемой в DODAG.

3.6. Метрика маршрутов и ограничения в RPL

Метрика маршрутов используется протоколами маршрутизации для расчёта кратчайшего пути. Протоколы внутренней маршрутизации (Interior Gateway Protocol или IGP), такие как IS-IS [RFC5120] и OSPF [RFC4915], используют статическую метрику каналов. Эта метрика может просто отражать пропускную способность или полиномиальную функцию нескольких параметров, определяющих характеристики каналов. Некоторые протоколы маршрутизации поддерживают не одну метрику, но в подавляющем большинстве случаев для каждой (суб)топологии применяется одна метрика. Реже может использоваться вторая метрика для выбора при наличии нескольких равноценных путей (Equal Cost Multiple Path или ECMP). Оптимизацию нескольких метрик называют проблемой NP-complete и она иногда поддерживается централизованными машинами расчёта путей.

Для LLN нужна одновременная поддержка статической и динамической метрики, а также метрики каналов и узлов. В случае RPL практически невозможно определить одну метрику (даже составную) для всех случаев. Кроме того, RPL поддерживает маршрутизацию на основе ограничений, которые могут применяться для каналов и узлов. Каналы и узлы, не соответствующие требуемым ограничениям, исключаются из числа кандидатов при выборе кратчайшего пути.

[1]Expected transmission count - ожидаемое число передач (беспристрастная метрика LLN, определенная в [RFC6551]).

Предметная функция OF задаёт цели, используемые при расчёте пути (с ограничениями). Кроме того, узлы настраиваются для поддержки набора метрик и ограничений, а также выбора родителей в DODAG в соответствии с метрикой и ограничениями, анонсируемыми в сообщениях DIO. Метрики восходящих и нисходящих маршрутов могут объединяться или анонсироваться отдельно в зависимости от OF и метрики. При раздельном анонсировании наборы родителей DIO и DAO[1] могут отличаться, тем не менее, все они будут считаться родителями DODAG при расчёте Rank.

Функция OF отделена от метрики и ограничений, используемых RPL. С учётом диктуемых функцией OF правил ограничения выбора родителей DODAG, балансировки нагрузки, а также набора метрик и/или ограничений и т. п., выбор предпочтительного пути основывается на данных, передаваемых в опции контейнера DAG в сообщениях DIO. Набор поддерживаемых ограничений и метрик для узлов и каналов задан в [RFC6551]. Примерами могут служит путь с минимальной сквозной задержкой или путь, не проходящий через устройства с батарейным питанием.

3.7. Предотвращение петель

RPL пытается предотвратить возникновение петель при изменении базовой топологии и включает механизмы проверки путей на основе ранга для обнаружения петель (11. Пересылка пакетов, обнаружение и предотвращение петель). На практике это не гарантирует ни отсутствия петель, ни жёсткого ограничения времени сходимости, но может обнаруживать и устранять петлю, как только она будет использована. RPL применяет детектирование петель для продвижения пакетов в DODAG Version и запуске восстановления при необходимости.

3.7.1. «Жадность» и нестабильность

Узел считается «жадным», если он пытается переместиться вглубь (рост Rank) DODAG Version для расширения своего набора родителей или улучшения иной метрики. После присоединения узла к DODAG Version протокол RPL запрещает некоторые черты поведения (включая жадность) для предотвращения нестабильности DODAG Version.

Предположим, что узел желает получить и обработать сообщение DIO от узла в своём суб-DODAG, который обычно расположен глубже. В этом случае имеется вероятность наличия петли обратной связи, когда два или более узла продолжают попытки перемещения в DODAG Version, пытаясь оптимизировать друг друга. Такое поведение создаёт нестабильность. По этой причине RPL ограничивает случаи, когда узел может обрабатывать сообщения DIO от более глубоких узлов, некоторыми формами локального восстановления. Такой подход создаёт «горизонт событий», при котором на узел нет возможности влиять за пределами некого вклада в нестабильность, вносимого действиями узлов, которые могут находиться в его суб-DODAG.

3.7.1.1. Пример нестабильности при «жадном» выборе родителей

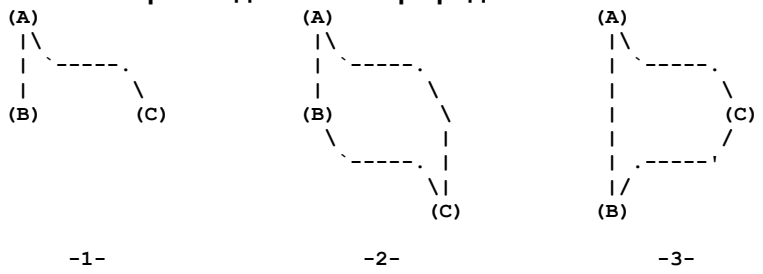


Рисунок 3. «Жадный» выбор родителей DODAG.

На рисунке 3 показаны три конфигурации DODAG, где имеется пригодный для использования канал между (B) и (C). Слева (1) узел (A) является родителем DODAG для (B) и (C), посредине (A) является родителем DODAG для (B) и (C), а (B) также служит родителем (C). Справа узел (A) служит родителем DODAG для (B) и (C), а (C) - родителем для (B).

Если узел RPL слишком жаден и пытается выполнить оптимизацию для дополнительных родителей сверх наиболее предпочтительных, может возникать нестабильность. В примере на рисунке 3-1 узлы (B) и (C) могут предпочесть (A) как родителя DODAG, но мы рассмотрим вариант «жадности» с попыткой выполнить оптимизацию для двух родителей.

- Пусть рисунок 3-1 задаёт начальные условия.
- Предположим, что (C) может выйти из DODAG и вернуться с меньшим Rank, делая узлы (A) и (B) родителями DODAG, как показано на рисунке 3-2. Узел (C) в этом случае глубже (A) и (B) и имеет 2 родителей DODAG.
- Предположим, что узел (B) жаден и хочет получать и обрабатывать сообщения DIO от (C), вопреки правилам RPL. Для этого (B) покидает DODAG и возвращается с меньшим Rank, принимая (A) и (C) как родителей DODAG. Сейчас узел (B) глубже (A) и (C) и имеет 2 родителей DODAG.
- Узел (C) тоже жаден и покидает граф, возвращаясь в него глубже, чтобы снова получить двух родителей и иметь меньший ранг, чем они.
- Затем узел (B) ещё раз выходит и возвращается глубже, снова получая двух родителей.
- Узел (C) также повторяет выход и более глубокое возвращение.
- Процедура зацикливается и DODAG будет осциллировать между состояниями рисунков 3-2 и 3-3, пока счётчик узлов не достигнет «бесконечности» и не начнёт новый цикл.
- Этот цикл можно разорвать с помощью механизмов RPL:
  - узлы (B) и (C) оставляют Rank достаточным для присоединения к предпочтительному родителю (A) и не пытаются присоединиться к более глубоким родителям (узлы не жадные);
  - узлы (B) и (C) не обрабатывают DIO от узлов глубже себя (эти узлы могут быть в их суб-DODAG).

Эти механизмы дополнительно рассматриваются в параграфе 8.2.2.4. Ранг и перемещение внутри версии DODAG.

[1]Родителем DAO является узел, которому передается индивидуальное сообщение DAO.

3.7.2. Петли DODAG

Петля DODAG может возникнуть при выходе устройства из DODAG с последующим подключением к устройству в прежнем суб-DODAG. В частности, это может происходить при пропуске сообщений DIO. Строгое применение DODAGVersionNumber может устранить этот тип петель, но они могут возникать при использовании некоторых механизмов локального восстановления.

Например, механизм локального ремонта, позволяющий узлу отсоединиться от DODAG, анонсировать Rank = INFINITE_RANK (для порчи своих маршрутов или информирования суб-DODAG), а затем снова соединиться с DODAG. В некоторых случаях узел может заново соединиться со своим прежним суб-DODAG, создавая петлю DODAG, поскольку порча маршрутов может завершиться отказом, если анонс INFINITE_RANK потеряется в среде LLN (в этом случае механизмы проверки пути на основе Rank в конечном итоге увидят и устранят петлю).

3.7.3. Петли DAO

Петля DAO может возникать при наличии у родителя маршрута, установленного в результате приёма и обработки сообщения DAO от дочернего узла, если тот позднее сбросил соответствующее состояние DAO. Такая петля возникает при пропуске No-Path (сообщение DAO, аннулирующее ранее анонсированный префикс, см. параграф 6.4.3. Опции DAO) и сохраняется до полной очистки состояния. RPL включает необязательный механизм подтверждения DAO, который может ослабить влияние потери одного сообщения DAO. В RPL имеются механизмы обнаружения петель, снижающие влияние петель DAO и вызывающие их устранение (см. параграф 11.2.2.3).

4. Поддерживаемые RPL потоки трафика

RPL поддерживает три базовых типа потоков трафика «множество с одним», «один со многими» и «точка-точка».

4.1. Трафик Multipoint-to-Point

Трафик MP2P доминирует во многих приложениях LLN ([RFC5867], [RFC5826], [RFC5673], [RFC5548]). Адресатами потоков MP2P являются узлы, имеющие определённое значение для приложений, например, обеспечивающие связность с Internet или ядром частной сети IP. RPL поддерживает трафик MP2P, обеспечивая доступ к получателям MP2P через корни DODAG.

4.2. Трафик Point-to-Multipoint

Трафик P2MP требуется для некоторых приложений LLN ([RFC5867], [RFC5826], [RFC5673], [RFC5548]). RPL поддерживает трафик P2MP с использованием механизма анонсирования получателей, обеспечивающего нисходящие маршруты к адресатам (префиксы, адреса, multicast-группы) и от них. Анонсирование получателей может обновлять таблицы маршрутизации при изменении базовой топологии DODAG.

4.3. Трафик Point-to-Point

RPL DODAG обеспечивают базовую структуру для трафика P2P. Для поддержки в сети RPL трафика P2P корень должен быть способен маршрутизировать пакеты к адресатам. Узлы сети также могут иметь таблицы маршрутов к адресатам. Пакеты перемещаются в направлении корня, пока не достигнут узла-предка, знающего путь к адресату. Как будет указано ниже, в наиболее ограниченном случае (узлы не могут хранить маршруты) общим предком может быть корень DODAG. В иных случаях это может быть узел, расположенный ближе к отправителю и получателю. RPL также поддерживает случай, когда получателем P2P является непосредственный сосед (one-hop).

RPL не задаёт и не исключает дополнительных механизмов для расчёта и организации более оптимальных путей маршрутизации произвольного трафика P2P.

5. Экземпляр RPL

В данной сети LLN может быть множество логически независимых экземпляров RPL. Узел RPL может относиться к нескольким RPL Instance, действуя в одних как маршрутизатор, в других - как лист. Этот документ описывает поведение одного экземпляра.

Имеется два типа RPL Instance - локальные и глобальные. RPL делит пространство RPLInstanceID между экземплярами Global и Local для согласованного и одностороннего выделения RPLInstanceID. Глобальные экземпляры RPL скоординированы, имеют не менее 2 DODAG и обычно применяются долго. Локальные экземпляры всегда имеют один граф DODAG, единственный корень которого владеет соответствующим DODAGID и выделяет локальное значение RPLInstanceID в одностороннем порядке. Локальные экземпляры RPL могут применяться, например, для создания DODAG в поддержку будущей маршрутизации по запросам. Режим работы локальных экземпляров RPL выходит за рамки спецификации и будет описан в сопровождающих документах.

Определение и представления экземпляров RPL выходит за рамки спецификации. Рекомендации могут зависеть от приложения и реализации, предполагается их разработка в будущих сопроводительных спецификациях. Ожидается, что эти операции будут такими, что пакеты, приходящие извне в сеть RPL, можно будет однозначно связать хотя бы с одним RPL Instance и безопасно маршрутизировать через любой экземпляр, соответствующий пакету.

Пакеты данных и управления в сети RPL помечаются для однозначного указания экземпляра RPL Instance, к которому относится пакет. Каждое управляющее сообщение RPL имеет поле RPLInstanceID. Некоторые управляющие сообщения RPL при обращении к локальному RPLInstanceID (см. ниже) могут включать также DODAGID.

Пакеты данных в сети RPL раскрывают RPLInstanceID как часть RPL Packet Information, требуемой RPL (11.2.2.1. Пересылка пакетов экземпляром). Для приходящих извне сети RPL пакетов входной маршрутизатор определяет RPLInstanceID и помещает этот идентификатор в результирующий пакет для сети RPL.

5.1. RPLInstanceID

Значение глобального RPLInstanceID **должно** быть уникальным в масштабе всей сети LLN. Механизмы выделения значений глобальных RPLInstanceID выходят за рамки этой спецификации. В сети может быть до 128 глобальных

экземпляров. Локальные экземпляры всегда применяются с DODAGID (задаётся явно или неявно) и поддерживается до 64 локальных экземпляров на DODAGID. Локальные экземпляры выделяются и поддерживаются узлом, владеющим DODAGID, без явной координации с другими узлами, как отмечено ниже.

Глобальное значение RPLInstanceID представляется в одноимённом поле, как показано на рисунке 4.

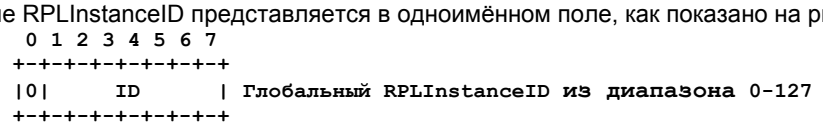


Рисунок 4. Формат поля RPLInstanceID для глобального экземпляра.

Локальные значения RPLInstanceID автоматически задаются узлом, владеющим DODAGID, и **должны** быть уникальны в рамках данного DODAGID. Идентификатором DODAGID, используемым для локального экземпляра RPLInstanceID, **должен** быть адрес IPv6 достижимого узла и этот адрес **должен** служить конечной точкой во всех коммуникациях, связанных с этим локальным экземпляром. Представление локального RPLInstanceID показано на рисунке 5.

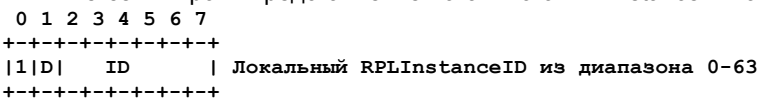


Рисунок 5. Формат поля RPLInstanceID для локального экземпляра.

Флаг D в локальных RPLInstanceID всегда имеет значение 0 в управляющих сообщениях RPL. В пакетах данных он применяется для указанию DODAGID как источника или получателя пакета - при установленном (1) флаге D адресом получателя IPv6 **должно** быть значение DODAGID, а при сброшенном флаге значение DODAGID **должно** быть адресом отправителя IPv6.

Рассмотрим, например, узел A, являющийся корнем DODAG для Local RPL Instance и имеющий локальное значение RPLInstanceID. По определению весь трафик, проходящий через этот локальный экземпляр RPL, будет исходить от узла A или завершаться на нем. В этом случае DODAGID будет доступным адресом IPv6 узла A. Весь трафик, будет содержать адрес узла A (т. е. DODAGID) в поле отправителя или получателя. Таким образом, локальное значение RPLInstanceID может указывать, что DODAGID является эквивалентом адреса отправителя или получателя путём установки или сброса флага D.

6. Сообщения ICMPv6 RPL Control

Этот документ определяет управляющие сообщение RPL как новое сообщение ICMPv6 [RFC4443]. Управляющие сообщения RPL указываются кодом, который определяет формат базового сообщения и включение в него опций. Областью действия большинства управляющих сообщений RPL является канал. Исключением являются лишь сообщения DAO и DAO-ACK в режиме Non-Storing, которые передаются по индивидуальному адресу через несколько интервалов пересылки (hop) и используют глобальные или уникальные в локальном масштабе адреса отправителей и получателей. Для прочих управляющих сообщений RPL адресом источника служит link-local, а адресом получателя - групповой адрес all-RPL-nodes или индивидуальный адрес link-local для получателя. Групповой адрес all-RPL-nodes является новым multicast-адресом и имеет значение ff02::1a.

В соответствии с [RFC4443] сообщение RPL Control состоит из заголовка ICMPv6, за которым следует тело сообщения, состоящее из базы и необязательного набора опций, как показано на рисунке 6.

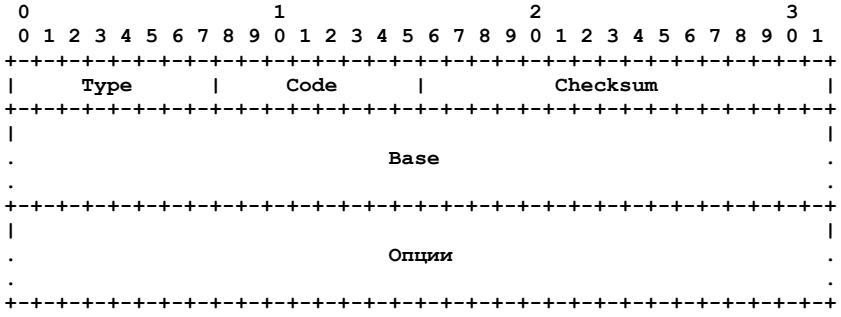


Рисунок 6. Сообщение RPL Control.

Сообщение RPL Control является информационным сообщением ICMPv6 с Type = 155. Поле Code указывает тип сообщения RPL Control. Определяемые эти документом типы перечислены ниже.

- 0x00: DODAG Information Solicitation (6.2. Сообщение DIS)
- 0x01: DODAG Information Object (6.3. Информационный объект DODAG)
- 0x02: Destination Advertisement Object (6.4. Сообщение DAO)
- 0x03: Destination Advertisement Object Acknowledgment (6.5. Подтверждение DAO)
- 0x80: Secure DODAG Information Solicitation (6.2.2. Secure DIS)
- 0x81: Secure DODAG Information Object (6.3.2. Secure DIO)
- 0x82: Secure Destination Advertisement Object (6.4.2. Secure DAO)
- 0x83: Secure Destination Advertisement Object Acknowledgment (6.5.2. Secure DAO-ACK)
- 0x8A: Consistency Check (6.6. Проверка согласованности)

Узел, получивший сообщение RPL Control с неизвестным значением Code, **должен** отбросить сообщение без обработки, **может** создать сигнал для системы управления, но передача ответных сообщений **недопустима**.

Контрольная сумма рассчитывается в соответствии с [RFC4443]. В поле устанавливается значение 0 для описанных ниже операций защиты RPL, а после установки всех полей сообщения RPL, включая защитные, рассчитывается контрольная сумма и помещается в поле Checksum.

Старший бит кода (0x80) показывает использование защиты для сообщения RPL. Формат сообщений RPL с защитой целостности и конфиденциальности показан на рисунке 7.

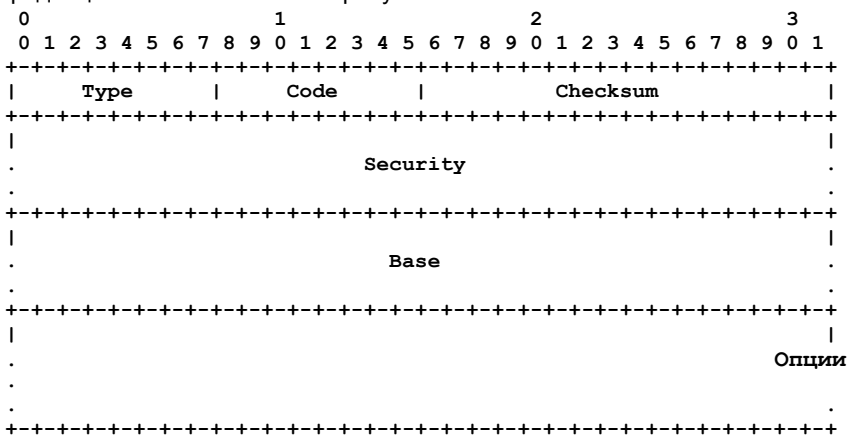


Рисунок 7. Управляющее сообщение Secure RPL.

Далее в этом разделе описаны базовые форматы определённых в настоящее время управляющих сообщений RPL и опций RPL Control Message.

6.1. Поля RPL Security

Каждое сообщение RPL имеет защищённый вариант, обеспечивающий контроль целостности и предотвращение повторного использования, а также возможность защиты конфиденциальности и предотвращение задержки. Поскольку защита охватывает базовое сообщение и опции, данные защиты размещаются в сообщении между полями Checksum и Base, как показано на рисунке 7.

Уровень защиты и используемые алгоритмы указываются в протокольном сообщении, показанном ниже.

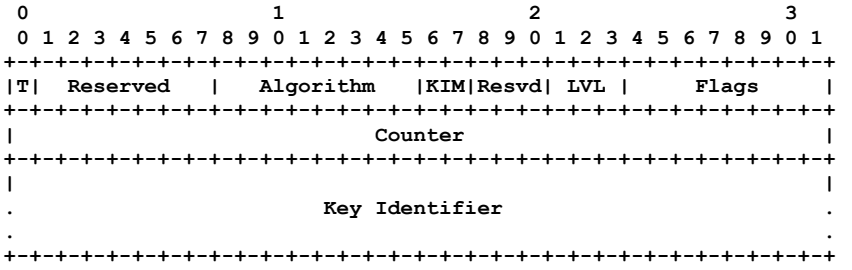


Рисунок 8. Раздел Security.

Коды аутентификации сообщений (MAC[1]) и подписи обеспечивают проверку подлинности всего сообщения ICMPv6 RPL, включая раздел Security со всеми полями, но с временной установкой в поле контрольной суммы ICMPv6 значения 0. Шифрование обеспечивает конфиденциальность защищённого сообщения RPL ICMPv6 с первого байта после раздела Security до последнего байта в пакете. Защитное преобразование даёт защищённое сообщение ICMPv6 RPL с включением криптографических полей (MAC, подпись и т. п.). Иными словами, способ встраивания криптографических полей (например, используемые Signature и/или Algorithm) определяет само защитное преобразование определяет. Раздел Security не включает явно эти криптографические поля. Подробное описание применения полей Security приведено в разделах 19 и 10.

Counter is Time (T)

Установка флага T говорит о том, что поле Counter содержит временную метку. При сброшенном флаге это поле содержит инкрементируемый счётчик. Подробное описание флага T и поля Counter дано в параграфе 10.5.

Reserved

7 резервных битов, которые **должны** сбрасываться (0) отправителем. Получатель **должен** игнорировать эти биты.

Security Algorithm (Algorithm)

Поле Security Algorithm задаёт схему шифрования, MAC и подписей, используемую в сети. Поддерживаемые значения приведены на рисунке 9. Подробное описание алгоритмов дано в параграфе 10.9.

Алгоритм	*Шифрование/MAC*	*Подпись*
0	CCM с AES-128	RSA с SHA-256
1-255	Не заданы	Не заданы

Рисунок 9. Кодирование алгоритмов защиты.

Key Identifier Mode (KIM)

2-битовое поле, указывающее способ задания ключа, применяемого для защиты пакета (явно или неявно), а также конкретное представление поля Key Identifier. Возможные значения поля KIM показаны ниже.

Режим	*KIM*	*Значение*	*Число октетов идентификатора*
0	00	Использование группового ключа, определяемого полем Key Index. Key Source 1 отсутствует, Key Index имеется.	
1	01	Использование ключа пары, определяемого отправителем и получателем пакета. 0 Поля Key Source и Key Index отсутствуют.	

[1]Message Authentication Code - код проверки подлинности сообщения.

2	10	Использование группового ключа, определяемого полями Key Index и Key Source. 9 Поля Key Source и Key Index присутствуют.
3	11	Использование ключа подписи узла. Если пакет шифруется, применяется ключ 0/9 группы, заданный полями Key Index и Key Source. Поля Key Source и Key Index могут присутствовать.

Рисунок 10. Кодирование KIM.

В режиме 3 (KIM=11) наличие Key Source и Key Identifier зависит от описанного ниже уровня защиты (LVL). Если уровень задаёт шифрование, эти поля присутствуют, в режиме без шифрования их нет.

Resvd

3-битовое резервное поле. Отправитель **должен** устанавливать 0, а получатель **должен** игнорировать поле.

Security Level (LVL)

3-битовое поле уровня защиты указывает обеспечиваемую пакету защиту. Значение может задаваться на уровне пакета и позволяет обеспечивать разные уровни защиты целостности и конфиденциальности данных. Поле KIM определяет применение подписей и смысл поля Level. Отметим, что значения Security Level не упорядочены и большее значение LVL не обязательно задаёт более сильную защиту. Значения LVL показаны на рисунке 11.

KIM=0,1,2

LVL	*Атрибуты*	*Размер MAC*
0	MAC-32	4
1	ENC-MAC-32	4
2	MAC-64	8
3	ENC-MAC-64	8
4-7	Не выделены	-

KIM=3

LVL	*Атрибуты*	*Размер подписи*
0	Sign-3072	384
1	ENC-Sign-3072	384
2	Sign-2048	256
3	ENC-Sign-2048	256
4-7	Не выделены	-

Рисунок 11. Кодирование уровней защиты (LVL).

Атрибут MAC указывает, сто сообщение имеет код MAC указанного размера. Атрибут ENC указывает шифрование сообщения, Sign - наличие подписи указанного размера.

Flags

8-битовое резервное поле. Отправитель **должен** устанавливать 0, а получатель **должен** игнорировать поле.

Counter

Указывает не повторяющееся 4-октетное значение, служащее для создания криптографического механизма, реализующего защиту пакета и позволяющего обеспечить семантическую защиту (10.9.1. CCM Nonce).

Key Identifier

Поле, указывающее ключ, применяемый для защиты пакета. Поле обеспечивает разные уровни детализации защиты, включая парные (peer-to-peer) и групповые ключи, а также ключи подписи. Формат показан на рисунке 12.

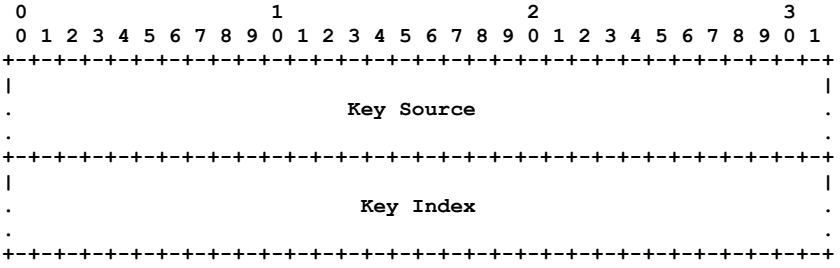


Рисунок 12. Идентификатор ключа.

Key Source

Необязательное 8-битовое поле, показывающее источник группового ключа.

Key Index

Необязательное 1-байтовое поле, однозначно указывающее разные ключи из одного источника. Создатель каждого ключа должен гарантировать, что выпущенные им активные ключи имеют разные индексы, отличные от 0x00. Значение 0x00 зарезервировано для предустановленного общего ключа.

Не выделенные биты являются резервными. Отправитель **должен** сбрасывать их, а получатель **должен** игнорировать.

6.2. Сообщение DIS

Сообщение запроса информации DODAG (DIS) может служить для запроса информационного объекта DODAG у узла RPL. Это аналогично Router Solicitation в IPv6 Neighbor Discovery и узел может применять DIS для проверки соседства в DODAG. Отклики на сообщения DIS описаны в параграфе 8.3. Передача DIO.

6.2.1. Формат базового объекта DIS

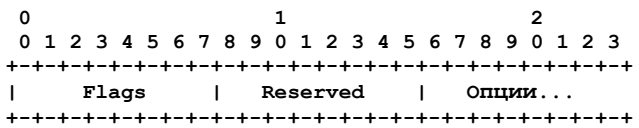


Рисунок 13. Базовый объект DIS.

Flags

8-битовое резервное поле. Отправитель **должен** устанавливать 0, а получатель **должен** игнорировать поле.

Reserved

8 резервных битов, которые **должны** сбрасываться (0) отправителем. Получатель **должен** игнорировать эти биты. Не выделенные биты являются резервными. Отправитель **должен** сбрасывать их, а получатель **должен** игнорировать.

6.2.2. Secure DIS

Сообщение Secure DIS имеет формат, показанный на рисунке 7, а базовый формат DIS показан на рисунке 13.

6.2.3. Опции DIS

Сообщение DIS **может** включать действительные опции и данная спецификация разрешает опции:

0x00 Pad1;

0x01 PadN;

0x07 Solicited Information.

6.3. Информационный объект DODAG

Информационный объект DODAG содержит сведения, позволяющие узлу обнаружить экземпляр RPL, узнать параметры его конфигурации, выбрать родительский набор DODAG и поддерживать DODAG.

6.3.1. Формат базового объекта DIO

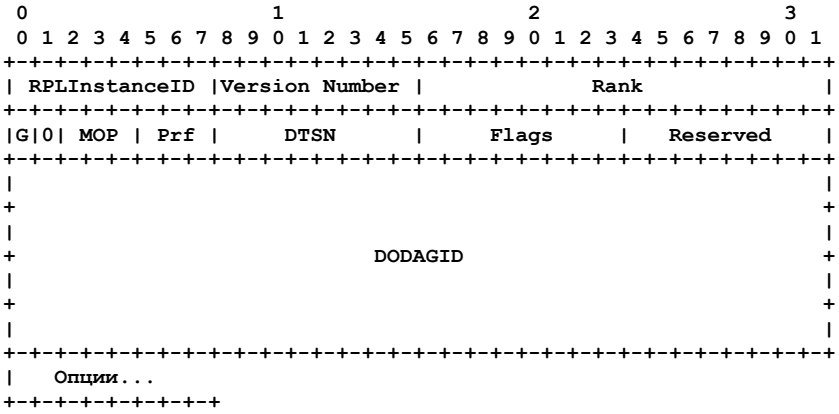


Рисунок 14. Базовый объект DIO.

Grounded (G)

Флаг G указывает соответствие анонсированного графа DODAG заданным приложением целям. Установленный флаг показывает «приземлённый» граф DODAG, сброшенный - «плавающий» граф.

Mode of Operation (MOP)

Поле MOP указывает режим работы RPL Instance, заданный административно и распространяемый корнем DODAG. Все присоединяющиеся к DODAG узлы должны быть способны поддерживать MOP для полноценной работы в качестве маршрутизаторов, а в противном случае могут служить лишь листьями графа. Кодирование режимов работы представлено на рисунке 15.

MOP	*Описание*
0	RPL не поддерживает нисходящих (Downward) маршрутов.
1	Режим работы без хранения маршрутов (Non-Storing).
2	Режим работы с хранением маршрутов (Storing) без поддержки групповой адресации.
3	Режим работы с хранением маршрутов и поддержкой групповой адресации.

Рисунок 15. Кодирование режимов работы (MOP).

Значение 0 указывает запрет сообщений с анонсами адресатов и поддержку в DODAG лишь восходящих (Upward) маршрутов. Не указанные на рисунке 15 значения не выделены.

DODAGPreference (Prf)

3-битовое целое число без знака, определяющее сравнение предпочтительного корня этого графа DODAG с другими корнями DODAG внутри экземпляра. DAGPreference принимает значения от 0x00 (наименьший приоритет) до 0x07 (наибольший приоритет). По умолчанию задано значение 0. Влияние DAGPreference на обработку DIO описано в параграфе 8.2. Обнаружение и поддержка восходящего маршрута.

Version Number

8-битовое целое число без знака, устанавливаемое корнем DODAG в соответствии с DODAGVersionNumber. Правила для DODAGVersionNumber и влияние на обработку сообщений DIO приведены в параграфе 8.2.

Rank

16-битовое целое число без знака, указывающее DODAG Rank узла, передающего сообщение DIO. Установка ранга и его влияние на обработку сообщений DIO описаны в параграфе 8.2.

RPLInstanceID

8-битовое поле, устанавливаемое корнем DODAG и указывающее экземпляр RPL для графа DODAG.

Destination Advertisement Trigger Sequence Number (DTSN)

8-битовое целое число без знака, устанавливаемое узлом, создавшим сообщение DIO. Поле DTSN является частью процедуры поддержки нисходящих маршрутов (см. раздел 9. Нисходящие маршруты).

Flags

8-битовое резервное поле. Отправитель **должен** устанавливать 0, а получатель **должен** игнорировать поле.

Reserved

8 резервных битов, которые **должны** сбрасываться (0) отправителем. Получатель **должен** игнорировать эти биты.

DODAGID

128-битовый адрес IPv6, устанавливаемый корнем DODAG и однозначно указывающий граф DODAG. Поле DODAGID **должно** содержать маршрутизируемый адрес IPv6, относящийся к корню DODAG.

Не выделенные биты являются резервными. Отправитель **должен** сбрасывать их, а получатель **должен** игнорировать.

6.3.2. Secure DIO

Сообщение Secure DIO имеет формат, показанный на рисунке 7, а базовый формат DIO показан на рисунке 14.

6.3.3. Опции DIO

Сообщение DIO **может** включать действительные опции и данная спецификация разрешает опции:

0x00 Pad1;

0x01 PadN;

0x02 DAG Metric Container;

0x03 Routing Information;

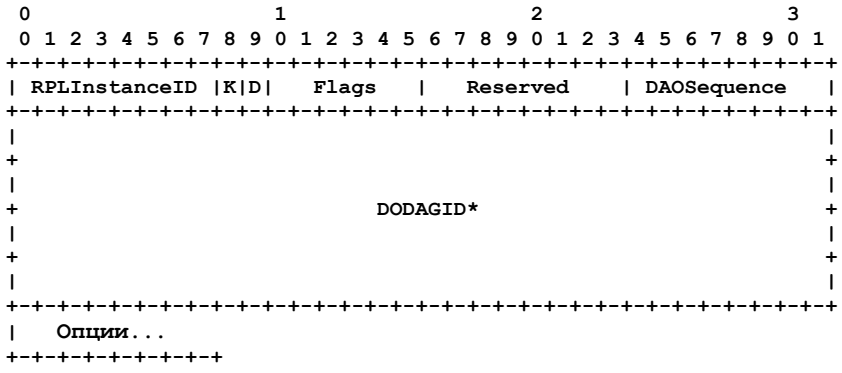
0x04 DODAG Configuration;

0x08 Prefix Information.

6.4. Сообщение DAO

Объект анонсирования получателя (DAO) служит для распространения сведений о получателе в восходящем направлении по графу DODAG. В режиме Storing сообщения DAO передаются дочерним узлом по индивидуальным адресам выбранных родителей, в режиме Non-Storing - по индивидуальному адресу корня DODAG. Сообщения DAO могут подтверждаться (по запросу или при ошибке) получателем в сообщении DAO-ACK.

6.4.1. Формат базового объекта DAO



* указывает необязательность поля DODAGID, как описано ниже.

Рисунок 16. Базовый объект DAO.

RPLInstanceID

8-битовое поле, указывающее экземпляр топологии, связанный с DODAG и определённый из DIO.

K

Флаг K указывает, что от получателя ожидается подтверждение DAO-ACK (см. 9.3. Базовые правила DAO).

D

Флаг D указывает наличие поля DODAGID и **должен** устанавливаться в случае локального RPLInstanceID.

Flags

6-битовое резервное поле. Отправитель **должен** устанавливать 0, а получатель **должен** игнорировать поле.

Reserved

8 резервных битов, которые **должны** сбрасываться (0) отправителем. Получатель **должен** игнорировать эти биты.

DAOSequence

Инкрементируется для каждого уникального сообщения DAO от узла и возвращается в DAO-ACK.

DODAGID

128-битовый адрес IPv6, устанавливаемый корнем DODAG и однозначно указывающий граф DODAG. Поле присутствует лишь при установленном флаге D. Обычно это связано с использованием локального RPLInstanceID и служит для указания DODAGID, связанного с RPLInstanceID. При глобальном RPLInstanceID поле не требуется.

Не выделенные биты являются резервными. Отправитель **должен** сбрасывать их, а получатель **должен** игнорировать.

6.4.2. Secure DAO

Сообщение Secure DAO имеет формат, показанный на рисунке 7, а базовый формат DAO показан на рисунке 16.

6.4.3. Опции DAO

Сообщение DAO **может** включать действительные опции и данная спецификация разрешает опции:

0x00 Pad1;

0x01 PadN;

0x05 RPL Target;

0x06 Transit Information;

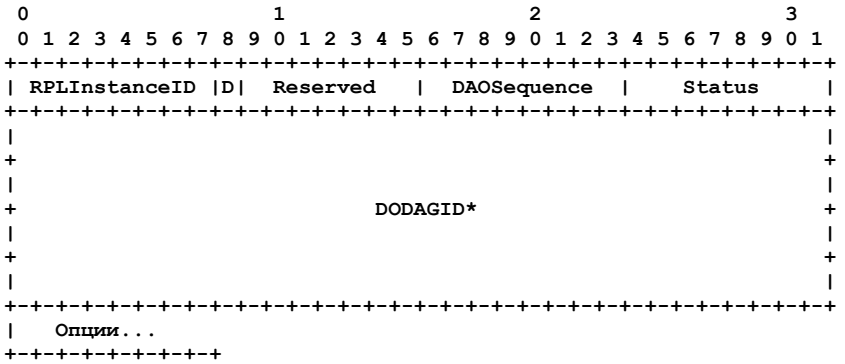
0x09 RPL Target Descriptor.

Особым случаем является сообщение DAO, называемое No-Path и применяемое в режиме Storing для очистки состояния маршрутизации Downward, полученного с помощью операции DAO. Сообщение No-Path содержит опцию Target и связанную с ней опцию Transit Information со сроком действия 0x0000[1] для индикации потери связности с данной целью Target.

6.5. Подтверждение DAO

Сообщение DAO-ACK передаётся в индивидуальном пакете получателем DAO (родитель DAO или корень DODAG) в ответ на индивидуальное сообщение DAO.

6.5.1. Формат базового объекта DAO-ACK



* указывает необязательность поля DODAGID, как описано ниже.

Рисунок 17. Базовый объект DAO-ACK.

RPLInstanceID

8-битовое поле, указывающее экземпляр топологии, связанный с DODAG и определённый из DIO.

D

Флаг D указывает наличие поля DODAGID и **должен** устанавливаться в случае локального RPLInstanceID.

Reserved

7 резервных битов для флагов.

DAOSequence

Инкрементируется для каждого уникального сообщения DAO от узла и возвращается в DAO-ACK. Поле служит для сопоставления DAO с DAO ACK и его не следует путать с Path Sequence в опции Transit Information, связанным с Target Down в DODAG.

Status

Указывает статус завершения. Значение 0 данная спецификация задаёт как безоговорочное восприятие, а все остальные значения служат для указания причины отказа. Спецификация не задаёт кодов отказа, их **следует** выделять в соответствии с приведёнными ниже рекомендациями:

0

Безусловное восприятие (т. е. узел, получивший DAO-ACK, не отвергается).

1-127

Неполный отказ - узел, передавший DAO-ACK, готов служить родителем, но принимающему узлу предлагается найти и применять другого родителя.

128[2]-255

Отказ - узел, передавший DAO-ACK, не готов служить родителем.

DODAGID

128-битовый адрес IPv6, устанавливаемый корнем DODAG и однозначно указывающий граф DODAG. Поле присутствует лишь при установленном флаге D. Обычно это связано с использованием локального RPLInstanceID и служит для указания DODAGID, связанного с RPLInstanceID. При использовании глобального RPLInstanceID это поле не требуется.

Не выделенные биты являются резервными. Отправитель **должен** сбрасывать их, а получатель **должен** игнорировать.

6.5.2. Secure DAO-ACK

Формат сообщения Secure DAO-ACK показан на рисунке 7, а базовый формат DAO-ACK - на рисунке 17.

6.5.3. Опции DAO-ACK

Эта спецификация не задаёт опций, передаваемых в сообщении DAO-ACK.

6.6. Проверка согласованности

Сообщение CC[3] служит для проверки счётчиков защищённых соединений и возврата откликов. Сообщение CC **должно** передаваться как защищённое сообщение RPL.

6.6.1. Формат базового объекта CC

RPLInstanceID

8-битовое поле, указывающее экземпляр топологии, связанный с DODAG и определённый из DIO.

R

Флаг R указывает, что сообщение CC является откликом. В запросах этот флаг сбрасывается (0).

[1] В оригинале ошибочно указано 0x00000000, см. https://www.rfc-editor.org/errata/eid3581. *Прим. перев.*
[2] В оригинале ошибочно указано 127, см. https://www.rfc-editor.org/errata/eid3287. *Прим. перев.*
[3] Consistency Check.

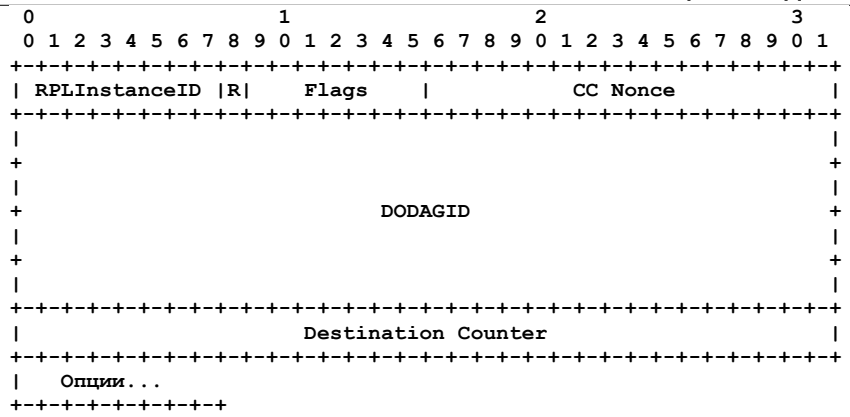


Рисунок 18. Базовый объект CC.

Flags
7-битовое резервное поле. Отправитель **должен** устанавливать 0, а получатель **должен** игнорировать поле.

CC Nonce
16-битовое целое число без знака, устанавливаемое запросом CC и включаемое в соответствующий отклик.

DODAGID
128-битовое поле с идентификатором корня DODAG.

Destination Counter
32-битовое целое число без знака, указывающее оценку отправителем текущего значения счётчика защиту у получателя. Если у отправителя нет такой оценки **следует** устанавливать значение 0.

Не выделенные биты являются резервными. Отправитель **должен** сбрасывать их, а получатель **должен** игнорировать.

Значение Destination Counter позволяет новым или восстановленным узлам ресинхронизировать обмен сообщениями CC. Это важно для того, чтобы значение счётчика не повторялось для данного ключа защиты при восстановлении устройств после отказа с потерей состояния счётчика. Например, когда запрос CC или иное сообщение RPL принято с инициализированным счётчиком в разделе Security, предоставление входящего значения счётчика в отклике CC позволяет запрашивающему узлу установить для исходящего счётчика значение, превышающее последнее значение, полученное от отвечающего узла. Входящий счётчик также будет обновлён из полученного отклика CC.

6.6.2. Опции CC

Эта спецификация позволяет передавать в сообщениях CC опции:

0x00 Pad1;

0x01 PadN.

6.7. Опции управляющих сообщений RPL

6.7.1. Базовый формат опций

Формат опций управляющего сообщения RPL показан на рисунке 19.

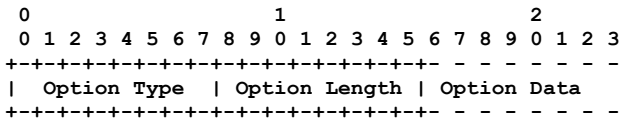


Рисунок 19. Базовый формат опций RPL.

Option Type
8-битовый идентификатор типа опции. Значения Option Type выделяются IANA (см. параграф 20.4).

Option Length
8-битовое целое число без знака, указывающее размер опции в октетах без полей Option Type и Option Length.

Option Data
Поле переменного размера, содержащее данные опции.

При обработке сообщения RPL с неизвестным значением Option Type получатель **должен** игнорировать неизвестную опцию и продолжать обработку.

В опциях сообщений RPL может применяться различное выравнивание. В соответствии с соглашениями IPv6 опции с требованиями к выравниванию размещаются в пакете так, что многооктетные значения в поле Option Data каждой опции выравниваются по естественной границе (т. е. поле размером n октетов размещается с кратным n смещением от начала заголовка, n может принимать значения 1, 2, 4, 8).

6.7.2. Pad1

Опция Pad1 **может** включаться в сообщения DIS, DIO, DAO, DAO-ACK, CC и имеет показанный на рисунке 20 формат.

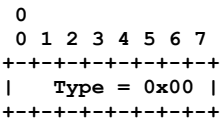


Рисунок 20. Опция Pad1.

Опция Pad1 служит для вставки в сообщение одного октета заполнения с целью выравнивания. Если нужно заполнить несколько октетов, следует применять опцию PadN, а не множество опций Pad1.

Важно подчеркнуть, что эта опция отличается от прочих отсутствием полей Option Length и Option Data.

6.7.3. PadN

Опция PadN **может** включаться в сообщения DIS, DIO, DAO, DAO-ACK, CC и имеет показанный на рисунке 21 формат.

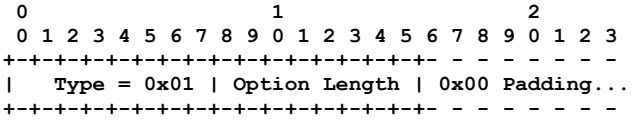


Рисунок 21. Опция PadN.

Опция PadN служит для вставки двух и более октетов в целях выравнивания. Получатели **должны** игнорировать PadN.

Type
0x01.

Option Length
Для N октетов заполнения (2 <= N <= 7) поле Option Length содержит значение N-2. Нулевое значение Option Length указывает 2 октета заполнения, значение 5 указывает максимальное заполнение в 7 октетов.

Option Data
Для N октетов заполнения (N > 1) поле Option Data содержит N-2 октетов со значением 0.

6.7.4. Контейнер метрики DAG

Опция DAG Metric Container **может** применяться в сообщениях DIO[1], а её формат показан на рисунке 22

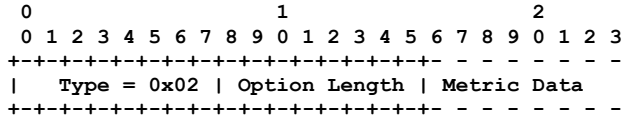


Рисунок 22. Опция DAG Metric Container.

DAG Metric Container служит для передачи метрики по графу DODAG и может содержать множество дискретных параметров узлов, каналов, агрегированных путей и ограничений, заданных в [RFC6551], по выбору разработчиков.

Опция DAG Metric Container **может** указываться неоднократно в одном сообщении RPL Control, например, для случаев, когда размер Metric Data превышает 256 байтов. Дополнительные сведения приведены в [RFC6551].

Обработка и распространение DAG Metric Container определяются зависящими от реализации функциями.

Option Type
0x02

Option Length
Размер Metric Data в октетах.

Metric Data
Порядок, содержимое и кодирование данных DAG Metric Container заданы в [RFC6551].

6.7.5. Route Information

Опция Route Information (RIO) **может** использоваться в сообщениях DIO и содержит такие же сведения, как IPv6 Neighbor Discovery (ND) RIO [RFC4191]. Информацию может устанавливать корень DODAG и она не меняется при распространении вниз по графу DODAG. Маршрутизатор RPL легко может преобразовать эти данные в опцию ND для анонсирования своих RA, поэтому подключенный к маршрутизатору RPL узел будет в конечном итоге использовать граф DODAG, корень которого является наиболее предпочтительным для адресата пакета. В дополнение к имеющейся семантике ND предметная функция OF может использовать эту информацию для предпочтения графа DODAG, корень которого лучше всего подходит для конкретного адресата. Формат опции слегка изменён (Type, Length, Prefix) для передачи в качестве опции RPL, как показано на рисунке 23.

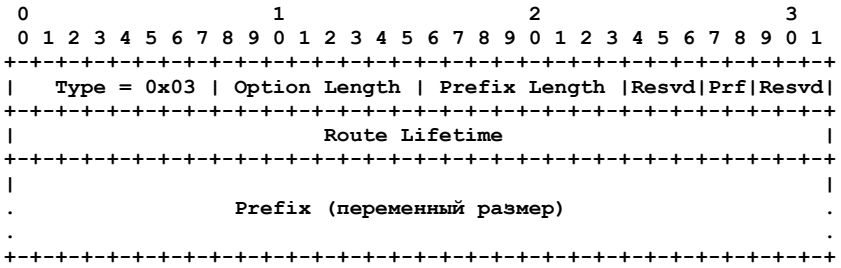


Рисунок 23. Опция Route Information.

Опция RIO служит для индикации связности с конкретным адресным префиксом из корня DODAG. Если в сообщении RPL Control нужно указать связность с множеством адресатов, опцию RIO можно использовать неоднократно. Следует использовать [RFC4191] как источник полномочных сведений о RIO. Поля опции представлены ниже.

Type
0x03

Option Length
Размер опции в октетах без учёта полей Type и Length (отметим, что в IPv6 ND размер указывается иначе).

Prefix Length
8-битовое целое число без знака, указывающее число действительных начальных биток префикса (0 - 128). Поле Prefix включает число битов, выводимое из поля Option Length, которое должно быть не меньше Prefix Length. В RPL это означает, что размер поля Prefix может отличаться от 0, 8, 16.

[1] В оригинале ошибочно сказано «DIO or DAO», см. https://www.rfc-editor.org/errata/eid5160. *Прим. перев.*

Prf

2-битовое целое число со знаком, указывающее предпочтительность маршрутизатора, связанного с данным префиксом, перед другими маршрутами при получении нескольких идентичных префиксов (для разных маршрутизаторов). При установке значения Reserved (10) опция RIO **должна** игнорироваться. В соответствии с [RFC4191] передача значения Reserved (10) **недопустима** (в [RFC4191] используется лишь 3 значения поля).

Resvd

Два 3-битовых резервных поля. Отправитель **должен** устанавливать 0, а получатель **должен** игнорировать поля.

Route Lifetime

32-битовое целое число без знака, указывающее интервал времени (в секундах с момента отправки пакета), в течение которого префикс действителен для определения маршрута. 0xFFFFFFFF задаёт неограниченное время.

Prefix

Поле переменного размера с IP-адресом или префиксом IPv6. Число действительных начальных битов префикса указывает поле Prefix Length. Биты префикса после указанного размера (при наличии) являются резервными и **должны** устанавливаться отправителем в 0, а получатель **должен** игнорировать их. В RPL размер поля Prefix может отличаться от 0, 8, 16.

Не выделенные биты являются резервными. Отправитель **должен** сбрасывать их, а получатель **должен** игнорировать.

6.7.6. Конфигурация DODAG

Опция DODAG Configuration **может** применяться в сообщениях DIO. Формат опции показан на рисунке 24.

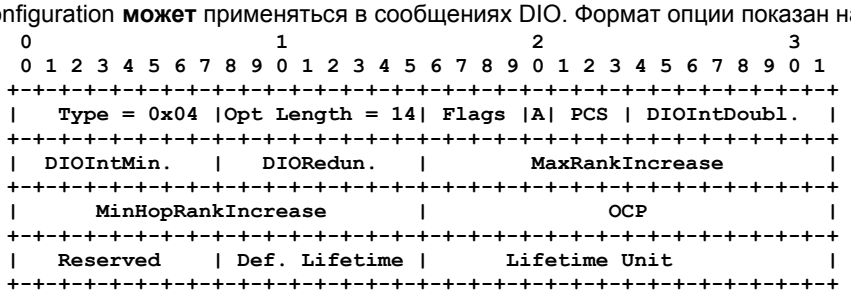


Рисунок 24. Опция DODAG Configuration.

Опция DODAG Configuration служит для распространения конфигурации DODAG Operation по графу DODAG. Эта опция обычно содержит статические сведения, которые не меняются внутри DODAG, поэтому её не требуется включать в каждое сообщение DIO. Информация задаётся в корне DODAG и распространяется в DODAG через опцию DODAG Configuration. Узлам, не являющимся корнем DODAG, **недопустимо** менять эту информацию при распространении опции DODAG Configuration. Опцию **может** время от времени включать корень DODAG (по своему усмотрению) и она **должна** включаться в отклики на индивидуальные запросы (например, индивидуальные сообщения DIS.

Type

0x04

Option Length

14

Flags

4-битовое резервное поле. Отправитель **должен** устанавливать 0, а получатель **должен** игнорировать поле.

Authentication Enabled (A)

Флаг режима защиты в сети, указывающий, должен ли узел аутентифицироваться в удостоверяющем центре перед подключением к сети в качестве маршрутизатора. Если сообщение DIO не является защищённым, бит A **должен** быть сброшен (0).

Path Control Size (PCS)

3-битовое целое число без знака, используемое для указания числа битов, которые могут быть выделены для поля Path Control (9.9. Управление путями). При определении размера поля к значению PCS добавляется 1, т. е. PCS = 0 задаёт 1 активный бит в поле Path Control. По умолчанию PCS = DEFAULT_PATH_CONTROL_SIZE.

DIOIntervalDoublings

8-битовое целое число без знака, служащее для настройки Imax таймера DIO Trickle (8.3.1. Параметры Trickle). По умолчанию DIOIntervalDoublings = DEFAULT_DIO_INTERVAL_DOUBLINGS.

DIOIntervalMin

8-битовое целое число без знака, служащее для настройки Imin таймера DIO Trickle (8.3.1. Параметры Trickle). По умолчанию DIOIntervalMin = DEFAULT_DIO_INTERVAL_MIN.

DIORedundancyConstant

8-битовое целое число без знака, служащее для настройки k таймера DIO Trickle (8.3.1. Параметры Trickle). По умолчанию DIORedundancyConstant = DEFAULT_DIO_REDUNDANCY_CONSTANT.

MaxRankIncrease

16-битовое целое число без знака, служащее для настройки DAGMaxRankIncrease, определяющего разрешённое увеличение Rank в поддержку локального восстановления. DAGMaxRankIncrease = 0 отключает механизм.

MinHopRankIncrease

16-битовое целое число без знака, служащее для настройки MinHopRankIncrease, как описано в параграфе 3.5.1. Сравнение ранга. По умолчанию MinHopRankInc = DEFAULT_MIN_HOP_RANK_INCREASE.

Objective Code Point (OCP)

16-битовое целое число без знака, указывающее OF (поддерживается IANA).

Reserved

8[1] резервных битов, которые **должны** сбрасываться (0) отправителем, а получатель **должен** игнорировать их.

Default Lifetime

8-битовое целое число без знака, задающее используемый по умолчанию срок действия маршрутов RPL в единицах Lifetime Unit.

[1] В оригинале ошибочно сказано 7, см. https://www.rfc-editor.org/errata/eid4618. *Прим. перев.*

Lifetime Unit

16-битовое целое число без знака, задающее единицу измерения времени действия маршрутов в RPL. Для стабильных сетей это могут быть часы или дни.

6.7.7. RPL Target

Опция RPL Target **может** присутствовать в сообщениях DAO. Формат опции показан на рисунке 25.

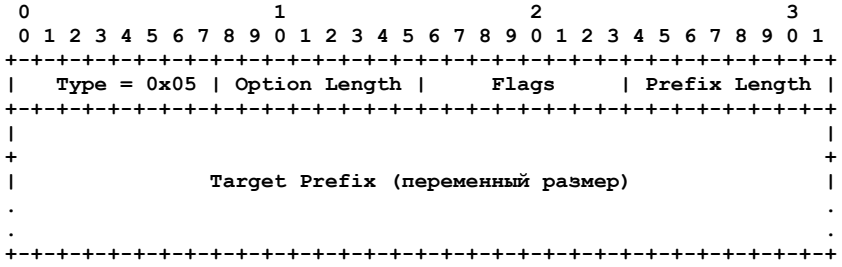


Рисунок 25. Опция RPL Target.

Опция RPL Target служит для указания адреса или префикса Target IPv6 или группы, доступной или запрашиваемой с DODAG. В сообщениях DAO опция RPL Target указывает доступность. Опция **может** сопровождаться опцией RPL Target Descriptor (рисунок 30), определяющей цель.

Одна или несколько опций Transit Information (параграф 6.7.8) **могут** следовать непосредственно за опциями Target в сообщении DAO (каждая такая опция Target **может** иметь парную опцию RPL Target Descriptor). Структура сообщения DAO, определяющая использование опций Target и Transit Information, описана в параграфе 9.4.

Опция RPL Target может повторяться для указания нескольких целей.

Type

0x05.

Option Length

Размер опции в октетах без учёта полей Type и Option Length.

Flags

8-битовое поле, зарезервированное для флагов. Отправитель **должен** устанавливать значение 0, а получатель **должен** игнорировать поле.

Prefix Length

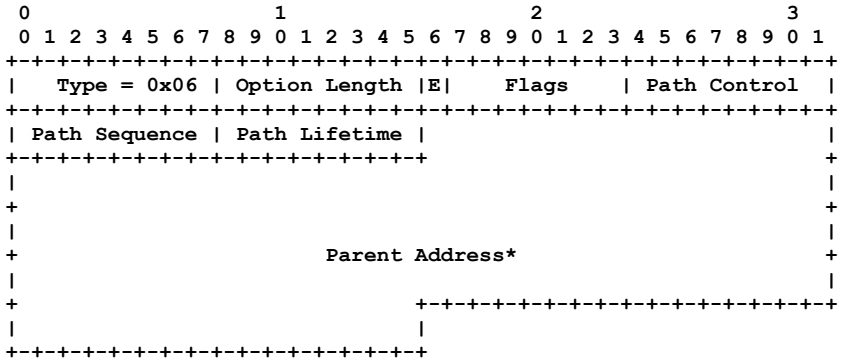
8-битовое целое число без знака, указывающее число действительных начальных битов в IPv6 Prefix.

Target Prefix

Поле переменного размера, указывающее адрес получателя IPv6, префикс или multicast-группу. Поле Prefix Length указывает число действительных начальных битов префикса. Биты префикса с большими номерами (при наличии) **должны** устанавливаться отправителем в 0, а получатель **должен** игнорировать их.

6.7.8. Transit Information

Опция Transit Information (рисунок 26) **может** присутствовать в сообщениях DAO.



* указывает, что поле DODAG Parent Address присутствует не всегда (см. ниже).

Рисунок 26. Опция Transit Information.

Опция Transit Information служит для указания атрибутов пути к одному или нескольким получателям, задаваемым одной или множеством опций Target, непосредственно предшествующих опции Transit Information.

Опция может применяться узлом с целью указания его родителей DODAG предку, собирающему маршрутные данные DODAG (обычно для создания задаваемых источником маршрутов). В режиме Non-Storing предком будет корень DODAG и опция передаётся в сообщении DAO. В режиме Storing поле Parent Address не нужно, поскольку сообщение DAO передаётся родителю напрямую. Поле Option Length позволяет определить наличие поля Parent Address.

Узел без хранения, имеющий несколько родителей DAO, **может** включать опцию Transit Information для каждого родителя DAO как часть операции анонсирования получателя без сохранения. Узел может распространять биты поля Path Control в разные группы родителей DAO для указания своих предпочтений. Эти предпочтения могут влиять на выбор корня DODAG среди родителей и путей при создании нисходящего маршрута (Downward).

Одна или несколько опций Transit Information **должны** присутствовать перед опциями RPL Target, при этом опция RPL Target указывает дочерний узел, а Transit Information - родителей DODAG. В параграфе 9.4, описывающем структуру сообщения DAO, приведены дополнительные детали об использовании опции Target вместе с Transit Information.

Типичный узел без хранения будет применять несколько опций Transit Information и передавать сформированное таким способом сообщение DAO напрямую корневому узлу. Типичный узел с хранением будет использовать одну опцию

Transit Information без поля родителя и передавать сообщение DAO с дополнительными аргументами в Path Control, как описано ниже, для одного или нескольких родителей.

Например, для режима Non-Storing предположим, что Tgt(T) обозначает опцию Target для Target T, а Trnst(P) - опцию Transit Information, содержащую адрес родителя P. Рассмотрим случай, когда узел N без хранения анонсирует принадлежащие ему цели N1 и N2 и имеет родителей P1, P2, P3. В этом случае предполагается, что сообщение DAO содержит последовательность ((Tgt(N1), Tgt(N2)), (Trnst(P1), Trnst(P2), Trnst(P3))) и группа опций Target {N1, N2} описывается опциями Transit Information, как имеющая родителей {P1, P2, P3}. Узел без сохранения будет адресовать это сообщение DAO напрямую корню DODAG и пересылать сообщение DAO через одного из родителей (P1, P2, P3).

Type
0x06.

Option Length
Определяет наличие или отсутствие поля DODAG Parent Address.

External (E)
Флаг E индикации того, что родительский маршрутизатор распространяет внешние цели в сеть RPL. Внешней считается цель, полученная от другого протокола. Внешние цели указываются в опциях Target, которые непосредственно предшествуют опции Transit Information. Для внешних целей не предполагается поддержка сообщений и опций RPL.

Flags
7-битовое поле для флагов. Отправитель **должен** устанавливать 0, а получатель **должен** игнорировать поле.

Path Control
8-битовое поле, ограничивающее число родителей DAO, которым передаётся сообщение DAO, анонсирующее связность с конкретным адресатом, а также обеспечивается та или иная индикация относительных предпочтений. Это поле задаёт некоторое ограничение общего числа сообщений DAO в сети LLN. Назначение и порядок битов Path Control служат также для передачи предпочтений. Не все эти биты могут быть активированы в соответствии с PCS в DODAG Configuration. Поле Path Control делится на 4 части по два бита в каждой (PC1, PC2, PC3, PC4), как показано на рисунке 27. Субполя упорядочиваются по уровню предпочтения, начиная с более предпочтительного (PC1). В субполях порядок предпочтения отсутствует. Утем группировки (как в ECMP) и упорядочения родителей можно связать их в конкретными битами поля Path Control для указания предпочтений.

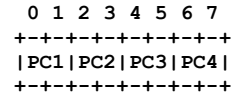


Рисунок 27. Поле Path Control.

Path Sequence
8-битовое целое число без знака. Когда опция RPL Target вводится узлом, владеющим префиксом Target (т. е. в сообщении DAO), этот узел устанавливает Path Sequence и инкрементирует поле каждый раз при внесении опции RPL Target с обновлённой информацией.

Path Lifetime
8-битовое целое число без знака, указывающее срок действия определения маршрута в Lifetime Unit (из опции Configuration). Отсчёт начинается с получения нового значения Path Sequence. Поле, включающее только 1 (0xFF), указывает неограниченный срок, а заполнение поля нулями (0x00) говорит о потере связности. Сообщение DAO с опцией Transit Information, включающей Path Lifetime = 0x00 для цели Target, в этом документе называется No-Path (нет пути к Target).

Parent Address
Необязательное поле с адресом IPv6 родителя DODAG для узла, изначально внесшего опцию Transit Information. Наличие этого поля зависит от режима работы DODAG (Storing или Non-Storing) и указывается полем размера опции Transit Information.

Не выделенные биты являются резервными. Отправитель **должен** сбрасывать их, а получатель **должен** игнорировать.

6.7.9. Solicited Information

Опция Solicited Information (рисунок 28) **может** включаться в сообщения DIS.

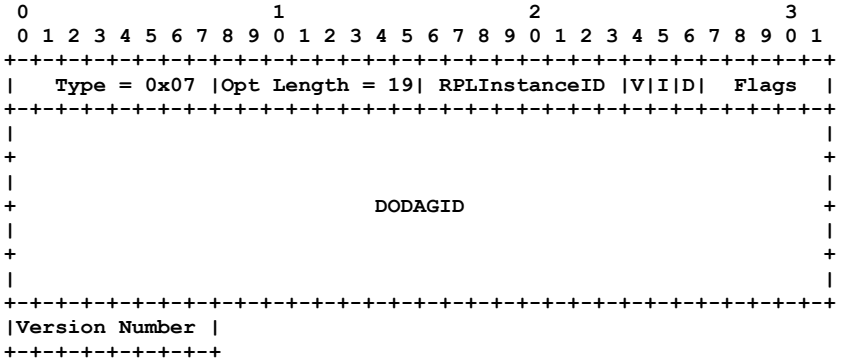


Рисунок 28. Опция Solicited Information.

Опция служит узлам для запроса сообщений DIO от подмножества соседних узлов и может указывать предикаты, проверяемые запрашивающей стороной. Это служит для ограничения запрашивающим узлом числа откликов от «неинтересных» узлов и влияет на сброс узлом таймера DIO Trickle, как описано в параграфе 8.3. Передача DIO.

Опция Solicited Information содержит флаги, указывающие, что следует проверять узлу при решении вопроса о сбросе таймера Trickle. Таймер сбрасывается узлом при выполнении всех условий. При установленном флаге узел RPL **должен** проверить соответствующий предикат. Если флаг сброшен, узлу RPL недопустимо проверять условие (предполагается, что оно выполнено).

Type
0x07
Option Length
19
V
Флаг V является предикатом версии. Условие выполняется, если значение DODAGVersionNumber у получателя совпадает с запрошенным Version Number. При сброшенном флаге V поле Version недействительно, **должно** быть сброшено в 0 при передаче и игнорироваться при получении.
I
Флаг I является предикатом InstanceID. Условие выполняется, если текущее значение RPLInstanceID для узла RPL совпадает с запрошенным RPLInstanceID. При сброшенном флаге I поле RPLInstanceID недействительно, **должно** быть сброшено в 0 при передаче и игнорироваться при получении.
D
Флаг D является предикатом DODAGID. Условие выполняется, если текущее значение DODAGID для набора родителей узла RPL совпадает с полем DODAGID. При сброшенном флаге D поле DODAGID недействительно, **должно** быть сброшено в 0 при передаче и игнорироваться при получении.
Flags
5 резервных битов, которые **должны** сбрасываться (0) отправителем. Получатель **должен** игнорировать эти биты.
Version Number
8-битовое целое число без знака с запрошенным значением DODAGVersionNumber, если оно действительно.
RPLInstanceID
8-битовое целое число без знака с запрошенным значением RPLInstanceID, если оно действительно.
DODAGID
128-битовое целое число без знака с запрошенным значением DODAGID, если оно действительно.
Не выделенные биты являются резервными. Отправитель **должен** сбрасывать их, а получатель **должен** игнорировать.

6.7.10. Prefix Information

Опция Prefix Information Option (PIO) **может** применяться в сообщениях DIO и передаёт сведения, заданные для опции IPv6 ND Prefix Information в [RFC4861], [RFC4862], [RFC6275], используемые узлами RPL и хостами IPv6. В частности, узел RPL может применять эту опцию для автоматической настройки адресов без учёта состояния (Stateless Address Autoconfiguration или SLAAC) по анонсированному родителем префиксу, как указано в [RFC4862], и анонсировать свой адрес в соответствии с [RFC6275]. Устанавливать эту опцию может корень DODAG. Данные распространяются в нисходящем направлении DODAG без изменения, за исключением того, что маршрутизатор RPL может изменить Interface ID (если установлен флаг R) для указания его полного адреса в PIO. Формат опции изменён (Type, Length, Prefix) для передачи в RPL, как показано на рисунке 29.

Если в результате получения PIO в сообщении DIO нужно лишь представить глобальный адрес родителя принимающему узлу, отправитель сбрасывает (0) флаги A и L, а флаг R устанавливает (1). При получении RPL на будет автоматически настраивать адрес или присоединённый маршрут по полученному префиксу [RFC4862]. Во всех случаях при сброшенном флаге L узел RPL **может** включать префикс в передаваемые своим потомкам опции PIO.

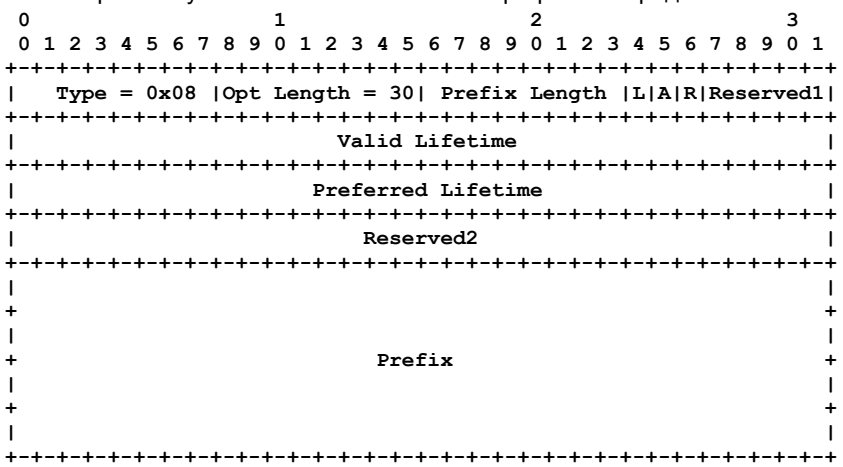


Рисунок 29. Опция Prefix Information.

Опцию PIO можно применять для распространения используемого в DODAG префикса, например, для автоматической настройки адресов.

Документы [RFC4861] и [RFC6275] содержат полномочные сведения о PIO. Поля опции представлены ниже.

Type
0x08
Option Length
30. В отличии от IPv6 ND размер указывается в октетах.
Prefix Length
8-битовое целое число без знака, указывающее число действительных начальных битов в поле Prefix (0 - 128). Поле обеспечивает сведения для определения принадлежности к каналу (вместе с флагом L), а также помогает при автоматической настройке адресов [RFC4862], для которой могут быть заданы дополнительные ограничения.
L
Флаг принадлежности к каналу (on-link), установка которого показывает, что этот префикс можно использовать для определения принадлежности к каналу. При сброшенном флаге анонс ничего не говорит о принадлежности префикса к каналу. Иными словами, при сброшенном флаге L узлу RPL **недопустимо** считать, что полученный из префикса адрес не относится к каналу (off-link), т. е. узлу **недопустимо** обновлять прежнюю индикацию

принадлежности адреса к каналу. Узлу RPL, действующему как маршрутизатор, **недопустимо** распространять PIO с установленным флагом L, но **можно** распространять PIO со сброшенным флагом L.

A

Флаг автономной настройки адреса, установка которого показывает, что префикс можно применять для автоматической настройки адресов без учёта состояния, как задано в [RFC4862]. При использовании обоих протоколов (ND RA и RPL DIO) для передачи PIO на одном канале, узлу RPL можно применять любой из них для SLAAC. Любой из протоколов можно также задать нежелательным применять для работы SLAAC сбросив (0) флаг A в опциях PIO этого протокола.

R

Флаг, установка которого указывает, что поле Prefix содержит полный адрес IPv6, присвоенный передающему маршрутизатору, который может указывать родителя в опции Transit[1]. Указанный префикс содержится в первых Prefix Length битах поля Prefix. Адрес маршрутизатор IPv6 имеет те же область и срок действия, что и анонсируемый префикс. Такое использование поля Prefix совместимо с применением для анонсирования самого префикса, где тоже применяются лишь старшие биты. Таким образом, интерпретация флага не зависит от обработки, требуемой для флагов L и A.

Reserved1

5 резервных битов, которые **должны** сбрасываться (0) отправителем. Получатель **должен** игнорировать эти биты.

Valid Lifetime

8-битовое целое число без знака, указывающее срок действия префикса (в секундах с момента передачи пакета) для определения принадлежности к каналу. Значение 0xFFFFFFFF указывает неограниченный срок. Значение Valid Lifetime применяется также в [RFC4862].

Preferred Lifetime

8-битовое целое число без знака, указывающее срок (в секундах с момента передачи пакета), в течение которого адрес, созданный из префикса автоматической настройкой без учёта состояния, остаётся предпочтительным [RFC4862]. Значение 0xFFFFFFFF указывает неограниченный срок. Отметим, что в этом поле **недопустимо** указывать значение больше Valid Lifetime во избежание предпочтительности недействительного адреса.

Reserved2

Резервное поле, которое **должно** сбрасываться (0) отправителем. Получатель **должен** игнорировать это поле.

Prefix

Адрес или префикс IPv6, число действительных старших битов которого указывает поле Prefix Length. Остальные биты **должны** устанавливаться в 0 отправителем и игнорироваться получателем. Маршрутизаторам **не следует** передавать опцию для префиксов link-local, а хостам **следует** игнорировать такие опции. В режиме без хранения **следует** воздерживаться от анонсирования не принадлежащих узлу префиксов, а для тех **следует** анонсировать в этом поле полный адрес с установкой флага R. Потомки узла, анонсирующего полный адрес с флагом R, могут применять этот адрес для определения содержимого поля DODAG Parent Address в опции Transit Information.

Не выделенные биты являются резервными. Отправитель **должен** сбрасывать их, а получатель **должен** игнорировать.

6.7.11. RPL Target Descriptor

За опцией RPL Target **может** сразу же следовать неанализируемый (opaque) дескриптор, указывающий цель.

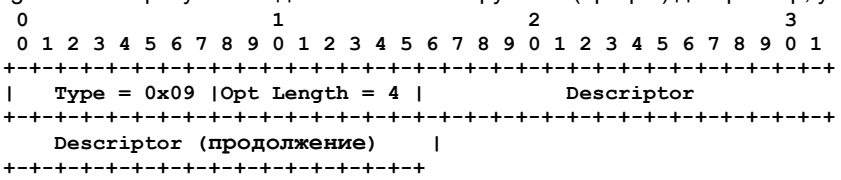


Рисунок 30. Опция RPL Target Descriptor.

Опция RPL Target Descriptor служит для указания цели и это иногда называют «тегированием» (tagging).

В большинстве случаев используется один дескриптор на цель. Дескриптор устанавливается узлом, вводящим Target в сеть RPL. Он **должен** копироваться без изменения маршрутизаторами, которые распространяют Target Up DODAG в сообщениях DAO.

Type

0x09

Option Length

4

Descriptor

32-битовое целое число без знака. Не анализируется (opaque).

7. Счётчики

В этом разделе описана схема инициализации и работы счётчиков последовательностей в RPL, таких как DODAGVersionNumber в сообщении DIO, DAOSequence в сообщении DAO, Path Sequence в опции Transit Information.

7.1. Обзор счётчиков

В этой спецификации применяется 3 порядковых номера для обеспечения свежести и синхронизации данных.

DODAGVersionNumber

Этот счётчик применяется в базовых сообщениях DIO для указания версии формируемого графа DODAG. Значение DODAGVersionNumber монотонно увеличивается корневым узлом каждый раз, когда он принимает решение о создании новой версии DODAG для повторной проверки целостности и обеспечения возможности глобального восстановления. DODAGVersionNumber распространяется без изменения в нисходящем направлении графа DODAG по мере присоединения маршрутизаторов к новой DODAG Version. DODAGVersionNumber имеет глобальную значимость в DODAG и указывает версию графа DODAG, с которой работает маршрутизатор. Более старое (меньшее) значение говорит о том, что маршрутизатор не перешёл к новой версии DODAG Version и не может служить родителем для узлов, перешедших к более новой версии DODAG.

[1] В оригинале ошибочно сказано target, см. https://www.rfc-editor.org/errata/eid3311. *Прим. перев.*

DAOSequence

Этот счётчик применяется в базовых сообщениях DAO для сопоставления сообщений DAO с DAO-ACK. DAOSequence имеет локальную значимость (узел) и служит для обнаружения потери и повтора DAO.

Path Sequence

Этот счётчик применяется в опции Transit Information сообщений DAO и служит для того, чтобы различать ситуации замены устаревшего маршрута новым и наличия нескольких (избыточных) маршрутов к одному адресату. Path Sequence имеет глобальную значимость в DODAG и показывает свежесть маршрута к адресату. Более старое (меньшее) значение, полученное от маршрутизатора, говорит о том, что он сохраняет устаревший маршрут, который больше не может служить следующим интервалом пересылки для адресата. Значение Path Sequence рассчитывается узлом, анонсирующим адресата, т. е. Target или маршрутизатором, анонсирующим Target от имени хоста, и не меняется при распространении содержимого DAO родительскими маршрутизаторами в направлении корня. Если хост не передаёт маршрутизатору значение счётчика, маршрутизатор сам рассчитывает Path Sequence от имени хоста и хост может регистрировать для этого лишь один маршрутизатор. Если сообщения DAO включают одно значение Target, заданное для нескольких родителей одновременно с целью создания избыточных маршрутов, значение Path Sequence будет одинаковым во всех сообщениях DAO для этого адресата.

7.2. Работа счётчиков

Счётчики порядковых номеров RPL делятся в стиле lollipop[1] [Perlman83], где значения 128 и выше служат линейной последовательностью для счётчика перезапусков и загрузок, а значения от 0 до 127 образуют циклическое пространство порядковых номеров как в [RFC1982]. При этом учитывается переход между линейной и циклической областью. Если в циклической области обнаруживается слишком большой зазор между номерами, эти номера считаются не сравнимыми, как описано ниже.

Окно сравнения SEQUENCE_WINDOW = 16 настраивается на основе значения 2^N и данная спецификация задаёт N = 4. Для данного счётчика операции сравнения следуют приведённым ниже правилам.

1. Счётчик номеров **следует** инициализировать заданным реализацией значением не меньше 128. Рекомендуется использовать значение 240 (256 - SEQUENCE_WINDOW).

2. Когда инкрементирование ведёт к переходу через максимальное значение, счётчик **должен** сбрасываться в 0. Для линейного (128 и больше) счётчика максимальным значением является 255, для циклического (меньше 128) - 127.

3. При сравнении значений счётчиков **должны** применяться приведённые ниже правила.

    1. Если первый счётчик A относится к интервалу [128-255], а второй счётчик B - к интервалу [0..127]:

        1. если (256 + B - A) не больше SEQUENCE_WINDOW, значение B считается большим, нежели A, A меньше B и счётчики не равны;

        2. если (256 + B - A) больше SEQUENCE_WINDOW, значение A считается большим, нежели B, B меньше A и счётчики не равны.

        Например, при A = 240 и B = 5 выражение (256 + 5 - 240) даёт значение 21, которое превышает SEQUENCE_WINDOW (16). Таким образом, 240 больше чем 5. Если A = 250, а B = 5, выражение (256 + 5 - 250) даёт значение 11, которое меньше SEQUENCE_WINDOW (16). В результате получаем, что 250 меньше чем 5.

    2. Если оба сравниваемых значения не превышают 127 или оба не меньше 128:

        1. применяется сравнение [RFC1982], если абсолютное значение разницы между счётчиками не превышает SEQUENCE_WINDOW, ;

        2. считается, что возникла рассинхронизация и значения не сравнимы, если абсолютное значение разницы между счётчиками превышает SEQUENCE_WINDOW,.

4. Если два порядковых номера не сравнимы, узлу следует отдать предпочтение номеру, который был инкрементирован позднее. Если это невозможно, узлу следует выбирать результат, минимизирующий смену его состояния.

8. Восходящие маршруты

В этом разделе описано, как RPL обнаруживает и поддерживает восходящие (Upward) маршруты. Описано использование объектов DIO, которые служат сообщениями для обнаружения и поддержки таких маршрутов. Описано, как RPL генерирует DIO и отвечает на них. Рассматриваются также сообщения DIS, которые служат для инициирования передачи DIO.

Как отмечено в параграфе 3.2.8, узлы, принявшие решение о присоединении к DODAG, **должны** указать по меньшей мере одного родителя DODAG в качестве принятого по умолчанию маршрута для связанного экземпляра. Этот маршрут позволяет пересылать пакеты в восходящем направлении, пока они не достигнут общего предка, где направление будет сменено на нисходящее (к получателю). Если адресат не входит в DODAG, корень DODAG может оказаться способным переслать пакет, используя связность за пределами DODAG. При невозможности такой пересылки корень DODAG будет отбрасывать пакет.

Сообщение DIO может также включать явную маршрутную информацию.

DODAGID

DODAGID - это глобальный или уникальный в локальном масштабе адрес IPv6 для корня. Присоединяющемуся к DODAG узлу **следует** обеспечивать маршрут (host route) через родителя DODAG к адресу, используемому корнем как DODAGID.

[1] Леденец на палочке.

RIO Prefix

Корень **может** включить одну или несколько опций Route Information в сообщение DIO. Опции RIO служат для анонсирования внешних маршрутов, доступных через корень и связанных с ними предпочтений, как указано в параграфе 6.7.5, заимствующем RIO из [RFC4191]. Это считается возможностью корня, а не анонсом маршрута и такие опции **недопустимо** распространять в другие протоколы маршрутизации, хотя их **следует** использовать на входных маршрутизаторах RPL для выбора DODAG при внесении в домен RPL пакетов от узла, подключённого к этому маршрутизатору RPL. Предметная функция OF **может** использовать анонсированные в RIO маршруты или предпочтения для них при выборе DODAG среди прочих для того же экземпляра.

8.1. Базовые правила DIO

1. Для перечисленных ниже полей сообщений DIO Base узел, не являющийся корнем DODAG, **должен** анонсировать те же значения, что и его предпочтительный родитель DODAG (8.2.1. Соседи и родители внутри версии DODAG). Таким способом эти значения будут распространяться в нисходящем направлении графа DODAG без изменении и анонсироваться каждым узлом, имеющим маршрут к корню DODAG.
    1. Grounded (G).
    2. Mode of Operation (MOP).
    3. DAGPreference (Prf).
    4. Version.
    5. RPLInstanceID.
    6. DODAGID.
2. На каждом этапе пересылки узлы **могут** обновлять поля:
    1. Rank;
    2. DTSN.
3. Значение DODAGID каждого корня **должно** быть уникальным в рамках экземпляра RPL, а также **должно** быть маршрутизируемым адресом IPv6, относящимся к корню.

8.2. Обнаружение и поддержка восходящего маршрута

Обнаружение восходящих маршрутов позволяет узлу присоединиться к графу DODAG путём обнаружения соседей, входящих в интересующий граф DODAG, и определения набора родителей. Правила выбора соседей и родителей зависят от реализации и выполняются предметной функцией OF. В этом разделе дан набор правил, которые следует соблюдать для поддержки взаимодействия.

8.2.1. Соседи и родители внутри версии DODAG

Алгоритмы обнаружения и обработки восходящих маршрутов RPL основаны на использовании трёх наборов локального канала. Набор кандидатов в соседи является подмножеством узлов, доступных по групповому адресу link-local. Выбор узлов зависит от реализации и функции OF. Набор родителей является частью набора кандидатов в соседи. Предпочтительными родителями являются узлы из набора родителей, являющиеся предпочтительными интервалами пересылки для маршрутов Upward. Концептуально предпочтительным является один родитель, но их может быть несколько, если предпочтения и Rank для них идентичны. Требования к родительским узлам даны ниже.

1. Набор родителей DODAG **должен** быть подмножеством набора кандидатов в соседи.
2. Корень DODAG **должен** иметь пустой набор родителей DODAG.
3. Не являющийся корнем DODAG узел **может** поддерживать непустой набор родителей DODAG.
4. Предпочтительный родитель DODAG для узла **должен** входить в его набор родителей DODAG.
5. Ранг узла должен быть больше рангов каждого из его родителей DODAG.
6. Когда механизм обнаружения недоступности соседа (Neighbor Unreachability Detection или NUD) [RFC4861] или его эквивалент определяет утрату доступности соседа, узлу RPL **недопустимо** включать такой узел в число кандидатов в соседи при расчёте и анонсировании маршрутов, пока узел снова не станет доступным. Маршруты через недоступных соседей **должны** удаляться из таблицы маршрутизации.

Эти правила обеспечивают согласованный частичный порядок узлов в графе DODAG. Пока ранг узлов не меняется, выполнение этих правил гарантирует отсутствие петель на пути от каждого узла к корню DODAG, поскольку значение Rank уменьшается на каждом этапе пересылки в направлении корня.

Функция OF может управлять набором кандидатов в соседи и выбором набора родителей, как описано в [RFC6552].

8.2.2. Соседи и родители в разных версиях DODAG

Приведённые выше правила работают в одной версии DODAG, а ниже описана работа RPL при наличии разных DODAG Version.

8.2.2.1. DODAG Version

1. Триплет (RPLInstanceID, DODAGID, DODAGVersionNumber) однозначно указывает DODAG Version. Каждый элемент в наборе родителей узла, как указано в последнем полученном сообщении DIO от каждого из родителей DODAG, **должен** относиться к одной версии DODAG. Элементы набора кандидатов в соседи узла **могут** относиться к разным DODAG Version.
2. Узел относится к версии DODAG, если каждый элемент его набора родителей DODAG относится к этой версии DODAG или этот узел является корнем соответствующего графа DODAG.

3. Узлу **недопустимо** передавать сообщения DIO для версий DODAG, в которые он не входит.

4. Корни DODAG **могут** инкрементировать анонсируемое значение DODAGVersionNumber, переходя таким образом к новой версии DODAG. При увеличении корнем DODAG значения DODAGVersionNumber он **должен** следовать правилам раздела 7. Счётчики. События, вызывающие инкрементирование DODAGVersionNumber, описаны ниже в этом разделе и разделе 18. Вопросы управляемости.

5. В данном графе DODAG узлу, не являющемуся корнем, **недопустимо** анонсировать DODAGVersionNumber, значение которого превышает максимальное полученное узлом значение DODAGVersionNumber. Сравнение значений рассмотрено в разделе 7. Счётчики.

6. Когда узел анонсировал DODAG Version, отправив сообщение DIO, ему **недопустимо** быть членом предыдущей версии DODAG в том же графе DODAG (с теми же RPLInstanceID, DODAGID и меньшим DODAGVersionNumber). Сравнение значений рассмотрено в разделе 7. Счётчики.

Когда для узла, не являющегося корнем, набор родителей DODAG становится пустым (удаляется последний предок и узел теряет связь с графом DODAG), информацию DODAG не следует подавлять до завершения отсчёта локального таймера, определяемого реализацией. В течение интервала, предшествующего удалению прежнего состояния DODAG, узел сможет наблюдать появление новых родителей с новым значением DODAGVersionNumber. Это помогает предотвращать возникновение петель при непреднамеренном повторном подключении узла к старой версии DODAG в прежнем субграфе DODAG. Когда DODAGVersionNumber инкрементируется, новое значение DODAG Version распространяется от корня DODAG. Предок, анонсирующий новое значение DODAGVersionNumber, не может относиться к субграфу DODAG узла, анонсирующего более старое значение DODAGVersionNumber, поэтому он может добавить родителя любого ранга с более новым значением DODAGVersionNumber без риска возникновения петли.

Предположим, например, что узел покинул DODAG с DODAGVersionNumber = N. Предположим, что узел имел суб-DODAG и пытается «отравить» этот субграф, анонсируя Rank = INFINITE_RANK, но эти анонсы могут теряться в LLN. Если узел наблюдал кандидата в соседи, анонсирующего позицию в исходном DODAG с DODAGVersionNumber = N, этот кандидат мог быть в прежнем суб-DODAG узла и возможен случай, когда добавление кандидата в соседи как родителя приведёт к созданию петли. В этом случае кандидат в соседи, анонсирующий DODAGVersionNumber N+1, безопасен, поскольку он не входит в исходный субграф DODAG узла, о чем говорит увеличение DODAGVersionNumber, переданное из корня DODAG, тогда как исходный узел был отсоединён. Поэтому для отсоединённого узла полезно помнить исходные сведения DODAG, включая DODAGVersionNumber = N.

Момент увеличения корнем DODAG значения DODAGVersionNumber зависит от реализации и это выходит за рамки данной спецификации. Примерами могут служить периодическое увеличение DODAGVersionNumber, изменение администратором или обнаружение потери связности или неэффективности DODAG на прикладном уровне.

После перехода узла и анонсирования новой версии DODAG приведённые выше правила не позволяют узлу анонсировать прежнюю версию DODAG (с меньшим DODAGVersionNumber).

8.2.2.2. Корни DODAG

1. Корню DODAG, не имеющему возможности достичь заданной приложением цели, **недопустимо** устанавливать флаг Grounded.

2. Корень DODAG **должен** анонсировать Rank = ROOT_RANK.

3. Узел с пустым набором родителей DODAG **может** стать корнем DODAG для «плавающего» графа DODAG, а также **может** установить DAGPreference, чтобы быть менее предпочтительным.

С системах, использующих каналы, не относящиеся к LLN, для объединения нескольких корней LLN, можно запустить RPL на чужих (не RPL) каналах и использовать один маршрутизатор как «корень магистрали». Этот корень будет виртуальным корнем DODAG с Rank = BASE_RANK, предоставляемым через магистраль. Все корни LLN, для которых магистральных корень является родителем, включая этот корень, если он является корнем LLN, показывают для LLN значение Rank = ROOT_RANK. Эти виртуальные корни являются частью одного графа DODAG и анонсируют одно значение DODAGID, координируя DODAGVersionNumber и другие параметры DODAG с виртуальным корнем через магистраль. Метод координации выходит за рамки спецификации (будет определён в сопровождающих документах).

8.2.2.3. Выбор DODAG

Предметная функция OF, а также набор анонсируемой маршрутной метрики и ограничений DAG определяют способ выбора узлом соседей и родителей, а также предпочтительных родителей. Этот выбор также неявно задаёт граф DODAG в DAG. Выбор может учитывать административные предпочтения (Prf), метрику и другие параметры.

Если у узла есть возможность присоединиться к более предпочтительному графу DODAG с соблюдением других целей оптимизации, узел обычно будет стремиться выбрать более предпочтительный граф DODAG, определённый функцией OF. При прочих равных условиях выбор предпочтительного DODAG остаётся за реализацией (напомним, что узел должен присоединяться лишь к одному графу DODAG на экземпляр RPL).

8.2.2.4. Ранг и перемещение внутри версии DODAG

1. Узлу **недопустимо** анонсировать Rank, не превышающий ранг любого из его родителей в DODAG Version.

2. Узел **может** анонсировать Rank меньше ранее анонсированного значения в DODAG Version.

3. Пусть L - наименьший ранг в DODAG Version, анонсированный данным узлом. В той же версии DODAG этому узлу **недопустимо** анонсировать эффективный Rank выше L + DAGMaxRankIncrease. Значение INFINITE_RANK является исключением и узел **может** анонсировать его в DODAG Version без ограничений. Если нужен ранг узла выше разрешённого L + DAGMaxRankIncrease, **должно** анонсироваться значение INFINITE_RANK.

4. Узел может в любой момент принять решение о присоединении к другому графу DODAG внутри RPL Instance. Для такого присоединения Rank не ограничивается, если граф DODAG не относится к DODAG Version, куда

данный узел был присоединён ранее. В последнем случае должно выполняться правило 3. Пока узел не передал сообщение DIO с новым графом DODAG, он **должен** пересылать пакеты по прежнему DODAG.

5. После получения следующего DODAGVersionNumber от подходящего родителя DODAG узел **может** в любой момент перейти к следующей версии DODAG внутри графа DODAG.

Реализация поддерживает набор родителей DODAG в рамках DODAG Version. Перемещение влечёт смену набора родителей DODAG. При перемещении вверх риска создания петли не возникает, но перемещение вниз связано с таким риском, поэтому для него имеются дополнительные ограничения.

При переходе узла к следующей версии DODAG набор родителей DODAG требуется перестроить для новой версии. Реализация может отложить переход на некоторое время, чтобы увидеть, анонсируют ли себя соседи с потенциально лучшей метрикой и большим рангом. Точно так же при переходе узла в новый граф DODAG ему требуется создать новый набор родителей DODAG.

Если узлу нужно переместить вниз граф DODAG, к которому он присоединён, путём увеличения ранга, он **может** «испортить» свои маршруты и задержать переход, как описано в параграфе 8.2.2.5. «Порча» маршрутов.

Узлу разрешается присоединение к любой версии DODAG, в которую он не входил ранее, но если узел входил в DODAG Version, он должен продолжать соблюдение правила, запрещающего анонсировать Rank выше L+DAGMaxRankIncrease в течение всего срока действия DODAG Version. Это требуется для предотвращения лазеек, позволяющим узлам повышать свой ранг до INFINITE_RANK, что может влиять на другие узлы и поглощать ресурсы.

8.2.2.5. «Порча» маршрутов

1. Узел «портит» (poison) свои маршруты, анонсируя Rank = INFINITE_RANK.
2. Узлу **недопустимо** иметь в своём наборе родителей узлы с Rank = INFINITE_RANK.

Хотя реализация может анонсировать INFINITE_RANK для «порчи» маршрутов, это отличается от установки Rank = INFINITE_RANK. Например, узел может продолжать передачу пакетов данных, где опция RPL Packet Information включает ранг, отличный от INFINITE_RANK, анонсируя INFINITE_RANK в своих сообщениях DIO.

Когда наблюдается анонсирование (прошлым) родителем Rank = INFINITE_RANK, это говорит об отсоединении (прошлого) родителя от графа DODAG и его невозможности оставаться родителем, а также отсутствии возможности найти узел с рангом больше INFINITE_RANK. Поэтому (прошлый) родитель удаляется из набора родителей.

8.2.2.6. Отсоединение

Узел, не способный оставаться соединенным с DODAG в данной версии DODAG (т. е. сохранять непустой набор родителей) без нарушения правил этой спецификации, **может** отсоединиться от DODAG Version. Отсоединённый узел становится корнем своего «плавающего» графа DODAG и ему **следует** незамедлительно анонсировать эту ситуацию в сообщении DIO альтернативу порче маршрутов.

8.2.2.7. Следование за родителем

Если узел получает от одного из родителей DODAG сообщение DIO, указывающее выход родителя из графа DODAG, этому узлу **следует** оставаться в текущем графе DODAG через другого родителя, когда это возможно. Узел **может** последовать за ушедшим родителем. Родитель DODAG может переместиться, поменять DODAG Version или перейти в другой граф DODAG. Узлу предпочтительно сохранять прежний граф DODAG через другого родителя, если это возможно, в противном случае он должен следовать за родителем.

8.2.3. Обмен сообщениями DIO

Получив сообщение DIO, узел должен сначала решить вопрос о его восприятии для обработки, как показано ниже.

1. Если сообщение DIO имеет некорректный формат, узел **должен** отбросить его, записав ошибку (раздел 18).
2. Если отправитель DIO является кандидатом в соседи, и формат сообщения корректен, узел **должен** обработать DIO.

8.2.3.1. Обработка сообщения DIO

Поскольку сообщения DIO принимаются от кандидатов в соседи, соседей можно объявить родителями DODAG, следуя правилам обнаружения, описанным в параграфе 8.2. При включении узлом соседа в набор родителей DODAG этот узел подключается к графу DODAG через нового родителя.

Следует использовать наиболее предпочтительного родителя для ограничения возможностей других узлов стать родителями DODAG. Некоторые узлы в наборе родителей DODAG могут иметь Rank не выше ранга предпочтительного родителя DODAG. Это может возникать, например, в случае, когда устройство с ограничениями по питанию имеет меньший ранг, но его следует избегать в целях оптимизации, предпочитая родителя с большим значением Rank.

8.3. Передача DIO

Узлы RPL передают сообщения DIO, используя таймер Trickle [RFC6206]. Сообщение DIO от отправителя с меньшим DAGRank, не вызывающее изменении в наборе родителей получателя, предпочтительного родителя или Rank, **следует** считать согласованным с таймером Trickle.

Ниже указаны события и пакеты, которые **должны** считаться не согласованными с таймером Trickle и сбрасывать его.

- Узел обнаруживает несогласованность при пересылке пакета, как указано в 11.2. Обнаружение и предотвращение петель.
- Узел получает групповое сообщение DIS без опции Solicited Information, пока флаг DIS не ограничивает это.
- Узел получает групповое сообщение DIS с опцией Solicited Information и соответствует всем предикатам в ней, пока флаг DIS не ограничивает это.

- Узел присоединяется к новой версии DODAG (например, обновляя DODAGVersionNumber, входя в новый экземпляр RPL и т. п.).

Список не является исчерпывающим и реализация может счесть несогласованными иные события и сообщения.

Узлу **не следует** сбрасывать таймер DIO Trickle в ответ на индивидуальное сообщение DIS. При получении такого сообщения без опции Solicited Information узел **должен** передать в ответ индивидуальное сообщение DIO, которое **должно** включать опцию DODAG Configuration. При получении индивидуального DIS с опцией Solicited Information и соответствии всем предикатам в этой опции узел **должен** передать в ответ индивидуальное сообщение DIO, которое **должно** включать опцию DODAG Configuration. Таким образом, узел **может** передать индивидуальное сообщение DIS потенциальному родителю DODAG для проверки DODAG Configuration и других параметров.

8.3.1. Параметры Trickle

Конфигурационные параметры таймера Trickle включают:

Imin
: извлекается из сообщения DIO как $2^{\text{DIOIntervalMin}}$ мсек (по умолчанию DIOIntervalMin = DEFAULT_DIO_INTERVAL_MIN);

Imax
: значение DIOIntervalDoublings в DIO (по умолчанию DIOIntervalDoublings = DEFAULT_DIO_INTERVAL_DOUBLINGS);

k
: значение DIORedundancyConstant в DIO (по умолчанию DEFAULT_DIO_REDUNDANCY_CONSTANT). В RPL значение k = 0x00 считается бесконечностью, т. е. таймер Trickle не будет подавлять сообщения.

8.4. Выбор DODAG

Выбор DODAG зависит от реализации и предметной функции OF. Для минимизации ошибочных переходов и равнозначности всех метрик узлам **следует** сохранять свой предшествующий выбор. Кроме того, узлам **следует** обеспечивать способ фильтрации родителей, для которых отмечены флуктуации доступности (хотя бы в при наличии других претендентов).

Когда подключение к приземлённому графу DODAG невозможно или нежелательно изи соображений безопасности или по иным причинам, **можно** агрегировать разрозненные DODAG в более крупные графы для обеспечения связности в сети LLN.

Узлу **следует** проверять наличие двухсторонней связи и адекватного качества канала для кандидатов в соседи до их рассмотрения в качестве возможных родителей DODAG.

8.5. Работа листа

В некоторых случаях узел RPL может подключать граф DODAG лишь в качестве листа. Такой случай возникает, например, когда узел не понимает или не поддерживает правила OF экземпляра RPL или анонсированную метрику или ограничения. Как отмечено в параграфе 18.6 применительно к функции правил, узел может присоединиться к DODAG как лист или отказаться от присоединения к DODAG. Как отмечено в параграфе 18.5, такие события рекомендуется записывать в журнал как отказы.

Лист не расширяет связность DODAG, однако в некоторых случаях от него может требоваться передача сообщений DIO, в частности, он может не всегда оставаться листом и может обнаруживаться несогласованность. Узел, являющийся листом, должен соблюдать перечисленные ниже правила.

1. **Недопустима** передача листом сообщений DIO с DAG Metric Container.
2. Сообщения DIO от листа **должны** анонсировать DAGRank = INFINITE_RANK.
3. Лист **может** подавлять отправку сообщений DIO, если передача не вызвана обнаружением несогласованности при пересылке пакета или индивидуальным сообщением DIS, когда подавлять отправку DIO **недопустимо**.
4. Лист **может** передавать индивидуальные сообщения DAO, как указано в параграфе 9.2.
5. Лист **может** передавать групповые сообщения DAO соседям 1-hop, как описано в параграфе 9.10.

Конкретная необходимость отправки сообщений DIO листом возникает в случае, когда этот лист прежде входил в другой граф DODAG, а другой узел пересылает сообщение на основе прежней топологии, что вызывает несогласованность. Следует отметить, что сетям LLN присущи потери пакетов и даже при возможности листа «испортить» свои маршруты путём анонсирования Rank = INFINITE_RANK в старом графе DODAG до перехода в статус листа эти анонсы могут быть потеряны и лист должен иметь возможность отправить сообщение DIO для устранения несогласованности.

В общем случае листу **недопустимо** анонсировать себя в качестве маршрутизатора (т. е. передавать сообщения DIO).

8.6. Административный ранг

В некоторых случаях может быть полезна корректировка анонсируемого узлом значения Rank, рассчитанного функцией OF на основе тех или иных зависящих от реализации правил и свойств узла. Например, недостаточная ёмкость батареи может требовать от узла быть листом, если нет иных вариантов, и такой узел может увеличить ранг, вычисленный функцией OF.

9. Нисходящие маршруты

В этом разделе описано, как RPL обнаруживает и поддерживает нисходящие (Downward) маршруты с помощью сообщений DAO. Нисходящие маршруты поддерживают потоки P2MP от корня DODAG к листьям. Для таких маршрутов поддерживаются также потоки P2P и сообщения P2P могут передаваться в направлении корня DODAG (или общего предка) через восходящие маршруты, затем от корня DODAG к адресату через маршрут Downward.

Данная спецификация описывает для RPL Instance два режима поддержки нисходящих маршрутов. В первом режиме, называемом Storing (сохранение) узлы хранят таблицы нисходящих маршрутов для своих суб-DODAG. Каждый интервал (hop) нисходящего маршрута в сохраняющей сети просматривает свою таблицу для выбора следующего интервала. Во втором режиме, называемом Non-Storing, узлы не хранят таблицы нисходящих маршрутов. Пакеты в нисходящем направлении маршрутизируются с помощью source route, задаваемых корнем DODAG [RFC6554].

RPL поддерживает простую одноинтервальную (one-hop) оптимизацию P2P для сетей с хранением и без него. Узел может передавать пакеты P2P, напрямую адресованные соседу (one-hop neighbor).

9.1. Родители для анонсов получателей

Для организации нисходящих маршрутов узлы RPL передают сообщения DAO в восходящем направлении (Upward). Узлы next-hop этих сообщений DAO называются родителями DAO (DAO parent). Набор родителей DAO для узла называется DAO parent set.

1. Узел **может** передавать сообщения DAO по групповому адресу all-RPL-nodes, что является оптимизацией для маршрутизации «в один этап» (one-hop routing). При передаче групповых DAO бит K **должен** быть сброшен.
2. Набор родителей DAO для узла **должен** быть подмножеством его набора родителей DODAG.
3. В режиме Storing узлу **недопустимо** адресовать индивидуальные сообщения DAO, узлам, не являющимся его родителями DAO.
4. В режиме Storing адреса IPv6 для отправителя и получателя сообщения DAO **должны** быть link-local.
5. В режиме Non-Storing узлу **недопустимо** адресовать индивидуальные сообщения DAO, узлам, не являющимся корнем DODAG.
6. В режиме Non-Storing адреса IPv6 для отправителя и получателя сообщения DAO **должны** уникальными в локальном масштабе (unique-local) или глобальными.

Выбор родителей DAO зависит от реализации и предметной функции OF.

9.2. Обнаружение и поддержка нисходящих маршрутов

Для DAO можно настроить полный запрет, а также работу в одном из режимов (Storing или Non-Storing), как указано в поле MOP сообщения DIO.

1. Все присоединяющиеся к DODAG узлы **должны** соблюдать режим MOP, заданный корнем. Узлы, не способные быть маршрутизаторами (например, не соответствующие анонсированному MOP), **могут** присоединяться к графу DODAG как листья.
2. Если MOP = 0 (указывает маршрутизацию Downward), узлам **недопустимо** передавать сообщения DAO и они **могут** игнорировать DAO.
3. В режиме Non-Storing корню DODAG **следует** сохранять записи таблицы source-route для адресатов, определённых из DAO. Корень DODAG **должен** быть способен генерировать маршруты source-route для полученных из DAOадресатов, которые были сохранены.
4. В режиме Storing все узлы, не являющиеся корнями или листьями, **должны** хранить записи таблицы маршрутизации для адресатов, полученных из DAO.

DODAG может работать в одном из возможных режимов, указанном полем MOP, поддерживая нисходящие маршруты (Downward) через source-route из корней DODAG, либо через таблицы маршрутизации в сети.

Когда маршруты Downward поддерживаются за счёт source-route от корней DODAG, обычно предполагается, что корень DODAG хранит данные source-route, полученные из DAO, для создания маршрутов. Если корень DODAG не может сохранить ту или иную информацию, некоторые адресаты могут стать недоступными.

При поддержке маршрутов Downward через таблицы маршрутизации в сети, могут применяться заданные в этой спецификации групповые операции, что также указывается в поле MOP. В этом режиме предполагается, что служащие маршрутизаторами узлы способны сохранять требуемое состояние таблиц маршрутизации. Если служащий маршрутизатором узел не способен хранить полную таблицу маршрутов, состояние маршрутизации становится неполным и сообщения могут отбрасываться с записью таких событий в системный журнал (параграф 18.5). В будущих расширениях RPL могут быть уточнены действия и варианты поведения в таких случаях.

На момент написания этой спецификации протокол RPL не поддерживал смешанный режим работы, где часть узлов применяет source-route, а другие используют таблицы маршрутизации. В будущих расширениях RPL может появиться поддержка такого режима работы.

9.2.1. Поддержка Path Sequence

Для каждой цели Target, связанной с узлом (принадлежащей ему), этот узел отвечает за отправку сообщений DAO для обеспечения нисходящих маршрутов. Опции Target и Transit information в сообщениях DAO распространяют восходящий маршрут DODAG. Счётчик Path в опции Transit information служит для указания свежести и обновления устаревшей информации о нисходящих маршрутах, как описано в разделе 7.

Для связанной с узлом (принадлежащей ему) цели Target этот узел **должен** инкрементировать счётчик Path Sequence и создавать новой сообщение DAO, когда:

1. значение Path Lifetime обновлено (например, refresh или no-Path);
2. список в поле DODAG Parent Address изменён.

Для связанной с узлом (принадлежащей ему) цели Target этот узел **может** инкрементировать счётчик Path Sequence и создавать время от времени новое сообщение DAO для обновления информации о нисходящих маршрутах. В режиме Storing узел генерирует такое сообщение DAO для каждого из своих родителей DAO с целью поддержки множества

путей. Все DAO, созданные одновременно для одной цели Target, **должны** передаваться с одним значением Path Sequence в опции Transit Information.

9.2.2. Генерация сообщений DAO

Узел может передавать сообщения DAO при получении DAO в результате изменения набора родителей DAO или иного события, такого как завершение срока действия префикса. В случае получения DAO важно, является это сообщение «новым» или содержит новую информацию. В режиме Non-Storing каждое принятое узлом сообщение DAO является «новым», а в режиме Storing «новыми» будут лишь сообщения DAO, соответствующие любому из приведённых ниже критериев для содержащегося в нем значения Target.

1. Новый номер Path Sequence.
2. Дополнительные биты Path Control.
3. Сообщение No-Path DAO, удаляющее последний маршрут Downward к префиксу.

Узел, получающий сообщение DAO от своего субграфа DODAG, **может** отменять планирование отправки сообщения DAO, если оно не является новым.

9.3. Базовые правила DAO

1. Если узел передаёт DAO с более новой или отличающейся от предыдущего DAO информацией, он **должен** увеличить значение DAOSequence по меньше мере на 1. Передача DAO идентичного предыдущему **может** инкрементировать поле DAOSequence.
2. Поля RPLInstanceID и DODAGID в DAO **должны** иметь те же значения, что и элементы набора родителей узла и передаваемые им DIO.
3. Узел **может** установить флаг K в индивидуальном сообщении DAO для запроса индивидуального отклика DAO-ACK с подтверждением попытки.
4. Узлу, получившему DAO с установленным флагом K, **следует** отвечать сообщением DAO-ACK. При сброшенном в DAO флаге K узел **может** передать DAO-ACK, особенно при возникновении ошибки.
5. Узел, установивший флаг K в индивидуальном сообщении DAO, но не получивший в ответ DAO-ACK, **может** запланировать повторную передачу DAO (число попыток определяет реализация).
6. Узлам **следует** игнорировать DAO без новых порядковых номеров и **недопустимо** обрабатывать такие сообщения.

В отличие от поля Version в DIO, инкрементируемого лишь корнем DODAG и передаваемого другими узлами без изменения, значения DAOSequence уникальны для каждого узла. Пространства номеров для индивидуальных и групповых DAO могут быть раздельными или общим. **Рекомендуется** применять общее пространство номеров.

9.4. Структура сообщений DAO

Структура DAO одинакова в сетях Storing и Non-Storing. В наиболее общей форме сообщение DAO может включать несколько групп опций, каждая из которых включает одну или несколько опций Target, за которыми следует одна или несколько опций Transit Information. Вся группа опций Transit Information применяется ко всей группе опций Target. Ниже описаны детали каждого режима работы.

1. Узлы RPL **должны** включать одну или несколько опций RPL Target в каждое передаваемое сообщение DAO. Одна опция RPL Target **должна** иметь префикс, включающий адрес узла IPv6, если узлу нужен граф DODAG для предоставления нисходящего маршрута к себе. Сразу за RPL Target **может** следовать опция RPL Target Descriptor.
2. Когда узел обновляет данные в Transit Information для опции Target, охватывающей один из его адресов, узел **должен** инкрементировать номер Path Sequence в Transit Information. Номер Path Sequence **можно** увеличивать время от времени, чтобы вызвать обновление маршрутов Downward.
3. За опциями RPL Target в индивидуальном сообщении DAO **должна** следовать одна или несколько опций Transit Information, которые применяются к непосредственно предшествующим опциям Target.
4. В групповые DAO **недопустимо** включать поле DODAG Parent Address (в опциях Transit Information).
5. Узел, получивший и обрабатывающий сообщение DAO с информацией для конкретной цели Target, о которой у него есть прежняя информация, **должен** использовать номер Path Sequence из опции Transit Information, связанной с этой целью, для определения наличия в сообщении DAO обновлённой информации в соответствии с разделом 7.
6. При получении сообщения DAO, не соответствующего указанным правилам, узел **должен** отбросить его.

В режиме Non-Storing корень создаёт строгий заголовок strict source по этапам пересылки путём рекурсивного поиска информации о каждом интервале, которая связывает цель Target (адрес или префикс) с транзитным адресом. В некоторых случаях, когда адрес потомка выводится из префикса, который принадлежит или анонсируется родителем, связь родителя с потомком может быть выведена корнем для создания заголовка source routing. В остальных случаях требуется информировать корень связки транзит-Target со стороны доступной цели, для последующего рекурсивного создания заголовка маршрутизации. Адрес, анонсированный как Target в сообщении DAO, **должен** размещаться на том же маршрутизаторе или быть доступен на канале для маршрутизатора, которому принадлежит адрес, указанный в Transit Information. Ниже приведены правила для обеспечения сквозной непрерывности маршрута source-route.

1. Адрес родителя в опции транзита **должен** браться из PIO от родителя с флагом R, указывающим, что поле префикса действительно содержит полный адрес родителя, но не следует считать его относящимся к каналу (on-link).

2. PIO с флагом A указывает, что дочерний узел RPL может использовать префикс для автоматической настройки адресов. Родитель, анонсирующий префикс в PIO с флагом A, **должен** гарантировать, что адрес или весь префикс из PIO доступен из корня путём его анонсирования как цели DAO. Если родитель установил также флаг L, указывающий, что префикс относится к каналу, он **должен** анонсировать весь префикс как Target в сообщении DAO. Если флаг L сброшен, а R установлен, это говорит, что родитель представляет свой адрес в PIO, и тогда этот родитель **должен** анонсировать данный адрес как цель DAO.

3. Адрес, анонсируемый как Target в сообщении DAO, **должен** размещаться на том же маршрутизаторе или быть доступен на канале для маршрутизатора, которому принадлежит адрес, указанный в Transit Information.

4. Для обеспечения возможности оптимального сжатия заголовка маршрутизации родителю **следует** устанавливать флаг R во всех PIO с установленным флагом A и сброшенным флагом L, а потомку **следует** предпочитать использование в качестве транзитного адреса родителя из сообщения PIO, которое служит для автоматической настройки адреса, анонсируемого как Target в сообщении DAO.

5. Маршрутизатор может иметь цели, для которых неизвестно о нахождении на одном канале с родителем, поскольку они имеют адреса, размещённые на другом интерфейсе, или относятся к внешним по отношению к RPL узлам (например, к подключённым хостам). Для внедрения таких целей (Target) в сеть RPL маршрутизатор **должен** анонсировать себя в поле DODAG Parent Address опции Transit Information для этой цели, указывая адрес на канале с узлом родителя DAO. Если Target относится к внешнему узлу, маршрутизатор **должен** установить флаг E в Transit Information.

Дочернему узлу, автоматически настраивающему адрес из родительской опции PIO с флагом L, не нужно анонсировать этот адрес как DAO Target, поскольку родитель гарантирует, что весь префикс доступен из корня. Если флаг L не установлен, тогда в режиме Non-Storing потомок должен информировать корень о связке «родитель-потомок», используя доступный адрес родителя, чтобы обеспечить рекурсивное создание заголовка маршрутизации. Это делается путём связывания адреса родителя (как транзитного) с адресом потомка как Target в сообщении DAO.

9.5. Планирование передачи DAO

Поскольку DAO распространяются вверх (Upward), приём индивидуального сообщения DAO может вызвать передачу индивидуального DAO родителю DAO.

1. При получении индивидуального сообщения DAO с обновлённой информацией (такой как опция Transit Information с новым полем Path Sequence) узлу **следует** передать DAO. Это сообщение DAO **не следует** передавать сразу же, а **следует** задержать передачу DAO для агрегирования сведений DAO от других узлов, для которых этот узел является родителем.

2. Узлу **следует** задержать отправку DAO по таймеру DelayDAO, запускаемому приёмом сообщения DAO. Сообщения DAO, полученные в интервале DelayDAO, не сбрасывают таймер. По завершении отсчёта DelayDAO узел передаёт сообщение DAO.

3. При добавлении узлом другого узла в свой набор родителей DAO ему **следует** запланировать передачу DAO.

Значение и расчёт DelayDAO зависят от реализации. Данная спецификация определяет принятое по умолчанию значение DEFAULT_DAO_DELAY.

9.6. Инициирование сообщений DAO

Узлы могут инициировать отправку сообщений DAO в своих субграфах DODAG. Каждый узел поддерживает порядковый номер триггера DAO (DAO Trigger Sequence Number или DTSN), передаваемый в сообщениях DIO.

1. Если узел видит увеличение DTSN в DAO одного из своих родителей, он **должен** запланировать передачу сообщения DAO в соответствии с правилами параграфов 9.3 и 9.5.

2. В режиме Non-Storing при увеличении DTSN одним из родителей DAO узел **должен** инкрементировать DTSN.

В режиме Storing узел **может** в процессе поддержки и обновления своей таблицы маршрутизации инкрементировать DTSN для надёжного запуска набора обновлений DAO от своих непосредственных потомков. При этом не требуется запускать обновления от всего субграфа DODAG, поскольку сведения поэтапно распространяются вверх по DODAG.

В режиме Non-Storing обновление DTSN будет также заставлять непосредственных потомков узла инкрементировать свои DTSN, запуская набор обновлений DAO от всего субграфа DODAG. Обычно в режиме Non-Storing только корень непосредственно инкрементирует DTSN при необходимость обновления DAO без глобального восстановления (как при инкрементировании DODAGVersionNumber). Обычно в режиме Non-Storing некорневые узлы обновляют DTSN после того, как это сделают их родители.

В общем случае узел может запускать обновления DAO в соответствии с логикой реализации, например при обнаружении несогласованности маршрута Downward или по внутреннему таймеру.

В сетях с хранением выбор подходящего значения DelayDAO для запускаемых DAO может существенно снизить число передаваемых сообщений DAO. Обновления распространяются вниз по DODAG, сообщения DAO в лучшем случае распространяются вверх по DODAG и сообщения DAO передают сначала листья (каждый по одному DAO). Такое планирование можно аппроксимировать устанавливая значение DelayDAO обратно пропорциональное Rank. Это предложение направлено на оптимизацию с эффективным агрегированием и в общем случае не требуется для работы.

9.7. Режим без хранения

В режиме Non-Storing протокол RPL маршрутизирует сообщения в нисходящем направлении с помощью IP source-route. Ниже приведены правила для узлов, работающих в режиме Non-Storing (для режима Storing см. параграф 9.)8.

1. Поле DODAG Parent Address в опции Transit Information **должно** содержать один или несколько адресов, которые **должны** быть адресами родителей DAO для отправителя.

2. Сообщения DAO передаются непосредственно в корень по заданному по умолчанию маршруту, установленному в процессе выбора родителей.

3. При удалении узлом одного из своих родителей он **может** создать сообщение DAO с обновлённой опцией Transit Information.

В режиме Non-Storing узлы используют сообщения DAO для указания своих родителей DAO корню DODAG, который может собирать нисходящий маршрут к узлу, используя наборы родителей DAO от каждого узла на этом маршруте. Данные Path Sequence позволяют обнаруживать устаревшую информацию DAO. Цель такого поэтапного расчёта маршрутов является минимизация трафика при смене родителей DAO. Если узел сообщает полные маршруты source-route, при изменении родителя DAO весь субграф DODAG будет передавать новые DAO корню DODAG. Поэтому в режиме Non-Storing узел может передать одно сообщение DAO, хотя ничто не препятствует отправке по одному DAO каждому из родителей DAO.

Узлы упаковывают DAO, передавая одно сообщение с несколькими опциями RPL Target, за каждой из которых следуют опции Transit Information.

9.8. Режим с хранением

В режиме Storing протокол RPL маршрутизирует сообщения вниз (Downward) по адресу получателя IPv6. Ниже приведены правила для узлов, работающих в режиме с хранением.

1. Поле DODAG Parent Address в опции Transmit Information **должно** быть пустым.

2. При получении индивидуального сообщения DAO узел **должен** вычислить, изменит ли DAO набор префиксов, анонсируемых узлом. В этот расчёт **следует** включать данные Path Sequence из опций Transit Information, связанных с DAO, для проверки наличия в сообщении DAO более свежей информации, заменяющей хранящиеся на узле сведения. При наличии таких данных узел **должен** создать новое сообщение DAO и передать его в соответствии с правилами параграфа 9.5. Такие изменения включают получение No-Path DAO.

3. Узлу, создающему новое сообщение DAO, **следует** передать его индивидуально каждому из своих родителей DAO. Передача индивидуальных сообщений DAO узлам, не являющимся родителями, **недопустима**.

4. При удалении узлом другого узла из числа родителей DAO ему **следует** передать сообщение No-Path DAO (параграф 6.4.3) этому удаляемому родителю DAO для аннулирования имеющегося маршрута.

5. Если сообщения по анонсируемому нисходящему адресу сталкиваются с ошибкой пересылки, NUD или похожим отказом, узел **может** пометить адрес как недоступный и создать сообщение No-Path DAO.

Сообщения DAO анонсируют адреса и префиксы, к которым у узла есть маршруты. В отличие от режима Non-Storing, DAO не содержат сведений о самом маршруте, эта информация хранится в сети и неявно выводится по адресу отправителя IPv6. Когда хранящий узел создаёт DAO, он использует сохранённое состояние полученных DAO для создания набора опций RPL Target и связанных с ними опций Transit[1] Information. Поскольку информация хранится на каждом узле (в таблицах маршрутизации), DAO передаются на прямую хранящим информацию родителям DAO.

9.9. Управление путями

Сообщение DAO от узла содержит одну или несколько опций Target, каждая из которых задаёт префикс, анонсируемый узлом, префикс адреса за пределами LLN, адрес получателя с суб-DODAG узла или multicast-группу, которую слушает узел в суб-DODAG. Поле Path Control в опции Transit Information позволяет узлам запрашивать или разрешать множество маршрутов Downward. Узел создаёт поле Path Control в опции Transit Information, как описано ниже.

1. Размер поля Path Control в битах **должен** составлять (PCS + 1), где значение PCS задано в поле управления опции DODAG Configuration. Биты сверх (PCS + 1) **должны** сбрасываться (0) при передаче и **должны** игнорироваться при получении. Биты в переделах (PCS + 1) считаются активными.

2. Узел **должен** логически группировать своих родителей DAO при заполнении поля Path Control, включая в каждую группу родителей с одинаковым уровнем предпочтения. Группы **должны** быть упорядочены про предпочтительности, что позволит логически сопоставить родителей DAO с субполями Path Control (Рисунок 27). Группы **могут** повторяться для использования всего пространства поля Path Control, но порядок, включая повторяющиеся группы, **должен** сохраняться для корректной передачи предпочтений.

3. Для опции RPL Target, описывающей собственный адрес узла или префикс вне LLN, **должен** быть установлен хотя бы один активный бит поля Path Control и **может** быть установлено большее число активных битов.

4. Если узел получает множество DAO с одинаковой опцией RPL Target, он **должен** использовать побитовую операцию ИЛИ (OR) для полученных полей Path Control, в результате которой будет получено число маршрутов Downward для префикса.

5. При передаче узлом сообщения DAO одному из родителей он **должен** выбрать в субполе один или несколько активных битов, отображённых на группу этого родителя, в поле Path Control. Данный бит может быть активным лишь для одного родителя. Передаваемое этому родителю сообщение DAO **должно** иметь эти активные биты установленные, а прочие активные биты - сброшенными.

6. Для опции RPL Target и номера DAOSequence в DAO узел, передающий сообщения разным родителям DAO, **должен** иметь развязанные (disjoint) наборы активных битов Path Control. Узлу **недопустимо** передавать один и тот же активный бит в сообщениях DAO для разных родителей DAO.

7. Биты Path Control **следует** выделять в соответствии с отображением предпочтительности родителей DAO на субполя Path Control так, чтобы активные биты или группы битов Path Control, относящиеся к конкретному субполю Path Control выделялись родителям DAO из сопоставленной с этим субполем группы.

8. В режиме Non-Storing узел **может** «пропускать» DAO через себя без обработки поля Path Control.

[1] В оригинале ошибочно сказано Transmit, см. https://www.rfc-editor.org/errata/eid3895. *Прим. перев.*

9. Узлу **недопустимо** передавать сообщения DAO без активных битов в поле Path Control. Возможно, что для данной опции Target у узла не будет достаточно числа агрегированных битов Path Control для передачи сообщения DAO с этой опцией Target каждому из родителей DAO, и в этом случае наименее предпочтительные родитель DAO могут не получить сообщение DAO для этой цели Target.

Поле Path Control позволяет узлу ограничить число генерируемых для него маршрутов Downward. Узел устанавливает в Path Control число битов, равное предпочитаемому им числу маршрутов Downward. Каждый бит отправляется единственному родителю DAO и возможна передача одному родителю кластера битов для распределения между его родителями DAO.

Узлу, предоставляющему маршрут DAO для цели Target, которая имеет связанное поле Path Control, **следует** использовать содержимое этого поля Path Control для задания предпочтительности разных маршрутов DAO для Target. Назначение поля Path Control выводится из предпочтений (родителей DAO), определённых на основе знания этим узлом лучшей «сквозной» метрики нисходящих маршрутов в соответствии с предметной функцией OF. В режиме Non-Storing корень может определить маршрут Downward объединяя сведения из каждого полученного DAO, включающие указание в Path Control предпочтительных родителей DAO.

9.9.1. Пример Path Control

Представим сеть LLN в режиме Storing, содержащую узел N с родителями P1, P2, P3, P4 и потомками C1, C2, C3 в его суб-DODAG. Пусть PCS = 7, что указывает 8 активных битов Path Control: 11111111b. Поле Path Control делится на 4 субполя PC1 (11000000b), PC2 (00110000b), PC3 (00001100b), PC4 (00000011b), которые представляют 4 разных уровня предпочтения, как показано на рисунке 27. Реализация на узле N в этом примере создаёт группу {P1, P2} с одинаковым предпочтением, которая предпочтительней {P3}, а {P3} предпочтительней {P4}. Узел N создаёт в Path Control отображение, показанное ниже.

```
{P1, P2} -> PC1 (11000000b) в поле Path Control
{P3}     -> PC2 (00110000b) в поле Path Control
{P4}     -> PC3 (00001100b) в поле Path Control
{P4}     -> PC4 (00000011b) в поле Path Control
```

Отметим повторение {P4} для заполнение поля Path Control.

1. Пусть C1 передаёт DAO для Target T с Path Control 10000000b. N сохраняет запись, связывающую 10000000b с полем Path Control для C1 и Target T.
2. Пусть C2 передаёт DAO для Target T с Path Control 00010000b. N сохраняет запись, связывающую 00010000b с полем Path Control для C2[1] Target T.
3. Пусть C3 передаёт DAO для Target T с Path Control 00001100b. N сохраняет запись, связывающую 00001100b с полем Path Control для C3[1] и Target T.
4. Позднее N генерирует DAO для Target T. N создаёт поле Path Control, объединяя операцией ИЛИ (OR) вклады всех своих потомков из DAO для Target T. В результате поле Path Control имеет активные биты 10011100b.
5. N распространяет биты Path Control своим родителям P1, P2, P3, P4 для подготовки сообщений DAO.
6. P1 и P2 подходят для получения активных битов из наиболее предпочтительного субполя (11000000b) со значением 10000000b в агрегированном поле Path Control. Узел N должен установить бит лишь для одного из двух родителей. В данном случае бит выделен узлу P1 и он получает поле Path Control = 10000000b в DAO. Для узла P2 битов не выделено, он получает Path Control = 00000000b и DAO не может создаваться для P2 по причине отсутствия активных битов.
7. Второе по предпочтительности субполе (00110000b) имеет биты 00010000b. N сопоставляет это поле с P3 и может выделить активный бит узлу P3, создавая для него DAO с Target T и Path Control = 00010000b.
8. Третьим по предпочтительности является субполе (00001100b) с активными битами 00001100b, сопоставленное в N с узлом P4. N может выделить оба бита узлу P4, создав DAO для P4 с Target T и Path Control = 00001100b.
9. Наименее предпочтительное субполе (00000011b) не имеет активных битов. Если бы эти биты были, их следовало бы добавлять в поле Path Control сообщения DAO для P4.
10. Процесс заполнения сообщений DAO для P1, P2, P3, P4 с другими целями (не T) продолжается в соответствии с агрегированными полями Path Control для этих целей.

9.10. Сообщения с анонсами групповых адресатов

Особым случаем работы DAO, отличающимся от передачи индивидуальных DAO, является групповая передача DAO для заполнения таблицы маршрутизации записями с одним интервалом пересылки (1-hop).

1. Узел **может** передать групповое сообщение DAO в локальный канал по групповому адресу all-RPL-nodes.
2. Групповые сообщения DAO **должны** применяться лишь для анонсирования сведений о самом узле, т. е. префиксов, принадлежащих узлу или напрямую соединённых с ним, таких как multicast-группы, на которые узел подписан, или принадлежащий узлу глобальный адрес.
3. Групповые сообщения DAO **недопустимо** использовать для ретрансляции сведений о связности, полученных от другого узла (например, в индивидуальных DAO).
4. Узлу **недопустимо** выполнять другую (относящуюся к DAO) обработку полученного группового сообщения DAO, в частности, **недопустимо** выполнять действия родителя DAO при получении группового DAO.

Групповые сообщения DAO могут применяться для поддержки прямого взаимодействия P2P без использования DODAG для ретрансляции пакетов.

[1] В оригинале ошибочно указано C1, см. https://www.rfc-editor.org/errata/eid3580. *Прим. перев.*

10. Механизмы защиты

В этом разделе описана генерация и обработка защищённых сообщений RPL. Старший бит кода сообщения RPL указывает состояние защищенности сообщения. В дополнение к защищённым вариантам базовых управляющих сообщений (DIS, DIO, DAO, DAO-ACK) RPL включает несколько сообщений, которые актуальны лишь в сетях со включённой защитой.

Сложность реализации и размер важны для сетей LLN поэтому включение в реализацию RPL изощрённых механизмов защиты может быть экономически или физически невозможно. Кроме того, во многих сетях может применяться защита на канальном уровне или иные механизмы, позволяющие обеспечить безопасность без использования защиты в RPL.

Поэтому описанные здесь функции защиты **необязательны** для реализации. Конкретная реализация **может** поддерживать часть описанных механизмов, например, защиту целостности и конфиденциальности без подписей. Реализациям **следует** чётко указывает поддерживаемые механизмы защиты и **рекомендуется** внимательно рассмотреть требования безопасности и имеющиеся в сети механизмы защиты.

10.1. Обзор защиты

RPL поддерживает три режима защиты, описанных ниже.

Unsecured - без защиты

В этом режиме RPL использует базовые сообщения DIS, DIO, DAO, DAO-ACK без разделов Security. Поскольку в сети могут применяться иные методы защиты (например, защита на канальном уровне), использование этого режима не означает полного отсутствия защиты сообщений.

Preinstalled - с предустановленным ключом

В этом режиме применяются защищённые варианты сообщений RPL. Для присоединения к RPL Instance узел должен иметь заранее установленный ключ, применяемый для защиты конфиденциальности и целостности, а также для проверки подлинности сообщений. Используя предустановленный ключ, узел может присоединиться к сети RPL в качестве хоста или маршрутизатора.

Authenticated - с проверкой подлинности

В этом режиме применяются защищённые варианты сообщений RPL. Для присоединения к RPL Instance узел должен иметь заранее установленный ключ, применяемый для защиты конфиденциальности и целостности, а также для проверки подлинности сообщений. Используя предустановленный ключ, узел может присоединиться к сети RPL лишь в качестве хоста. Для подключения как маршрутизатора узлу требуется второй ключ.

Этот режим не поддерживает симметричные алгоритмы шифрования. На момент создания этого документа протокол RPL поддерживал лишь симметричные алгоритмы и данный режим включён с учётом возможной в будущем поддержки других криптографических примитивов (см. 10.3. Установка ключей).

Независимо от использования защиты экземпляром RPL он указывает применение защищённых сообщений RPL с помощью бита A в опции DAG Configuration.

Данная спецификация задаёт режим CCM (Counter с цепочками шифрованных блоков и кодом аутентификации сообщений) в качестве криптографической основы защиты RPL [RFC3610]. В этой спецификации CCM использует алгоритм шифрования AES-128. В разделе Security зарезервированы биты для задания в будущем иных алгоритмов.

Все защищённые сообщения RPL включают подпись или код MAC, а также могут использовать шифрование. Формат защищённых сообщений RPL поддерживает встроенное шифрование/подписи (CCM), а также внешние схемы шифрования и аутентификации пакетов.

10.2. Присоединение к защищённой сети

Защита RPL предполагает, что желающий присоединиться к защищённой сети узел имеет заранее созданный общий ключ для коммуникаций с соседями и корнем RPL. Для присоединения к защищённой сети RPL узел прослушивает защищённые DIO или инициирует их, отправляя защищённое сообщение DIS. В дополнение к правилам для DIO и DIS из раздела 8, защищённые сообщения DIO и DIS следуют приведённым ниже правилам.

1. При отправке начального защищённого сообщения DIS в поле Key Identifier Mode **должно** устанавливаться значение 0 (00), а в Security Level **должно** устанавливаться значение 1 (001). Используемый ключ **должен** быть заранее заданным ключом группы (Key Index 0x00).

2. При сбросе узлом таймера Trickle в ответ на защищённое сообщение DIS (параграф 8.3) следующее передаваемое им сообщение DIO **должно** быть защищённым DIO с такими же настройками защиты как в DIS. Если узел получает множество защищённых DIS до того, как передаст DIO, защищённое сообщение DIO **должно** иметь такие же параметры защиты, как последнее сообщение DIS, на которое узел отвечает.

3. При передаче узлом DIO в ответ на индивидуальное защищённое сообщение DIS (параграф 8.3), сообщение DIO **должно** быть защищённым.

Приведённые выше правила позволяют узлу присоединиться к защищённому экземпляру RPL с использованием заранее настроенного общего ключа. Когда узел присоединён с помощью такого ключа к DODAG, его возможности определяет бит A в опции Configuration. При сброшенном флаге A узел может использовать предустановленный общий ключ обычным способом и может вводить сообщения DIO, DAO и т. п. Если бит A в опции Configuration установлен и RPL Instance работает в режиме authenticated, выполняются приведённые ниже правила.

1. Узлу **недопустимо** анонсировать ранг, отличающийся от INFINITE_RANK, в DIO, защищённых с Key Index 0x00. При обработке DIO, защищённых с Key Index 0x00, обрабатывающий узел **должен** считать, что Rank = INFINITE_RANK. Все прочие значения ведут к отбрасыванию сообщения.

2. Защищённым с Key Index 0x00 сообщениям DAO **недопустимо** иметь опцию RPL Target с префиксом, отличающимся от адреса узла. Если узел получает сообщение DAO, защищённое с использованием предустановленного общего ключа, где опция RPL Target не совпадает с адресом отправителя IPv6, он **должен** отбросить защищённое сообщение DAO без дальнейшей обработки.

Приведённые выше правила означают, что для экземпляров RPL с установленным битом A, использующих Key Index 0x00, узел может присоединиться к RPL Instance как хост, но не маршрутизатор. Узел должен взаимодействовать с удостоверяющим центром для получения возможности действовать как маршрутизатор.

10.3. Установка ключей

Режим с аутентификацией требует, чтобы потенциальный маршрутизатор динамически устанавливал новые ключи после присоединения к сети в качестве хоста. Подключившись таким способом, узел использует стандартные сообщения IP для взаимодействия с сервером проверки полномочий, который может предоставить новые ключи.

Протокол получения ключей выходит за рамки этой спецификации и будет задан в будущих спецификациях. Эта доработка нужна RPL для защищённой работы в режиме с проверкой подлинности.

10.4. Проверка согласованности

Узлы RPL передают сообщения CC для синхронизации счётчиков и защиты от атак с повтором сообщений.

1. Если узел получает индивидуальное сообщение CC со сброшенным битом R входит или находится в процессе присоединения к связанному графу DODAG, ему **следует** отвечать отправителю индивидуальным сообщением CC. Отклик **должен** иметь установленный флаг R, а также **должен** включать значения полей CC nonce, RPLInstanceID и DODAGID из полученного сообщения.
2. При получении группового сообщения CC узел **должен** отбросить его без обработки.

Сообщения CC позволяют узлам использовать обмен «запрос-отклик» для проверки текущего значения счётчика на узле. Поскольку CC nonce создаётся запрашивающей стороной, повторно использующий сообщения злоумышленник вряд ли сможет создать корректный отклик. Счётчик в отклике CC позволяет запрашивающему проверить слышимые им значения счётчика.

10.5. Счётчики

В простейшем случае значение счётчика является целым числом без знака, которое узел увеличивает на 1 или больше при каждой защищённой транзакции RPL. Счётчик **может** быть временной меткой с указанными ниже свойствами.

1. Временная метка **должна** быть размером не менее 6 октетов.
2. Временная метка **должна** иметь точность 1024 Гц (двоичная миллисекунда).
3. Началом отсчёта временных меток **должно** быть 1 января 1970 г, 12:00:00AM UTC.
4. Если счётчик представляет временную метку, его значение **должно** вычисляться в соответствии с приведённым далее правилом. Пусть T обозначает метку, S - время начала использования ключа, E - время завершения использования ключа. Значения S и E представляются по трём описанным выше правилам. Если E > T < S, счётчик является недействительным и узлу **недопустимо** создавать пакет. В ином случае значение счётчика равно T-S.
5. Если счётчик представляет такую временную метку, узел **может** установить флаг T в разделе Security защищённых пакетов RPL.
6. Если поле Counter не представляет такую метку, узлу **недопустимо** устанавливать флаг T.
7. Если узел не имеет локальной временной метки, соответствующей приведённым выше требованиям, он **должен** игнорировать флаг T.

Если узел поддерживает такие временные метки и получает сообщение с установленным флагом T, он **может** использовать проверку полученных сообщений по времени, как описано в параграфе 10.7.1. Если в сообщении не установлен флаг T, проверка по времени **недопустима**. Правила безопасности узла в соответствии с требованиями приложений **могут** отвергать сообщения, в которых флаг T не установлен.

Наличие флага T обусловлено тем, что уже сегодня многие сети LLN поддерживают глобальную синхронизацию часов с точностью в доли миллисекунд в соответствии с потребностями защиты, приложений и по иным причинам. Возможность использовать имеющуюся функциональность в RPL значительно упрощает решение некоторых задач безопасности, таких как защиты от излишних задержек.

10.6. Исходящие пакеты

В этом параграфе рассматривается генерация защищённых пакетов для передачи с учётом типа исходящего пакета управления и требуемой защиты. Рассматривается также порядок выполнения криптографических операций.

Требования к защите и её уровню для исходящих пакетов RPL определяются базой данных узла о правилах защиты, конфигурация которой зависит от реализации. При передаче защищённых сообщений узел RPL **должен** включать раздел Security (T, Sec, KIM, LVL) в исходящие пакеты RPL для описания уровня и параметров применяемой защиты (параграф 6.1). Флаг Security в поле RPL Message Code **должен** быть установлен в защищённых сообщениях RPL.

Значение счётчика, применяемое при создании AES-128 CCM nonce (Рисунок 31) для защиты исходящих пакетов, **должно** увеличиваться по сравнению с последним значением, переданным по конкретному адресу получателя.

Если правила безопасности требуют применения защиты от задержки, счётчик Timestamp, используемый при создании CCM nonce для защиты исходящих пакетов, **должен** инкрементироваться в соответствии с правилами параграфа 10.5. Если применяется счётчик Timestamp (установлен флаг T), значение поддерживаемого локально счётчика Timestamp **должно** включаться как часть передаваемого защищённого сообщения RPL.

Используемый для защиты исходящих пакетов криптографический алгоритм, нужно задавать в базе правил безопасности узла и он **должен** указываться в значении поля Sec исходящего сообщения.

Правила защиты исходящих пакетов должны определять KIM и Key Identifier, задающие ключ, используемый для криптографической защиты, включая необязательное использование ключей подписи (параграф 6.1). Правила защиты могут также задавать алгоритм (Algorithm) и уровень (Level) защиты в форме аутентификации или аутентификации и шифрования, а также возможное использование подписей в исходящих пакетах.

При использовании шифрования узел **должен** заменить исходное содержимое пакета зашифрованными данными, используя параметры, указанные в разделе Security данного пакета.

Все защищённые сообщения RPL включают контроль целостности. В процессе применения алгоритма защиты узел выводит значение MAC или подписи, которое **должно** быть частью исходящего пакета RPL.

10.7. Входящие пакеты

В этом параграфе рассматривается приём и обработка защищённых пакетов RPL. С учётом флага Security в поле RPL Message Code входящего защищённого пакета описана расшифровка пакета RPL и проверка его целостности.

Получатель использует поля управления защитой RPL для определения требуемых при обработке защищённого пакета операций. Если уровень защиты для типа сообщения и его отправителя неизвестен или не соответствует поддерживаемым локально правилам защиты, узел **должен** отбросить пакет без обработки и **может** выдать сигнал для системы управления, но передача ответного сообщения **недопустима**. Правила могут включать уровни защиты, применяемые ключи, идентификаторы отправителей или отсутствие счётчиков на основе временных меток (флаг T). Конфигурация базы правил защиты для обработки входящих пакетов выходит за рамки спецификации (она может быть задана, например, через опции DIO Configuration или отдельной административной настройкой маршрутизаторов).

Если Security Level (LVL) в сообщении RPL указывает шифрование, узел использует сведения о ключе из поля KIM, а также CCM nonce в качестве входных данных для процесса расшифровки содержимого сообщения. Значение CCM nonce нужно выводить поля Counter в сообщении и других принятых и поддерживаемых локально сведений (см. параграф 10.9.1). Содержимое расшифрованного сообщения извлекается путём вызова режима, обратного криптографической операции, заданной полем Sec в принятом пакете.

Получателю нужно использовать CCM nonce и указанные сведения о ключе для проверки целостности входящего пакета. Если рассчитанное значение не совпадает с полученным MAC, узел **должен** отбросить пакет.

Если полученное сообщение содержит неинициализированное (0) значение счётчика, а у получателя имеется счётчик входящих сообщений от инициатора сообщения, получатель **должен** инициировать повторную синхронизацию счётчиков путём передачи отклика CC (параграф 6.6), который должен быть защищён с использованием полного текущего значения счётчика, поддерживаемого для конкретного узла. Значение исходящего счётчика включается в раздел Security, а значение входящего - в данные сообщения CC.

В соответствии с заданной политикой безопасности узел **может** применять защиту от повторного использования для полученных сообщений RPL. Проверку **следует** выполнять до аутентификации полученного пакета. Счётчик из входящего пакета нужно сравнивать с «водяным знаком» счётчика входящих сообщений для адреса источника сообщения. Если значение счётчика принятых сообщений отлично от нуля и меньше «водяного знака», это говорит о возможном повторе старого сообщения (replay) и узел **должен** отбросить входящий пакет.

Если указана защита от задержки как часть правил, применяется счётчик Timestamp для контроля своевременности полученного сообщения RPL. Если значение Timestamp в принятом сообщении указывает время передачи до поддерживаемого локально времени для инициатора сообщения, это считается попыткой повторного использования и узел **должен** отбросить пакет. Если принятое значение Timestamp указывает время передачи раньше разности текущего времени за вычетом допустимой задержки при передаче, это считается нарушением задержки и узел **должен** отбросить пакет.

После расшифровки сообщения (когда это применимо) и прохождения проверки целостности, а также, возможно, проверки задержки узел может обновить локальные данные защиты, такие как ожидаемое значение счётчика для отправителя, используемое для определения фактов повторного использования (replay). Узлу **недопустимо** обновлять данные защиты в результате приёма сообщения, которое не прошло проверку выполнения правил защиты, проверку целостности, повторного использования или защиты от задержки.

10.7.1. Проверка временной метки ключа

Если в сообщении установлен флаг T и узел имеет локальную временную метку, соответствующую требованиям параграфа 10.5, он **может** проверить сообщение на соответствие времени. Узел рассчитывает время отправки сообщения, добавляя значение счётчика к моменту начала использования соответствующего ключа. Если время отправки оказывается позже срока завершения действия ключа, узел **может** отбросить сообщение без обработки. Если время отправки сообщения существенно раньше или позже локального времени принимающего узла, узел **может** отбросить сообщение без обработки.

10.8. Область защиты целостности и конфиденциальности

Для сообщений RPL ICMPv6 защита RPL целиком охватывает пакет.

Коды MAC и подписи рассчитываются для всего незащищённого пакета IPv6, при этом изменяемые поля IPv6 считаются заполненными нулями в соответствии с правилами 3.3.3.1 [RFC4302] (IPsec AH). Расчёт MAC и подписи происходит до сжатия, которое может применяться нижележащими уровнями.

Шифрование сообщения RPL ICMPv6 начинается с первого байта после раздела Security и продолжается до последнего байта пакета. Заголовки IPv6 и ICMPv6, а также часть сообщения RPL до конца раздела Security не шифруются, поскольку они нужны для корректной расшифровки пакета.

Например, узел, передающий сообщение с LVL=1, KIM=0 и Algorithm=0, использует алгоритм CCM [RFC3610] для создания пакета с атрибутами ENC-MAC-32 - пакет шифруется и к нему добавляется 32-битовый код MAC. Ключ блочного шифра определяет Key Index. Значение CCM nonce рассчитывается в соответствии с параграфом 10.9.1, шифруемым и аутентифицируемым сообщением является часть сообщения RPL с первого байта после раздела

Security и до конца пакета. Дополнительные данные проверки подлинности начинаются с заголовка IPv6 и завершаются последним байтом раздела RPL Security.

10.9. Криптографический режим работы

Криптографический режим, описанный в этой спецификации (Algorithm = 0), основан на CCM и блочном шифре AES-128 [RFC3610]. Режим широко поддерживается имеющимися реализациями. Для режима CCM требуется CCM nonce.

10.9.1. CCM Nonce

Узел RPL создаёт CCM nonce с показанным на рисунке 31 форматом.

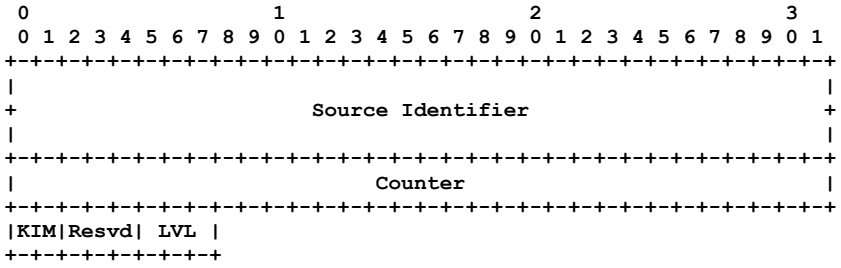


Рисунок 31. CCM Nonce.

Source Identifier

8-битовый логический идентификатор отправителя защищённого пакета.

Counter

4-битовое поле, содержащее (несжатое) значение соответствующего поля в опции Security управляющего сообщения RPL.

Key Identifier Mode (KIM)

2-битовое поле, содержащее значение соответствующего поля в опции Security управляющего сообщения RPL.

Security Level (LVL)

3-битовое поле, содержащее значение соответствующего поля в опции Security управляющего сообщения RPL.

Не выделенные биты CCM nonce являются резервными. Отправитель **должен** сбрасывать их в 0, а получатель **должен** игнорировать.

Все поля CCM nonce представляются с порядком битов и байтов от старшего к младшему.

10.9.2. Подписи

Если KIM (3) указывает применение подписи, узел добавляет к содержимому пакета подпись, размер которой определяет поле Security Level (LVL). Схема подписи в RPL для режима защиты 3 использует экземпляр алгоритма RSA (RSASSA-PSS), как задано в параграфе 8.1 [RFC3447]. Открытым ключом служит пара (n,e), где n - 2048- 3072-битовый модуль RSA, а e=2^{16}+1. В качестве схемы шифрования применяется режим CCM [RFC3610] с M=0 (как потоковый шифр). Отметим, что [RFC3610] не разрешает режим CCM с M=0, однако в RPL явно разрешён такой режим при использовании с подписью, поскольку та обеспечивает достаточную аутентификацию данных. Здесь режим CCM с M=0 применяется как указано в [RFC3610], но поле M' в параграфе 2.2 **должно** иметь значение 0. Применяется хэш-функция SHA-256, заданная в параграфе 6.2 [FIPS180], с правилами кодирования из параграфа 8.1 в [RFC3447].

Пусть a будет конкатенацией 6-байтового представления счётчика и заголовка сообщения. Содержимое пакета представляет собой конкатенацию данных m и подписи s (справа). Подпись создаётся для конкатенации частей a и m (справа), а при проверке используется конкатенация a и m, а также подпись s.

Эта форма подписей RSA обеспечивает достаточную защиту для сетей RPL. При необходимости могут применяться более сжатые схемы, которые выходят за рамки этой спецификации но могут быть предложены в будущем.

Реализации, поддерживающей подписи RSA размером 2048 или 3072 бита, **следует** поддерживать проверку для обоих размеров подписи RSA (2048 и 3072). Это позволит обновлять развёртывание RPL.

11. Пересылка пакетов, обнаружение и предотвращение петель

11.1. Предложения для пересылки пакетов

Этот документ задаёт протокол маршрутизации. Ниже представлены не являющиеся нормативными предложения, способные помочь при разработке реализации пересылки, которая сможет работать с RPL. При пересылке пакетов адресату выбор преемника (next-hop) происходит в указанном порядке.

1. Эта спецификация описывает лишь выбор преемника из DODAG Version, соответствующей RPLInstanceID, отмеченному в заголовке IPv6 пересылаемого пакета. Маршрутизация вне экземпляра может быть выполнена путём установки дополнительных правил, таких как строгое упорядочение экземпляров и протоколов маршрутизации для защиты от петель. Такие правила могут быть заданы отдельным документом.

2. Если локальные административные предпочтения задают выбор маршрута, полученного от другого (не RPL) протокола, используется этот преемник.

3. Если заголовок пакета содержит заданный источником маршрут путём включения заголовка RH4 в соответствии с [RFC6554], применяется этот маршрут. Если узел не может переслать пакет по заданному отправителем маршруту, пакет следует отбросить. Узел **может** записать это в системный журнал и может передать ошибку ICMPv6 в сообщении Source Routing Header для отправителя пакета (см. параграф 20.18).

4. Если в таблице маршрутов имеется соответствующая адресату запись, полученная из анонса multicast-получателя (например, получателем является one-hop-сосед), используется этот преемник.

5. Если в таблице маршрутизации для адресата имеется запись, полученная из анонса индивидуального получателя (например, он расположен ниже в субграфе DODAG), используется этот преемник. При наличии битов DAO Path Control, связанный с несколькими преемниками эти биты служат для упорядочения преемников при выборе. Если для данного бита DAO Path Control имеется несколько преемников, предпочтение отдаётся подтвердившему этот бит последним.

6. При наличии версии DODAG, предлагающей маршрут к префиксу, соответствующему адресату, в качестве преемника выбирается один из родителей DODAG в соответствии с предметной функцией OF и метрикой.

7. Можно выбрать ранее не использовавшегося родителя DODAG при следующей попытке переслать индивидуальный пакет, если нет более подходящего совпадения.

8. Если преемник не найден, пакет отбрасывается и **может** быть передано сообщение ICMP Destination Unreachable (обнаружено несоответствия).

При пересылке значение Hop Limit **должно** быть уменьшено в соответствии с [RFC2460].

Недопустимо выбирать преемником соседа, который был для пакета предшественником (расщепление горизонта), за исключением случаев, когда для пакета нужно изменить направление (Upward на Downward), как указано в таблице маршрутизации меняющего направление узла, например, при переключении с маршрута DIO на DAO по мере приближения к адресату.

11.2. Обнаружение и предотвращение петель

Механизмы предотвращения петель в RPL достаточно просты и предназначены для минимизации «мешанины» (churn) и числа состояний. Петли могут возникать по разным причинам, например, при потере пакетов управления. RPL включает реактивный метод обнаружения петель, предотвращающий зацикливание и восстанавливающий оборванные пути.

Для обнаружения петель RPL использует сведения о пакете (RPL Packet Information), передаваемые в пакетах данных на основе внешних механизмов, таких как [RFC6553], которые помещают RPL Packet Information в заголовок опции IPv6 Hop-by-Hop. Содержимое RPL Packet Information описано ниже.

Down 'O'
Флаг, указывающий продвижение пакета вверх (Up) или вниз (Down). Маршрутизатор устанавливает флаг O, когда предполагается перемещение пакета вниз (по маршрутам DAO), и сбрасывает его для пакетов, пересылаемых в направлении корня DODAG (к узлу с меньшим рангом). Хост или лист RPL **должен** сбрасывать (0) флаг O.

Rank-Error 'R'
Флаг, указывающий обнаружение ошибки ранга (Rank). Ошибкой считается несоответствие относительных рангов и направления, указанного битом O. Хост или лист RPL **должен** сбрасывать (0) флаг R.

Forwarding-Error 'F'
Флаг невозможности пересылки узлом пакета в направлении адресата. Флаг может устанавливать дочерний узел, у которого нет маршрута к получателю пакета с флагом O. Хост или лист RPL **должен** сбрасывать (0) флаг F.

RPLInstanceID
8-битовое поле, указывающее экземпляр DODAG, по которому передаётся пакет.

SenderRank
16-битовое поле, где отправитель устанавливает 0, а маршрутизатор, пересылающий пакет в сети RPL, - DAGRank(rank).

11.2.1. Работа узла-источника

Если источнику известно предпочтительное значение RPLInstanceID для пакета, он **должен** установить его в связанном с пакетом поле RPLInstanceID. В противном случае **должно** устанавливаться значение RPL_DEFAULT_INSTANCE.

11.2.2. Работа маршрутизатора

11.2.2.1. Пересылка пакетов экземпляром

Отправитель связывает с пакетом значение RPLInstanceID, которое **должно** соответствовать экземпляру RPL, куда пакет помещается узлом (хостом или маршрутизатором). RPLInstanceID является частью RPL Packet Information.

Маршрутизатор RPL, пересылающий пакет в сеть RPL, **должен** проверить наличие в пакете RPL Packet Information и **должен** вставить RPL Packet Information, если этих данных нет. Если маршрутизатор является входом в сеть RPL, он **должен** установить поле RPLInstanceID в RPL Packet Information. Детали выбора маршрутизатором значения RPLInstanceID выходят за рамки этой спецификации и оставлены на будущее.

Маршрутизатор, пересылающий пакет за пределы сети RPL, **должен** удалить RPL Packet Information.

Когда маршрутизатор получает пакет с RPLInstanceID и может переслать пакет по связанному с этим экземпляром графу DODAG, маршрутизатор **должен** переслать пакет, не меняя RPLInstanceID. Если узел не может переслать пакет по графу DODAG, связанному с RPLInstanceID, ему **следует** отбросить пакет и передать сообщение ICMP об ошибке.

11.2.2.2. Обнаружение петель несогласованности DAG

Граф DODAG не согласован, если направление пакета не соответствует соотношению Rank. Несоответствием считается получение пакета с установленным битом O (Down) от узла с большим Rank или со сброшенным флагом O (Up) от узла с меньшим рангом.

Когда корень DODAG инкрементирует DODAGVersionNumber, может образоваться временный разрыв Rank между следующей и прежней версией DODAG, в частности, если узлы корректируют свой ранг в следующей DODAG Version и откладывают переход к новой версии DODAG. Маршрутизатор, остающийся в прежней DODAG Version, может переслать пакет (будущему) родителю, который относится к новой версии DODAG. В некоторых случаях это может приводить к обнаружению родителем несогласованности, поскольку ранжирование в прежней DODAG Version не обязательно совпадает с текущей версией DODAG и пакет может быть сочтён не продвигающимся вперёд. Если передающий маршрутизатор знает, что выбранный потомок уже присоединён к новой версии DODAG, он **должен**

установить SenderRank = INFINITE_RANK, поскольку пересылает пакет через разрыв в следующую версию DODAG, чтобы предотвратить ложное детектирование несогласованности рангов.

Одна несогласованность на пути не считается критической ошибкой и обработка пакета может продолжаться. Однако обнаружение второй несогласованности на пути того же пакета является ошибкой и пакет **должен** быть отброшен. Этот процесс управляется битом Rank-Error, связанным с пакетом. При обнаружении несогласованности для пакета, где бит Rank-Error не установлен, флаг Rank-Error устанавливается, а если он уже установлен, пакет **должен** отбрасываться, а таймер Trickle **должен** сбрасываться.

11.2.2.3. Обнаружение и исправление несогласованности DAO

Механизм устранения несогласованности DAO применяется только в режиме Storing. В режиме Non-Storing маршрут пакетов задаётся отправителем и несогласованность DAO не корректируется локально. Вместо этого в корень передаётся сообщение ICMP с новым кодом ошибки в заголовке заданной источником маршрутизации (Error in Source Routing Header), которое имеет тот же формат, что и сообщение Destination Unreachable, определённое в [RFC4443]. Возвращаемая часть вызвавшего сообщение ICMP пакета должна включать содержимое пакета по меньшей мере до заголовка маршрутизации, а сам этот заголовок должен использоваться узлом, чтобы получателем в заголовке IPv6 был следующий интервал, который оказался недоступным для этого узла.

Несогласованность DAO возникает, когда у маршрутизатора имеется нисходящий маршрут, полученный ранее из сообщения DAO от потомка, но ставший недействительным у этого потомка, например, в результате сброса связанного состояния у потомка. При устранении петли несогласованности DAO пакет может служить для рекурсивного нахождения и сброса устаревших состояний DAO в субграфе DODAG.

В общем случае направленный вниз пакет никогда не должен поворачивать вверх. При устранении несогласованности DAO маршрутизатору **следует** возвращать пакет передавшему его родителю с установленным флагом F и неизменным флагом O. В иных случаях маршрутизатор **должен** просто отбрасывать пакет.

При получении пакета с установленным флагом F узел **должен** удалить состояния маршрутизации, вызвавшие пересылку этому соседу, сбросить бит F и попытаться передать пакет снова. Пакет может быть передан другому соседу по настраиваемому пользователем и определяемому реализацией таймеру. Если для этого соседа также возникает несогласованность DAO, процесс будет рекурсивным - этот узел установит флаг F и состояние маршрутизации в нем будет сброшено.

12. Групповой трафик

В этом разделе описана групповая маршрутизация в сети IPv6 RPL и, в частности, применение индивидуальных DAO для ретрансляции регистрации групп. Маршрутизация индивидуального и группового трафика может применять общий граф DODAG. Рассматривается способ расширения регистрации групп и работа инфраструктуры пересылки без полного описания группового трафика в LLN и без описания построения DODAG специально для групповой пересылки и детали групповой адресации в RPL. Это будет предметом других спецификаций.

При регистрации групп применяются сообщения DAO, которые отличаются от индивидуальных лишь типом передаваемого адреса. Нисходящий групповой трафик копируется для всех зарегистрированных потомков.

Узлам, поддерживающим режим RPL Storing, **следует** поддерживать групповые операции DAO, как описано ниже. От узлов, поддерживающих лишь режим Non-Storing, такая поддержка не ожидается. Групповые операции управляются полем MOP в DIO.

- Если поле MOP требует поддержки групповых операций, узел, присоединяющийся к сети RPL как маршрутизатор, должен выполнять описанные в этом разделе операции групповой сигнализации и пересылки в сети RPL. Узел, не поддерживаемых групповых операций, заданных полем MOP, может быть только листом.
- Если поле MOP не требует поддержки групповых операций, групповая обработка обеспечивается другим способом, не задаваемым данной спецификацией (например, копиями или ограниченной лавинной рассылкой).

Маршрутизатор может передать сообщение DAO о регистрации слушателя лишь его предпочтительному родителю и в этом случае возвращающиеся групповые пакеты могут быть потеряны для всех суб-DODAG, если при передаче по каналу произойдёт отказ. В дополнение маршрутизатор может выбрать копирование дополнительных родителей, как это делается для сообщений DAO, анонсирующих индивидуальных получателей. В этом случае могут появляться дубликаты, которые маршрутизатору потребуется удалять. В результате состояния групповой маршрутизации устанавливаются в каждом маршрутизаторе на пути от слушателей к корню DODAG, что позволяет корню копировать групповые пакеты всем дочерним маршрутизаторам, передавшим сообщение DAO с опцией Target для этой группы.

Групповые пакеты, исходящие из DODAG, передаются предпочтительным родителям, а при возникновении отказа - дополнительным родителям в DODAG. Пакеты также копируются для всех зарегистрированных потомков, за исключением передавшего пакет. При наличии слушателей во внешней инфраструктуре корень DODAG дополнительно отправляет пакеты в эту инфраструктуру.

В результате корень DODAG служит автоматическим посредником Rendezvous Point (точка встречи) для сети RPL и источником по отношению к внешнему (не RPL) домену для всех групповых потоков, начинающихся в домене RPL. Поэтому независимо от подключения корня к внешнему (не RPL) домену и независимо от приземленности DODAG корень может всегда обслуживать внутренние групповые потоки.

13. Поддержка маршрутной смежности

Выбор преемников на принятых по умолчанию восходящих путях по DODAG или путях, полученных из анонсов нисходящих маршрутов по DODAG ведёт к формированию маршрутной смежности, которую требуется поддерживать.

В протоколах IGP, таких как OSPF [RFC4915] или IS-IS [RFC5120], поддержка маршрутной смежности включает использование механизмов keepalive (Hello) или других протоколов, таких как двухстороннее обнаружение пересылки (Bidirectional Forwarding Detection или BFD) [RFC5881] и обнаружение соседства в MANET (MANET Neighborhood

Discovery Protocol или NHDP) [RFC6130]. К сожалению такой проактивный подход часто нежелателен в средах с ограничениями, где он будет создавать избыточный трафик, негативно влияющий на загрузку каналов и ресурсы узлов.

В отличие от таких протоколов, RPL не применяет механизмов keepalive для обнаружения отказов маршрутной смежности, поскольку такие механизмы в большинстве случаев слишком накладны в плане пропускной способности и, ещё важнее, потребляемой мощности (устройства с батарейным питанием не могут позволить себе периодическую отправку сообщений keepalive). Тем не менее, для RPL нужны внешние механизмы обнаружения недоступности соседей. Для таких механизмов предпочтительна реакция на трафик, чтобы минимизировать издержки на поддержание маршрутной смежности сосредоточится на реально используемых каналах. Примерами реактивных механизмов могут быть обнаружение недоступности соседей (Neighbor Unreachability Detection) [RFC4861] и триггеры канального уровня [RFC5184] на основе таких событий, как состояния ассоциаций и подтверждения L2.

14. Рекомендации для предметных функций

Предметная функция (OF) вкупе с метрикой маршрутизации и ограничениями позволяет выбрать граф DODAG для присоединения и несколько партнёров из этого DODAG в качестве родителей. OF служит для расчёта упорядоченного списка родителей, а также отвечает за расчёт ранга устройства внутри DODAG Version.

Функция OF указывается в сообщении DIO с использованием кода OCP и задаёт метод, который должен применяться для создания графа DODAG. Коды OCP заданы в [RFC6552] и сопровождающих спецификациях.

14.1. Поведение предметной функции

От большинства предметных функций OF ожидается описанное ниже поведение на узлах.

- Выбор родителя запускается всякий раз, когда событие указывает обновление сведений о потенциальном next-hop. Это может быть результатом приёма сообщения DIO, завершением отсчёта таймера, недоступности всех родителей DODAG или срабатывания триггера, указывающего смену состояния кандидата в соседи.
- OF сканирует все интерфейсы узла. Хотя во многих случаях приложения имеют лишь один интерфейс, их может оказаться несколько и работа с RPL на интерфейсе может быть отключена. Для интерфейса также может быть настроено предпочтение или динамическое определение как лучшего на основе той или иной эвристики, которая может зависеть от канального уровня и выходит за рамки спецификации. Кроме того, интерфейс может, но не обязан соответствовать критериям для OF, например, по уровню защиты. В результате некоторые интерфейсы могут полностью исключаться из расчёта (например, при несоответствии анонсированным ограничениям), тогда как другие могут быть более или менее предпочтительными.
- OF сканирует всех кандидатов в соседи на возможных интерфейсах для проверки их возможности работы в качестве маршрутизатора для DODAG. Кандидатов может быть много и от них может потребоваться прохождение проверки перед использованием. В частности, некоторые канальные уровни требуют «опыта» работы с маршрутизатором для его включения в качестве следующего интервала пересылки.
- OF рассчитывает Rank узла для сравнения путём добавления к рангу кандидата значения, представляющего относительное положение узла и кандидата в DODAG Version.
  - Увеличение ранга должно быть не меньше MinHopRankIncrease.
  - Чтобы избежать петель и сохранить оптимизацию метрики, увеличение ранга должно отражать любой рост значений метрики. Например, при сугубо аддитивной метрике, такой как ETX, рост Rank может быть пропорционален увеличению метрики.
  - Кандидаты в соседи, вызывающие рост Rank для узла, не учитываются при выборе родителей.
- Кандидаты в соседи, анонсирующие функцию OF, несовместимую с набором OF, указанных функциями политики, игнорируются.
- При сканировании всех кандидатов в соседи OF сохраняет текущего лучшего родителя и сравнивает его возможности с текущим кандидатом в соседи. OF определяет число тестов, которые имеют решающее значение для достижения цели. Тесты для маршрутизаторов определяют их упорядочение:
  - если результаты для маршрутизаторов совпадают, для них выполняется следующий тест;
  - в ином случае лучший из маршрутизаторов становится текущим лучшим родителем и сканирование переходит к следующему кандидату в соседи;
  - некоторые OF могут включать сравнение рангов, которые выполняются при соединении узла с любым из маршрутизаторов.
- По завершении сканирования выбирается предпочтительный родитель и рассчитывается ранг узла, как Rank предпочтительного родителя плюс приращение ранга с этим родителем.
- Для выбора дополнительных родителей могут потребоваться дополнительные раунды сканирования:
  - кандидаты в соседи из других DODAG игнорируются;
  - кандидаты в соседи, чей Rank больше ранга узла, игнорируются;
  - кандидаты в соседи, чей Rank равен рангу узла, игнорируются при выборе родителей;
  - кандидаты в соседи, чей Rank меньше ранга узла, являются предпочтительными.

15.Предложения по взаимодействию с ND

Эта спецификация заимствует опции информации о префиксе (Prefix Information Option или PIO) и маршруте (Route Information Option или RIO) напрямую из IPv6 ND. Предполагается, что в будущих спецификациях, основанных на этой,

могут возникнуть дополнительные причины использования IPv6 ND. В этом разделе приведены некоторые предложения для таких спецификаций.

RPL в первую очередь является протоколом маршрутизации. При сопоставлении функциональности RPL и ND следует соблюдать осторожность в целях сохранения архитектуры. Протокол RPL предназначен лишь для маршрутизации, но, тем не менее, могут быть технические причины совместно использования опций в RPL и IPv6 ND для конкретной реализации или развёртывания. В общем случае применимы приведённые ниже рекомендации.

- Коды RPL Type должны выделяться из реестра RPL Control Message Options.
- Поля RPL Length должны указывать размер в октетах в отличие от ND Length с 8-октетными словами.
- Опции RPL обычно не требуют выравнивания по 8-октетным границам.
- При отображении или транспонировании опции IPv6 ND для распространения в качестве опции RPL октеты заполнения следует по возможности удалять. Например, поля Prefix Length в PIO достаточно для указания размера поля Prefix. При отображении или транспонировании опции RPL для распространения как опции IPv6 ND октеты заполнения следует восстанавливать. Эта процедура должна быть однозначной.

16. Требования к взаимодействию
В этом разделе приведено базовое описание совместимости и ссылки на нормативные документы для реализаций RPL, работающих в одном из трёх основных режимов. Предполагается, что реализация поддерживает работу без маршрутов Downward, только режим Non-Storing или только режим Storing. Возможна также работа в качестве листа.

Реализации данной спецификации могут включать разные наборы возможностей в зависимости от применения. Для разработчиков важно поддерживать уровень взаимодействия, требуемый для варианта применения. С этой целью могут быть даны дополнительные рекомендации (например, заявления о применимости), которые в отдельных случаях могут переопределять данную спецификацию.

16.1. Общие требования
В общем случае максимальный уровень взаимодействия может быть достигнут при использовании всеми узлами RPL LLN общих MOP, OF, метрики и ограничений, обеспечивающих узлам возможность служить маршрутизаторами RPL. Если узел не может служить маршрутизатором RPL, возможно ограниченное взаимодействие в качестве листа RPL.

Все реализации RPL должны поддерживать использование опций RPL Packet Information, передаваемых в пакетах данных (параграф 11.2). Один из таких механизмов описан в [RFC6553].

Реализации RPL должны поддерживать обнаружение недоступности соседей (Neighbor Unreachability Detection или NUD) или эквивалентный механизм для контроля доступности соседних узлов RPL (параграф 8.2.1). Могут оптимизироваться дополнительные механизмы для реализаций с ограниченными возможностями, такие как рекомендации с канального уровня.

Данная спецификация предоставляет способы получения PIO и формирования таким образом адреса IPv6. При использовании этого механизма может потребоваться распознавание адресов и обнаружение дубликатов с помощью внешних процессов, таких как IPv6 ND [RFC4861] или 6LoWPAN ND [6LOWPAN-ND].

16.2. Работа в качестве листа RPL
- Реализация узла, являющегося лишь листом, не служит маршрутизатором RPL. Поведение совместимых реализаций листьев описано в параграфе 8.5. Работа листа.
- Поддержка конкретного кодирования MOP не требуется, хотя при передаче сообщений DAO для организации нисходящих маршрутов листу следует делать это в соответствии с режимом работы, указанным MOP.
- Поддержка конкретной функции OF не требуется.
- Лист обычно не выдаёт сообщений DIO и может выдавать сообщения DAO и DIS. Узел воспринимает сообщения DIO, хотя обычно игнорирует сообщения DAO и DIS.

16.3. Работа в качестве маршрутизатора RPL
При отсутствии дополнительных рекомендаций реализация маршрутизатора RPL **должна** поддерживать предметную функцию OF0 без метрики [RFC6552].

Для согласованной работы реализация маршрутизатора RPL должна поддерживать MOP, используемый в DODAG.

Все маршрутизаторы RPL должны реализовать механизм Trickle [RFC6206].

16.3.1. Поддержка лишь восходящих маршрутов
Требования к реализации маршрутизатора RPL, поддерживающего лишь восходящие маршруты, приведены ниже.

- Поддержка маршрутов Upward (8. Восходящие маршруты).
- Поддержка MOP = 0 (20.3. Новый реестр для режимов работы (MOP)).
- Выдаются сообщения DIO и DIS, но не DAO. Сообщения DIO и DIS воспринимаются, DAO игнорируются.

16.3.2. Поддержка маршрутов Upward и Downward в режиме Non-Storing
Требования к реализации маршрутизатора RPL, поддерживающего восходящие и нисходящие маршруты в режиме Non-Storing, приведены ниже.

- Поддержка маршрутов Upward (8. Восходящие маршруты).

- Поддержка маршрутов Downward в режиме Non-Storing (9. Нисходящие маршруты).
- Поддержка MOP = 1 (20.3. Новый реестр для режимов работы (MOP)).
- Заданные отправителем маршруты для трафика Downward ([RFC6554]).
- Выдаются сообщения DIO и DIS, а также сообщения DAO для корня DODAG. Сообщения DIO и DIS воспринимаются, а сообщения DAO игнорируются узлами, не служащими корнем DODAG. Групповой трафик не поддерживается описанными этой спецификацией способами, но может поддерживаться иным путём.

16.3.3. Поддержка маршрутов Upward и Downward в режиме Storing

Требования к реализации маршрутизатора RPL, поддерживающего восходящие и нисходящие маршруты в режиме Storing, приведены ниже.

- Поддержка маршрутов Upward (8. Восходящие маршруты).
- Поддержка маршрутов Downward в режиме Non-Storing (9. Нисходящие маршруты).
- Поддержка MOP = 2 (20.3. Новый реестр для режимов работы (MOP)).
- Выдаются и воспринимаются сообщения DIO, DIS, DAO. Групповой трафик не поддерживается описанными этой спецификацией способами, но может поддерживаться иным путём.

16.3.3.1. Необязательная поддержка Basic Multicast Scheme

В режиме Storing может быть реализована базовая поддержка группового трафика:

- Basic Multicast Support (12. Групповой трафик).
- MOP = 3 (20.3. Новый реестр для режимов работы (MOP))

16.4. Вопросы для будущих спецификаций

Ниже перечислен ряд вопросов, оставленных для будущих спецификаций.

- Подключение других (не RPL) узлов, таких как хосты IPv6, например, для согласованного распространения им по меньшей мере данных PIO.
- Получение данных аутентификации для поддержки режима с проверкой подлинности (10.3. Установка ключей).
- Детали одновременной работы с несколькими экземплярами.
- Расширенные механизмы настройки, такие как предоставление RPLInstanceID, параметризация функций OF, параметры управления защитой. Предполагается расширение сообщений DIO для этих механизмом как средства распространения через DODAG.

17. Константы и переменные RPL

BASE_RANK = 0
Ранг виртуального корня, который может служить для координации множества корней.

ROOT_RANK
Ранг для корня DODAG, имеющий значение MinHopRankIncrease (анонсированное корнем DODAG), поэтому DAGRank(ROOT_RANK) = 1.

INFINITE_RANK = 0xFFFF
Константа, ограничивающая максимальное значение Rank.

RPL_DEFAULT_INSTANCE = 0
Значение RPLInstanceID, применяемое протоколом на узлах без политики переопределения.

DEFAULT_PATH_CONTROL_SIZE = 0
Принятое по умолчанию значение для поля PCS в опции DODAG Configuration, указывающего число значимых битов поля Path Control в опции Transit Information. Это задаёт простейший случай, ограничивая разветвление до 1 и позволяя узлу отправлять сообщения DAO лишь одному родителю.

DEFAULT_DIO_INTERVAL_MIN = 3
Принятое по умолчанию значение, используемое при настройке Imin для таймера DIO Trickle. Это значение задаёт Imin = 8 мсек.

DEFAULT_DIO_INTERVAL_DOUBLINGS = 20
Принятое по умолчанию значение, используемое при настройке Imax для таймера DIO Trickle. Это значение задаёт максимальный интервал 2,3 часа.

DEFAULT_DIO_REDUNDANCY_CONSTANT = 10
Принятое по умолчанию значение, используемое при настройке k для таймера DIO Trickle. Это значение задаёт консервативный механизм подавления Trickle.

DEFAULT_MIN_HOP_RANK_INCREASE = 256
Принятое по умолчанию значение MinHopRankIncrease. Это задаёт 8-битовую целую часть Rank.

DEFAULT_DAO_DELAY = 1 секунда
Принятое по умолчанию значение для таймера DelayDAO (9.5. Планирование передачи DAO).

Таймер DIO
Один экземпляр на граф DODAG, в который узел входит. Завершение отсчёта вызывает передачу сообщения DIO. Таймер Trickle имеет переменный интервал [0, DIOIntervalMin..$2^{\text{DIOIntervalDoublings}}$] (8.3.1. Параметры Trickle).

таймер увеличения версии DAG
До одного экземпляра на граф DODAG, в котором узел служит корнем DODAG. Может не поддерживаться в некоторых реализациях. Завершение отсчёта вызывает увеличение DODAGVersionNumber, приводящее к отправке новой серии обновлённых DIO. Интервал следует выбирать в соответствии со временем распространения DODAG и требованиями приложения (например, время отклика в сравнении с издержками).

Таймер DelayDAO

До одного таймера на родителя DAO (подмножество родителей DODAG, выбранных для получения анонсов адресатов) в DODAG. Завершение отсчёта вызывает отправку DAO родителю DAO (см. параграф 9.5).

RemoveTimer

До одного таймера на запись DAO для соседа, передавшего данное сообщение DAO этому узлу как родителю DODAG. Завершение отсчёта вызывает анонс No-Path или немедленное удаление записи DAO, если нет родителей DAO.

18. Вопросы управляемости

В этом разделе рассматриваются вопросы управляемости RPL и работы протокола RPL в сети LLN, включая настройку, мониторинг, контроль отказов, учёт и производительность протокола в свете рекомендаций [RFC5706].

18.1. Введение

Большинство имеющихся стандартов IETF для управления - это модули MIB (модели данных на основе SMI[1]) для мониторинга и управления сетевыми устройствами. Для многих протоколов сообщество IETF использовало стандартную схему управления IETF (Standard Management Framework), включающую простой протокол управления сетью (Simple Network Management Protocol или SNMP) [RFC3410], SMI [RFC2578] и модели данных MIB для управления новыми протоколами.

Как указано в [RFC5706], общая политика в части операций и управления была преобразована в политику, более открытую для набора инструментов и протоколов управления, а не полагающуюся на единственный протокол, такой как SNMP. В 2003 Совет по архитектуре Internet (Internet Architecture Board или IAB) провёл семинар по управлению сетью [RFC3535], где обсуждались сильные и слабые стороны некоторых протоколов IETF для управления сетями и применительно к операционным потребностям, особенно к настройке. Одной из рассмотренных проблем является неудобство для пользователей двоичного формата SNMP [RFC3410]. В случае LLN следует отметить, что на момент подготовки документов рабочая группа CoRE активно занималась вопросами управления ресурсами устройств в LLN. Тем не менее, считается, что этот раздел содержит важные рекомендации в части развёртывания., эксплуатации и управления RPL. В [RFC5706] отмечено:

> В модель данных управления следует включать обсуждение того, что является управляемым, какие аспекты протокола нужно настраивать, какие типы операций разрешены, какие специфические для протокола события могут произойти, какие события можно учесть, а о каких следует уведомлять оператора.

Эти вопросы подробно рассматриваются в последующих параграфах.

RPL будет применяться на разных устройствах, размер памяти в которых может составлять от нескольких килобайт до сотен килобайт и даже мегабайт. При жёстко ограниченной памяти невозможно выполнить все требования, указанные в этом разделе. Тем не менее, следует перечислить все требования, чтобы разработчики могли выбрать из них выполнимые в имеющимися ресурсами.

18.2. Управление конфигурацией

В этом параграфе рассматривается управление конфигурацией и приведены параметры протокола, которые можно настроить. Некоторые параметры RPL необязательны. Требования по настройке относятся лишь к используемым опциям.

18.2.1. Режим инициализации

В параграфе 3.8 «Architectural Principles of the Internet» [RFC1958] сказано: «По возможности избегайте опций и параметров. Любые опции и параметры следует делать настраиваемыми или согласуемыми динамически, а не вручную». Это особенно важно в LLN, где число устройств может быть большим и настройка вручную нежелательна. Это было учтено при разработке RPL с помощью предоставления корнем DODAG множества параметров для устройств, присоединяющихся к DODAG, что позволяет избавиться от громоздкой настройки маршрутизаторов и возможных ошибок в конфигурации (например, в значениях таймеров Trickle и т. п.). Тем не менее, имеются параметры RPL, для которых реализации следует обеспечивать настройку, как описано ниже.

18.2.1.1. Режим работы DIS при загрузке

При первом включении узла возможны два описанных ниже варианта действий.

1. Узел может «хранить молчание», ожидая сообщений DIO от интересующего графа DODAG (анонсы поддерживаемых OF, метрики, ограничений) и не передавая групповых DIO до вхождения в DODAG.
2. Узел может передать одно или несколько сообщений DIS (возможно с запросом DIO для конкретного DODAG) в качестве начального зондирования ближайших DODAG и при отсутствии ответных сообщений DIO в течение настраиваемого интервала может принять себя роль корня плавающего DODAG и начать отправку групповых сообщения DIO.

Реализации RPL **следует** разрешать настройку предпочтительного режима из числа перечисленных выше с указанием требуемых параметров (для второго варианта - число сообщений DIS и значение таймера ожидания).

18.2.2. Базовые сообщения DIO и DAO, настройка опций

Протокол RPL задаёт множество параметров с учётом широкого спектра приложений, где применяется RPL. Особое внимание уделено ограничению числа параметров, которые нужно настраивать на каждом маршрутизаторе RPL. Вместо настройки можно применять установленные по умолчанию значения, которые при необходимости может динамически изменить корень DODAG. Реализациям RPL **следует** разрешать настройку описанных ниже параметров. Как уже отмечено, множество параметров задаёт корень DODAG.

[1] Structure of Management Information - структура управляющей информации.

18.2.3. Параметры протокола, настраиваемые на каждом маршрутизаторе в LLN

Реализация RPL **должна** обеспечивать возможность перечисленных ниже параметров RPL.

- RPLInstanceID [сообщение DIO, в DIO Base]. Хотя значение RPLInstanceID должен задавать корень DODAG, оно может определяться правилами каждого узла для решения вопроса о присоединении узла к конкретному графу DODAG. На узле может быть задано второе значение RPLInstanceID, если он становится корнем плавающего DODAG.
- Список поддерживаемых кодов OCP.
- Список поддерживаемых метрик. В [RFC6551] задано множество метрик и ограничений, применяемых при формировании DODAG. Поэтому реализация RPL должна разрешать настройку списка метрик, которые узел может воспринимать и понимать. При получении DIO и непонятной или неподдерживаемой метрикой или ограничением, как указано в параграфе 8.5, узел будет подключаться в качестве листа.
- Информация о префиксе с действительным и предпочтительным сроком действия, а также флаги L и A. [сообщения DIO, опция PIO]. Реализации RPL **следует** разрешать настройку, если опция Prefix Information должна передаваться с сообщением DIO для распространения сведений о префиксе с целью автоматической настройки. В этом случае реализация RPL **должна** разрешать анонсирование списка префиксов в PIO с соответствующими флагами.
- Запрашиваемая информация [сообщение DIS, опция Solicited Information]. Реализации RPL **следует** разрешать настройку условий, когда такие сообщения следует передавать вместе с RPLInstanceID и флагами V, I, D.
- Условия установки флага K в сообщении DAO [сообщение DAO, в DAO Base].
- Режим MOP[сообщение DIO, в DIO Base].
- Route Information (и предпочтение) [сообщение DIO, опция Route Information].

18.2.4. Параметры настройки маршрутизаторов, не являющихся корнем DODAG

Реализация RPL **должна** разрешать настройку префикса Target [сообщение DAO, опция RPL Target].

Кроме того, при некоторых обстоятельствах узел может пожелать обозначить Target для выполнения специальной обработки (например, приоритизация), правила которой выходят за рамки спецификации. В таких случаях реализации RPL **следует** разрешать настройку Target Descriptor на уровне цели (например, с помощью списков доступа).

Узел с пустым набором родителей DODAG может стать корнем плавающего DODAG, а также установить DAGPreference, чтобы быть менее предпочтительным. Поэтому реализация RPL **должна** разрешать указание действий, которые узлу следует инициировать в таком случае:

- запуск своего (плавающего) DODAG с возможностью настройки в дополнение к DAGPreference;
- «порча» оборванных путей (8.2.2.5. «Порча» маршрутов);
- инициирование локального восстановления.

18.2.5. Параметры для настройки в DODAG Root

Некоторые параметры настраиваются лишь в корне DODAG и анонсируются в опциях сообщений DIO. Как указано в параграфе 8.3, реализация RPL применяет таймеры Trickle для управления отправкой DIO. Работа алгоритма Trickle определяется набором параметров, которые **должны** быть настраиваемыми и анонсируются корнем DODAG по графу DODAG в сообщениях DIO:

- DIOIntervalDoublings [сообщение DIO, опция DODAG Configuration];
- DIOIntervalMin [сообщение DIO, опция DODAG Configuration];
- DIORedundancyConstant [сообщение DIO, опция DODAG Configuration].

Кроме того, реализации RPL **следует** разрешать настройку параметров RPL:

- Path Control Size [сообщение DIO, опция DODAG Configuration];
- MinHopRankIncrease [сообщение DIO, опция DODAG Configuration];
- поле DODAGPreference [сообщение DIO, объект DIO Base];
- DODAGID [сообщение DIO, опция DIO Base] и [сообщение DAO при установленном флаге D].

Поведение корня DAG. В некоторых случаях для узла может быть нежелательна постоянная работа в качестве корня плавающего DODAG, если он не может присоединиться к приземлённому DODAG. Например, узел с батарейным питанием может не пожелать долго служить корнем плавающего DODAG. Поэтому реализация RPL **может** поддерживать настройку времени работы в качестве корня плавающего DODAG.

Увеличение DAG Version Number. Реализация RPL может разрешать (через настройку в корне DODAG) обновление состояний DODAG путём изменения DODAGVersionNumber. Реализации RPL **следует** разрешать настройку периодического или вызываемого событиями механизма, позволяющего корню DODAG управлять сменой DODAGVersionNumber (запускающей глобальное восстановление, как описано в параграфе 3.2.2).

18.2.6. Параметры настройки RPL для механизмов на основе DAO

Сообщения DAO не обязательны и применяются графами DODAG, которым нужна маршрутизация в нисходящем направлении. В этом разделе рассматривается набор параметров, относящихся к сообщениям DAO, и приведены рекомендации по их настройке.

Как указано в параграфе 9.5, рекомендуется задерживать передачу DAO родителям DAO для повышения шансов агрегирования маршрутов. При получении сообщения DAO узлу следует запускать таймер DelayDAO, для которого по умолчанию установлено значение DEFAULT_DAO_DELAY. Реализация RPL **может** разрешать настройку DelayDAO.

В режиме Storing сохраняющий узел может увеличивать номер DTSN для надёжного вызова обновлений DAO от непосредственного потомка как части поддержки и обновления таблицы маршрутизации. Реализация RPL **может** разрешать настройку набора правил, задающих триггеры для инкрементирования DTSN (вручную или по событиям).

При завершении срока действия или аннулировании записи DAO узлу **следует** предпринимать разумные попытки передать No-Path каждому из родителей DAO. Число таких попыток **может** быть настраиваемым.

Реализации следует поддерживать ограничение частоты отправки сообщений DAO. Параметры ограничения **могут** быть настраиваемыми.

18.2.7. Настройка параметров RPL, связанных с защитой

Как указано в разделе 10, защитные свойства, описанные в этом документе, не обязательны и реализация может поддерживать часть описанных функций защиты и отказаться от них совсем. Реализация, поддерживающая функции защиты, может поддерживать базу правил защиты. Для поддержки механизмов защиты реализации RPL **следует** поддерживать настройку подмножества перечисленных ниже параметров:

- воспринимаемые режимы защиты [Unsecured, Preinstalled, Authenticated];
- воспринимаемые значения KIM [управляющие сообщения Secure RPL, раздел Security];
- воспринимаемые значения Level [управляющие сообщения Secure RPL, раздел Security];
- воспринимаемые значения Algorithm [управляющие сообщения Secure RPL, раздел Security];
- ключевой материал для поддержки режимов Authenticated и Preinstalled.

Кроме того, реализации RPL **следует** разрешать настройку в корне DODAG подмножества параметров:

- воспринимаемые значения KIM [сообщения Secure DIO, раздел Security];
- воспринимаемые значения KIM [сообщения Secure DIO, раздел Security];
- воспринимаемые значения Algorithm [сообщения Secure DIO, раздел Security].

18.2.8. Заданные по умолчанию значения

Этот документ задаёт принятые по умолчанию значения для перечисленных ниже переменных RPL:

DEFAULT_PATH_CONTROL_SIZE;
DEFAULT_DIO_INTERVAL_MIN;
DEFAULT_DIO_INTERVAL_DOUBLINGS;
DEFAULT_DIO_REDUNDANCY_CONSTANT;
DEFAULT_MIN_HOP_RANK_INCREASE;
DEFAULT_DAO_DELAY.

В протоколах рекомендуется указывать принятые по умолчанию значения, при этом, как отмечено в [RFC5706], такие значения имеют все меньше смысла. RPL является протоколом маршрутизации, применение которого предполагается в контексте, где характеристики сети, такие как число узлов и каналов, типы узлов, могут существенно меняться. Таким образом, принятые по умолчанию значения будут, вероятно, меняться в зависимости от контекста и развития технологий. Действительно, связанные с LLN технологии (например, оборудование, канальные уровни) за последние несколько лет существенно изменились и предполагается значительное развитие технологий в будущем.

Предлагаемые решения не основаны на значительной практике и их следует считать осторожными (консервативными).

18.3. Отслеживание работы RPL

Некоторые параметры RPL нужно отслеживать для контроля корректности работы протокола маршрутизации и самой сети. В этом параграфе рассматривается набор отслеживаемых параметров.

18.3.1. Параметры DODAG

Реализации RPL **следует** предоставлять информацию о перечисленных ниже параметрах

- DODAG Version [сообщение DIO, в DIO Base].
- Флаг G [сообщение DIO, в DIO Base].
- Поле MOP [сообщение DIO, в DIO Base].
- Значение DTSN [сообщение DIO, в DIO Base].
- Значение Rank [сообщение DIO, в DIO Base].
- Номер DAOSequence, увеличиваемый в каждом уникальном сообщении DAO и возвращаемый в DAO-ACK [DAO и DAO-ACK].
- Route Information [сообщение DIO, опция Route Information] (список префиксов IPv6 для родителя со сроком действия и предпочтением].
- Параметры Trickle:
  - DIOIntervalDoublings [сообщение DIO, в опции DODAG Configuration];
  - DIOIntervalMin [сообщение DIO, в опции DODAG Configuration];

- DIORedundancyConstant [сообщение DIO, в опции DODAG Configuration];
- Path Control Size [сообщение DIO, в опции DODAG Configuration];
- MinHopRankIncrease [сообщение DIO, в опции DODAG Configuration].

Некоторые значения могут отслеживаться лишь в корне DODAG.

- Transit Information [DAO, опция Transit Information]. Реализации RPL **следует** разрешать настройку отображения набора полученных опций Transit Information в корне DODAG. При включённом отображении в базу данных RPL с полученными Transit Information следует также включать Path Sequence, Path Control, Path Lifetime и Parent Address.

18.3.2. Отслеживание несогласованности DODAG и обнаружение петель

Обнаружение несогласованности DODAG имеет особую важность в сетях RPL, поэтому реализациям RPL рекомендуется поддерживать соответствующие средства мониторинга. Реализации RPL **следует** поддерживать счётчик, указывающий число случаев обнаружения узлом несогласованности по отношению к родителю DODAG, например, смены DODAGID.

По возможности следует предоставлять более детальные сведение о нахождении несогласованности. Реализация RPL **может** поддерживать счётчики, указывающие число обнаруженных несоответствий:

- пакеты с установленным флагом O (Down) от узла с более высоким рангом;
- пакеты со сброшенным флагом O (Up) от узла с меньшим рангом;
- пакеты с установленным флагом F;
- пакеты с установленным флагом R.

18.4. Отслеживание структур данных

18.4.1. Структура данных кандидатов в соседи

Список кандидатов в соседи содержит узлы, обнаруженные одним способом и пригодные стать родительскими (с достаточно высокой локальной вероятностью). Реализации RPL **следует** поддерживать средства отслеживания списка кандидатов в соседи с той или иной метрикой, отражающей локальную достоверность (степень стабильности соседства). Реализация RPL **может** поддерживать счётчик числа фактов игнорирования кандидата в соседи при превышении максимального числа кандидатов.

18.4.2. Таблица DODAG

Для каждого графа DODAG от реализации RPL ожидается отслеживание указанных ниже значений в таблице DODAG:

- RPLInstanceID;
- DODAGID;
- DODAGVersionNumber;
- Rank;
- OCP;
- набор родителей DODAG;
- набор префиксов, предлагаемых вверх по графу DODAG;
- таймеры Trickle, используемые при отправке сообщений DIO для DODAG;
- список родителей DAO;
- DTSN;
- статус узла (маршрутизатор или лист).

Реализации RPL **следует** разрешать отслеживание перечисленных выше параметров.

18.4.3. Таблица маршрутизации и маршрутные записи DAO

Реализация RPL поддерживает несколько информационных элементов, относящихся к DODAG и записям DAO (для узлов с хранением). В случае узлов без хранения поддерживается ограниченный объем сведений (таблица маршрутизации сводится в основном к набору родителей DODAG с упомянутыми выше характеристиками DODAG), а для узлов с хранением добавляются маршрутные записи.

Реализации RPL **следует** обеспечивать возможность мониторинга:

- Next Hop (родитель DODAG);
- интерфейс Next Hop;
- метрика пути для каждого родителя DODAG.

Запись в таблице маршрутизации DAO концептуально содержит (только для узлов с хранением):

- информацию об анонсирующем соседе;
- адрес IPv6;
- идентификатор интерфейса, на который была отправлена эта запись родителей DAO;

- счётчик повторов;
- логический эквивалент содержимого DAO:
  - DAO-Sequence;
  - Path Sequence;
  - DAO Lifetime;
  - DAO Path Control;
- префикс адресата (или адрес multicast-группы).

Реализации RPL **следует** предоставлять сведения о состоянии каждой записи в DAO Routing Table.

18.5. Обработка отказов

Контроль отказов является важной частью поиска неисправностей, проверки корректности работы протокола и организации сети, а также её мониторинга. Реализации RPL **следует** обеспечивать указанные ниже сведения:

- переполнение памяти с указанием причины (например, переполнение таблиц маршрутизации);
- число случаев невозможности передачи пакета родителю DODAG, указанному действующим;
- число случаев получения пакетов для которых у маршрутизатора нет соответствующего RPLInstanceID;
- число вызовов процедуры локального восстановления;
- число вызовов процедуры глобального восстановления корнем DODAG;
- число полученных сообщений с некорректным форматом;
- число секунд, когда были пакеты для пересылки и отсутствовал next hop (родитель DODAG);
- число секунд, когда отсутствовал next hop (родитель DODAG);
- число случаев присоединения узла к DODAG в качестве листа в результате получения DIO с непонятной метрикой или ограничением, когда это задано в конфигурации (18.6. Правила).

Рекомендуется при возникновении отказов информировать о них по меньшей мере записью в системный журнал. Для информирования можно применять другие протоколы.

18.6. Правила

Реализация RPL может использовать правила для определения возможности присоединения узла к конкретному графу DODAG, анонсированному соседом в сообщениях DIO.

Этот документ описывает работу в одном графе DODAG, характеризуемом парой (RPLInstanceID, DODAGID). Как было отмечено выше, сообщения DIO содержат анонсы других характеристик DODAG, таких как метрика маршрутов и ограничения, служащие для создания DODAG, и используемая предметная функция OF (задаётся в OCP).

Первые правила политик задают условия, которым должен удовлетворять узел RPL для присоединения к DODAG:

- RPLInstanceID;
- список поддерживаемых метрик маршрутов и ограничений;
- предметная функция OF (значения OCP).

Реализация RPL **должна разрешать настройку этих параметров, а также следует** указывать, должен ли узел просто игнорировать DIO при несоответствии DODAG локальной политике или присоединяться в качестве листа, если не поддерживается лишь список поддерживаемых метрик и ограничений, а также OF. Кроме того, реализации RPL **следует** разрешать добавление DODAGID как части политики.

Реализации RPL **следует** разрешать настройку набора воспринимаемых и предпочтительных функций OF, указываемых кодами OCP для узла при соединении с DODAG и действия, которые следует предпринимать, если ни один из кандидатов в соседи не предлагает ни одной из приемлемых функций OF или анонсированные метрики и ограничения не поддерживаются или непонятны. Здесь возможны два варианта:

- узел присоединяется к DODAG как лист (8.5. Работа листа);
- узел не присоединяется к DODAG.

Узел LLN может получать маршрутную информацию от разных протоколов маршрутизации, включая RPL. В этом случае желательно контролировать выбор предпочтительного маршрута административными средствами. Реализации **следует** разрешать задание административных предпочтений для протокола маршрутизации, из которого получен маршрут.

Внутренние структуры данных. Некоторые реализации RPL могут ограничивать размер списка кандидатов в соседи и в этом случае некоторые подходящие кандидаты могут не рассматриваться или исключаться из списка. Реализация RPL **может** указывать размер списка кандидатов в соседи.

18.7. Изоляция отказов

Рекомендуется помещать в «карантин» соседей, начинающих слишком часто передавать искажённые сообщения.

18.8. Влияние на другие протоколы

Влияние RPL на другие протоколы очень ограничено. Если маршрутизатору (например, LBR) нужны несколько протоколов маршрутизации, предполагается, что он будет поддерживать функции распространения информации между протоколами для обеспечения доступности между разными доменами. Такое распространение **следует** регулировать настраиваемой политикой.

В части влияния на сетевой трафик протокол RPL был разработан с учётом ограничения трафика управления простыми механизмами, такими как таймеры Trickle (8.3. Передача DIO). Поэтому влияние RPL на другие протоколы очень ограничено.

18.9. Управление производительностью

Контроль производительности всегда важен для протоколов и RPL не является исключением. Рабочая группа по мониторингу производительности IP (IP Performance Monitoring или IPPM) отметила несколько представляющих интерес параметров, однако они вряд ли будут применены в LLN с учётом ресурсов устройств и требуемой пропускной способности. Тем не менее, реализации RPL **могут** поддерживать некоторые из параметров, перечисленных ниже:

- число операций восстановления и время выполнения одной операции в секундах (среднее, вариации);
- число случаев и продолжительность времени, когда устройства не могли пересылать пакеты по причине отсутствия доступного соседа в таблице маршрутизации;
- отслеживание расхода ресурсов протоколом RPL (пропускная способность и память);
- число переданных и принятых управляющих сообщений RPL.

18.10. Диагностика

Возможны ситуации, когда узел следует переводить в режим подробного вывода (verbose) для улучшения диагностики. Поэтому реализации RPL **следует** поддерживать для узлов возможность управления режимом вывода диагностики.

19. Вопросы безопасности

19.1. Обзор

С точки зрения безопасности сети RPL ничем не отличаются от других сетей. Они уязвимы для атак с пассивным перехватом данных и, возможно, даже с активным вмешательством, когда для участия в обмене данными не требуется физический доступ к кабелю. Сама природа сетей ad hoc и их стоимость накладывают дополнительные ограничения на защиту, делая эти сети наиболее сложными в плане безопасности. Устройства недороги и имеют ограниченные возможности в части вычислительной мощности, доступного хранилища и потребляемой энергии. Не всегда можно надеяться на наличие в них доверенной вычислительной среды или качественного генератора случайных чисел.

Коммуникации не могут полагаться на постоянную доступность стационарной инфраструктуры и могут включать краткосрочные ассоциации между устройствами, которые могли не взаимодействовать ранее. Эти ограничения могут существенно снижать набор криптографических алгоритмов и протоколов, а также влиять на архитектуру защиты, поскольку организация и поддержка ассоциация между устройствами требует осторожности. Кроме того, срок работы и стоимость батарей могут вносить существенные ограничения на ресурсы, которые могут быть выделены для защиты. Большинство элементов архитектуры защиты могут реализоваться на более высоких уровнях, которые выходят за рамки этой спецификации. Следует проявлять осторожность в отношении интерфейсов с вышележащими уровнями.

Механизмы защиты в этом стандарте основаны на криптографии с симметричными и открытыми ключами, получаемые от протоколов вышележащего уровня. Организация и поддержка ключей выходят за рамки документа. Механизмы предполагают защищённую реализацию криптографических операций и доверенное хранение ключевого материала.

Ниже перечислены механизмы защиты, обеспечивающие безопасность протокола.

Конфиденциальность данных
Сокрытие передаваемой информации от сторон, которым она не предназначена.
Подлинность данных
Подтверждение подлинности источника информации (и неизменности данных в процессе передачи).
Защита от повторного использования
Гарантированное обнаружение дубликатов переданной информации.
Своевременность (защита от задержек)
Обеспечение своевременной доставки переданной информации.

Предоставляемую реально защиту можно настроить для каждого пакета, варьируя уровень контроля подлинности (минимизация издержек), а также защиту конфиденциальности. Может также предоставляться защита от повторного использования переданных ранее пакетов (replay), обеспечиваемая использованием не повторяющихся значений (CCM nonce) в пакетах и хранения сведений о состоянии (устройство-источник и счётчик CCM nonce от этого устройства), что позволяет обнаружить ранее использованные CCM nonce от того же устройства. Обеспечивается также защита от задержек среди устройств, на которых применяются слабо синхронизированные часы. Приемлемую задержку можно задавать на уровне пакета, что позволяет варьировать задержки с учётом пути передачи.

Для криптографической защиты может применяться общий ключ пары партнёров (ключ канала) или группы устройств (групповой ключ), что обеспечивает некоторую гибкость и определяемый приложением компромисс между обеспечиваемым уровнем защиты и расходами на хранение и поддержку ключей. При использовании группового ключа для связи между двумя партнёрами обеспечивается защита лишь от внешних устройств, поскольку другие устройства той же группы знают ключ.

Контроль подлинности данных может быть основан на симметричной криптографии или открытых ключей. При использовании открытых ключей (подписи) подтверждается достоверность отправителя, а симметричные ключи подтверждают лишь принадлежность отправителя к группе с общим ключом. Поэтому аутентификация на основе открытых ключей может быть полезна в случаях, когда требуется более чёткий контроль подлинности, а симметричные

ключи удобны для групповых или широковещательных коммуникаций или в случаях, когда не требуется обеспечивать неотказуемость.

20. Взаимодействие с IANA

20.1. Сообщение RPL Control

Сообщение RPL Control является информационным сообщением ICMP, используемым для передачи сообщений DIO, DIS и DAO при работе RPL. IANA поддерживает реестр ICMPv6 Type Number, где для сообщений RPL Control выделено значение 155.

20.2. Новый реестр для кодов RPL Control

Агентство IANA создало реестр RPL Control Codes для поля Code в сообщениях ICMPv6 RPL Control. Новые коды выделяются по процедуре IETF Review и в каждой записи указывается код, описание и RFC с определением. Определённые в настоящее время коды приведены ниже.

Коды RPL Control.

Код	***Описание***	***Документ***
0x00	DODAG Information Solicitation	Данный документ
0x01	DODAG Information Object	Данный документ
0x02	Destination Advertisement Object	Данный документ
0x03	Destination Advertisement Object	Данный документ
0x80	Secure DODAG Information Solicitation	Данный документ
0x81	Secure DODAG Information Object	Данный документ
0x82	Secure Destination Advertisement Object	Данный документ
0x83	Secure Destination Advertisement Object	Данный документ
0x8A	Consistency Check	Данный документ

20.3. Новый реестр для режимов работы (MOP)

Агентство IANA создало реестр 3-битовых кодов Mode of Operation (MOP), указываемых в DIO Base. Новые коды выделяются по процедуре IETF Review и в каждой записи указывается режим работы, описание возможности, RFC с определением. Включённые в реестр 4 значения приведены ниже.

Режимы работы DIO.

MOP	***Описание***	***Документ***
0	RPL не поддерживает маршрутов Downward	Данный документ
1	Режим Non-Storing	Данный документ
2	Режим Storing без поддержки группового трафика	Данный документ
3	Режим Storing с поддержкой группового трафика	Данный документ

Значения 4 - 7 пока не выделены.

20.4. Опции сообщения RPL Control

Агентство IANA создало реестр RPL Control Message Options. Новые значения выделяются по процедуре IETF Review и в каждой записи указывается код, назначение, RFC с определением.

Опции сообщения RPL Control.

Значение	***Описание***	***Документ***
0x00	Pad1	Данный документ
0x01	PadN	Данный документ
0x02	DAG Metric Container	Данный документ
0x03	Routing Information	Данный документ
0x04	DODAG Configuration	Данный документ
0x05	RPL Target	Данный документ
0x06	Transit Information	Данный документ
0x07	Solicited Information	Данный документ
0x08	Prefix Information	Данный документ
0x09	Target Descriptor	Данный документ

20.5. Реестр OCP

Агентство IANA создало реестр Objective Code Point (OCP).

Новые коды выделяются по процедуре IETF Review и в каждой записи указывается код, описание, RFC с определением. Значения пока не заданы.

20.6. Новый реестр для алгоритма раздела Security

Агентство IANA создало реестр 8-битовых значений поля Algorithm в разделе Security. Новые коды выделяются по процедуре IETF Review и в каждой записи указывается значение, роль (шифрование или MAC), подпись, RFC. Определённое в настоящее время значение приведено ниже.

Алгоритм раздела Security.

Значение	***Шифрование/MAC***	***Подпись***	***Документ***
0	CCM с AES-128	RSA с SHA-256	Данный документ

20.7. Новый реестр для флагов раздела Security

Агентство IANA создало реестр флагов Security Section. Новые флаги выделяются по процедуре IETF Review и в каждой записи указывается номер бита (0 для старшего бита), описание, RFC с определением. Флаги пока не заданы.

20.8. Новый реестр для уровней защиты

Агентство IANA создало реестр значения Security Level (LVL) для выделенных значений KIM. Новые уровни выделяются по процедуре IETF Review и в каждой записи указывается уровень, значение KIM, описание, RFC с определением. Выделенные в настоящее время уровни показаны в таблице.

Уровни защиты для разных KIM.

Уровень	Значение KIM	Описание	Документ
0	0	Рисунок 11	Данный документ
1	0	Рисунок 11	Данный документ
2	0	Рисунок 11	Данный документ
3	0	Рисунок 11	Данный документ
0	1	Рисунок 11	Данный документ
1	1	Рисунок 11	Данный документ
2	1	Рисунок 11	Данный документ
3	1	Рисунок 11	Данный документ
0	2	Рисунок 11	Данный документ
1	2	Рисунок 11	Данный документ
2	2	Рисунок 11	Данный документ
3	2	Рисунок 11	Данный документ
0	3	Рисунок 11	Данный документ
1	3	Рисунок 11	Данный документ
2	3	Рисунок 11	Данный документ
3	3	Рисунок 11	Данный документ

20.9. Новый реестр для флагов DIS

Агентство IANA создало реестр флагов DIS. Новые флаги выделяются по процедуре IETF Review и в каждой записи указывается номер бита (отсчёт с 0 для старшего бита), описание, RFC с определением. Флаги пока не заданы.

20.10. Новый реестр для флагов DIO

Агентство IANA создало реестр флагов DIO. Новые флаги выделяются по процедуре IETF Review и в каждой записи указывается номер бита (отсчёт с 0 для старшего бита), описание, RFC с определением. Флаги DIO[1] пока не заданы.

20.11. Новый реестр для флагов DAO

Агентство IANA создало реестр флагов DAO. Новые флаги выделяются по процедуре IETF Review и в каждой записи указывается номер бита (0 для старшего бита), описание, RFC с определением. Выделенные флаги указаны в таблице.

Флаги DAO.

Номер бита	Описание	Документ
0	Запрос DAO-ACK (K)	Данный документ
1	Присутствует поле DODAGID (D)	Данный документ

20.12. Новый реестр для флагов DAO Acknowledgement

Агентство IANA создало реестр флагов DAO. Новые флаги выделяются по процедуре IETF Review и в каждой записи указывается номер бита (0 для старшего бита), описание, RFC с определением. Выделенный флаг указан в таблице.

Флаги DAO-ACK.

Номер бита	Описание	Документ
0	DODAGID field is present (D)	Данный документ

20.13. Новый реестр для флагов CC

Агентство IANA создало реестр флагов CC. Новые флаги выделяются по процедуре IETF Review и в каждой записи указывается номер бита (0 для старшего бита), описание, RFC с определением. Выделенный флаг указан в таблице.

Флаг CC.

Номер бита	Описание	Документ
0	CC Response (R)	Данный документ

20.14. Новый реестр для флагов опции DODAG Configuration

Агентство IANA создало реестр флагов опции DODAG Configuration. Новые флаги выделяются по процедуре IETF Review и в каждой записи указывается номер бита (отсчёт с 0 для старшего бита), описание, RFC с определением. Определённые в настоящее время флаги указаны в таблице.

Флаги опции DODAG Configuration.

Номер бита	Описание	Документ
4	Authentication Enabled (A)	Данный документ
5-7	Path Control Size (PCS)	Данный документ

20.15. Новый реестр для флагов опции RPL Target

Агентство IANA создало реестр флагов опции RPL Target. Новые флаги выделяются по процедуре IETF Review и в каждой записи указывается номер бита (отсчёт с 0 для старшего бита), описание, RFC с определением. В настоящее время биты флагов не заданы.

[1] В оригинале ошибочно указано DIS, см. https://www.rfc-editor.org/errata/eid4458. *Прим. перев.*

20.16. Новый реестр для флагов опции Transit Information

Агентство IANA создало реестр флагов опции Transit Information (TIO). Новые флаги выделяются по процедуре IETF Review и в каждой записи указывается номер бита (отсчёт с 0 для старшего бита), описание, RFC с определением. Определённые в настоящее время флаги указаны в таблице.

Флаг опции Transit Information.

Номер бита	***Описание***	***Документ***
0	External (E)	Данный документ

20.17. Новый реестр для флагов опции Solicited Information

Агентство IANA создало реестр флагов опции Solicited Information (SIO). Новые флаги выделяются по процедуре IETF Review и в каждой записи указывается номер бита (отсчёт с 0 для старшего бита), описание, RFC с определением. Определённые в настоящее время флаги указаны в таблице.

Флаги опции Solicited Information.

Номер бита	***Описание***	***Документ***
0	Сопоставление Version (V)	Данный документ
1	Сопоставление InstanceID (I)	Данный документ
2	Сопоставление DODAGID (D)	Данный документ

20.18. ICMPv6 - ошибки в заголовке Source Routing

В некоторых случаях RPL может возвращать сообщения ICMPv6 об ошибке, если сообщение не может быть доставлено по заданному источником заголовку маршрутизации. Такие сообщения ICMPv6 называются «Ошибка в заголовке SRH» (Error in Source Routing Header).

Агентство IANA поддерживает реестр значений поля Code для типов сообщений ICMPv6. Сообщение ICMPv6 типа 1 описывает коды недоступности адресата (Destination Unreachable). Для сообщения «Error in Source Routing Header» был выделен код 7 в реестре ICMPv6 Code Fields Registry for ICMPv6 Message Type 1.

20.19. Область действия группового адреса Link-Local

Правила назначения групповых адресов IPv6 определены в [RFC3307]. Данная спецификация требует выделить новый постоянный групповой адрес с областью действия на локальном канале (link-local) для узлов RPL, названный all-RPL-nodes (все узлы RPL), со значением ff02::1a.

21. Благодарности

Авторы признательны Emmanuel Baccelli, Dominique Barthel, Yusuf Bashir, Yoav Ben-Yehezkel, Phoebus Chen, Quynh Dang, Mischa Dohler, Mathilde Durvy, Joakim Eriksson, Omprakash Gnawali, Manhar Goindi, Mukul Goyal, Ulrich Herberg, Anders Jagd, JeongGil (John) Ko, Ajay Kumar, Quentin Lampin, Jerry Martocci, Matteo Paris, Alexandru Petrescu, Joseph Reddy, Michael Richardson, Don Sturek, Joydeep Tripathi, Nicolas Tsiftes за рецензии, отклики и комментарии.

Спасибо руководителям ROLL Chairs, David Culler и JP. Vasseur, а также руководителю направления Adrian Farrel за руководство и вклад в работу. Спасибо за ранний вклад в работу Robert Assimiti, Mischa Dohler, Julien Abeille, Ryuji Wakikawa, Teco Boot, Patrick Wetterwald, Bryan Mclaughlin, Carlos J. Bernardos, Thomas Watteyne, Zach Shelby, Caroline Bontoux, Marco Molteni, Billy Moon, Jim Bound, Yanick Pouffary, Henning Rogge, Arsalan Tavakoli, предоставившим полезные соображения по устройству RPL.

Защита RPL, описанная в разделах 10, 19 и других местах документа, основана в основном на вкладе команды Security Design: Tzeta Tsao, Roger Alexander, Dave Ward, Philip Levis, Kris Pister, Rene Struik, Adrian Farrel.

Спасибо также Jari Arkko и Ralph Droms за внимательное рецензирование, особенно по вопросам взаимодействия и интеграфии с другими спецификациями IETF.

22. Участник работы

Stephen Dawson-Haggerty

UC Berkeley

Soda Hall

Berkeley, CA 94720

USA

EMail: stevedh@cs.berkeley.edu

23. Литература

23.1. Нормативные документы

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

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460[1], December 1998.

[RFC3447] Jonsson, J. and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, February 2003.

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, December 2005.

[1]Этот документ отменен RFC 8200. *Прим. перев.*

[RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

[RFC6206] Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko, "The Trickle Algorithm", RFC 6206, March 2011.

[RFC6275] Perkins, C., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, July 2011.

[RFC6551] Vasseur, JP., Ed., Kim, M., Ed., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks", RFC 6551, March 2012.

[RFC6552] Thubert, P., Ed., "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6552, March 2012.

[RFC6553] Hui, J. and JP. Vasseur, "The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams", RFC 6553, March 2012.

[RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, March 2012.

23.2. Дополнительная литература

[6LOWPAN-ND] Shelby, Z., Ed., Chakrabarti, S., and E. Nordmark, "Neighbor Discovery Optimization for Low Power and Lossy Networks (6LoWPAN)", Work in Progress[1], October 2011.

[FIPS180] National Institute of Standards and Technology, "FIPS Pub 180-3, Secure Hash Standard (SHS)", US Department of Commerce , February 2008, <http://www.nist.gov/itl/upload/fips180-3_final.pdf>.

[Perlman83] Perlman, R., "Fault-Tolerant Broadcast of Routing Information", North-Holland Computer Networks, Vol.7: p. 395-405, December 1983.

[RFC1958] Carpenter, B., "Architectural Principles of the Internet", RFC 1958, June 1996.

[RFC1982] Elz, R. and R. Bush, "Serial Number Arithmetic", RFC 1982, August 1996.

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC3307] Haberman, B., "Allocation Guidelines for IPv6 Multicast Addresses", RFC 3307, August 2002.

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

[RFC3535] Schoenwaelder, J., "Overview of the 2002 IAB Network Management Workshop", RFC 3535, May 2003.

[RFC3610] Whiting, D., Housley, R., and N. Ferguson, "Counter with CBC-MAC (CCM)", RFC 3610, September 2003.

[RFC3819] Karn, P., Bormann, C., Fairhurst, G., Grossman, D., Ludwig, R., Mahdavi, J., Montenegro, G., Touch, J., and L. Wood, "Advice for Internet Subnetwork Designers", BCP 89, RFC 3819, July 2004.

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

[RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC 4915, June 2007.

[RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, February 2008.

[RFC5184] Teraoka, F., Gogo, K., Mitsuya, K., Shibui, R., and K. Mitani, "Unified Layer 2 (L2) Abstractions for Layer 3 (L3)-Driven Fast Handover", RFC 5184, May 2008.

[RFC5548] Dohler, M., Watteyne, T., Winter, T., and D. Barthel, "Routing Requirements for Urban Low-Power and Lossy Networks", RFC 5548, May 2009.

[RFC5673] Pister, K., Thubert, P., Dwars, S., and T. Phinney, "Industrial Routing Requirements in Low-Power and Lossy Networks", RFC 5673, October 2009.

[RFC5706] Harrington, D., "Guidelines for Considering Operations and Management of New Protocols and Protocol Extensions", RFC 5706, November 2009.

[RFC5826] Brandt, A., Buron, J., and G. Porcu, "Home Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5826, April 2010.

[RFC5867] Martocci, J., De Mil, P., Riou, N., and W. Vermeylen, "Building Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5867, June 2010.

[RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, June 2010.

[RFC6130] Clausen, T., Dearlove, C., and J. Dean, "Mobile Ad Hoc Network (MANET) Neighborhood Discovery Protocol (NHDP)", RFC 6130, April 2011.

[ROLL-TERMS] Vasseur, J., "Terminology in Low power And Lossy Networks", Work in Progress[2], September 2011.

[1]Работа опубликована в RFC 6775. *Прим. перев.*
[2]Работа опубликована в RFC 7102. *Прим. перев.*

Приложение A. Примеры операций

В этом приложении даны примеры, иллюстрирующие распространение адресной информации и префиксов протоколом RPL. Показаны сведения, распространяемые в опциях PIO и RIO, а также использование сообщений DIO и DAO. Приложение не является нормативным и детали плана адресации RPL и автоматической настройки конфигурации могут зависеть от реализации. RPL просто обеспечивает распространение информации, которая может создаваться и применяться другими механизмами.

Примеры иллюстрируют применение схем автоматической настройки адресов, поддерживаемых сведениями, распространяемыми в RPL. При использовании других схем настройки адресов узлы RPL могут быть настроены не устанавливать флаг A в опциях PIO, хотя PIO все равно могут применяться для распространения префиксов и адресов.

A.1. Пример работы в режиме Storing с префиксами узла

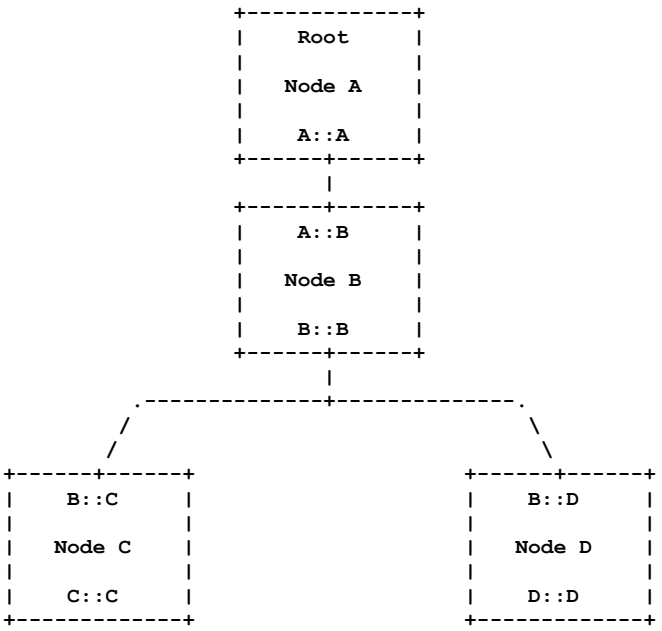


Рисунок 32. Работа в режиме Storing с чужими префиксами.

На рисунке 32 показана логическая архитектура адресации простой сети RPL, работающей в режиме Storing. Узлы A, B, C, D владеют своими префиксами и делают эти префиксы доступными для автоматической настройки адресов устройств на канале (путём установки флагов A и L в опции PIO сообщений DIO). Узел A владеет префиксом A::/64, B - B::/64 и т. д. Узел B автоматически настраивает адреса узлов на канале к узлу A - A::B. Узлы C и D аналогично настраивают адреса из префикса B - B::C и B::D. Узлы могут установить флаг R и опубликовать свои адреса в поле Prefix опции PIO.

A.1.1. Сообщения DIO и PIO

Узел A, например, будет передавать сообщения DIO с опцией PIO вида:

флаг A установлен
флаг L установлен
флаг R сброшен
размер префикса 64
префикс A::

Узел B будет передавать сообщения DIO с опцией PIO вида:

флаг A установлен
флаг L установлен
флаг R установлен
размер префикса 64
префикс B::B

Узел C будет передавать сообщения DIO с опцией PIO вида:

флаг A установлен
флаг L установлен
флаг R сброшен
размер префикса 64
префикс C::

Узел D будет передавать сообщения DIO с опцией PIO вида:

флаг A установлен
флаг L установлен
флаг R установлен
размер префикса 64
префикс D::D

A.1.2. Сообщения DAO

Узел B передаёт узлу A сообщения DAO, содержащие:

Target B::/64

Target C::/64
Target D::/64

Узел C передаёт узлу B сообщения DAO, содержащие:

Target C::/64

Узел D передаёт узлу B сообщения DAO, содержащие:

Target D::/64

A.1.3. База маршрутных данных

Узел A будет собирать в свою базу маршрутной информации (Routing Information Base или RIB) следующие сведения:

A::/64 подключён
B::/64 через link-local узла B
C::/64 через link-local узла B
D::/64 через link-local узла B

Узел B будет собирать в свою базу RIB следующие сведения:

::/0 через link-local узла A
B::/64 подключён
C::/64 через link-local узла C
D::/64 через link-local узла D

Узел C будет собирать в свою базу RIB следующие сведения:

::/0 через link-local узла B
C::/64 подключён

Узел D будет собирать в свою базу RIB следующие сведения:

::/0 через link-local узла B
D::/64 подключён

A.2. Пример работы в режиме Storing в префиксом подсети

На рисунке 33 показана логическая архитектура адресации простой сети RPL, работающей в режиме Storing. В этом примере корневой узел A является источником префикса, используемого для автоматической настройки адресов во всей подсети RPL (это выполняется путём установки флага A и сброса флага L в опции PIO сообщений DIO). Узлы A, B, C, D автоматически получают адреса из префикса A::/64. Узлы могут установить флаг R и опубликовать свои адреса

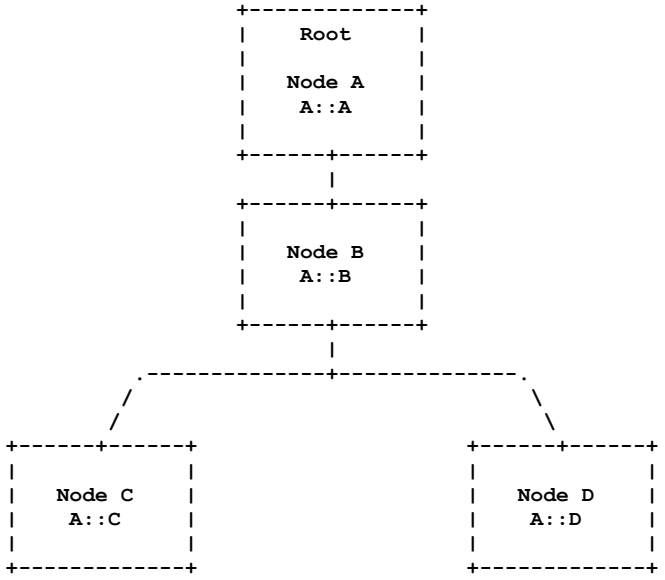


Рисунок 33. Режим Storing с префиксом Subnet-Wide.

в поле Prefix опции PIO.

A.2.1. Сообщения DIO и PIO

Узел A будет передавать сообщения DIO с опцией PIO вида:

флаг A установлен
флаг L сброшен
флаг R сброшен
размер префикса 64
префикс A::

Узел B будет передавать сообщения DIO с опцией PIO вида:

флаг A установлен
флаг L сброшен
флаг R установлен
размер префикса 64
префикс A::B

Узел C будет передавать сообщения DIO с опцией PIO вида:

флаг A установлен
флаг L сброшен

флаг R сброшен
размер префикса 64
префикс A::

Узел D будет передавать сообщения DIO с опцией PIO вида:

флаг A установлен
флаг L сброшен
флаг R установлен
размер префикса 64
префикс A::D

A.2.2. Сообщения DAO

Узел B передаёт узлу A сообщения DAO, содержащие:

Target A::B/128
Target A::C/128
Target A::D/128

Узел C передаёт узлу B сообщения DAO, содержащие:

Target A::C/128

Узел D передаёт узлу B сообщения DAO, содержащие:

Target A::D/128

A.2.3. База маршрутных данных

Узел A будет собирать в свою базу RIB следующие сведения:

A::A/128 подключён
A::B/128 через link-local узла B
A::C/128 через link-local узла B
A::D/128 через link-local узла B

Узел B будет собирать в свою базу RIB следующие сведения:

::/0 через link-local узла A
A::B/128 подключён
A::C/128 через link-local узла C
A::D/128 через link-local узла D

Узел C будет собирать в свою базу RIB следующие сведения:

::/0 через link-local узла B
A::C/128 подключён

Узел D будет собирать в свою базу RIB следующие сведения:

::/0 через link-local узла B
A::D/128 подключён

A.3. Пример работы в режиме Non-Storing с префиксами узла

На рисунке 34 показана логическая архитектура адресации простой сети RPL, работающей в режиме Non-Storing. Узлы A, B, C, D владеют своими префиксами и делают эти префиксы доступными для автоматической настройки адресов устройств на канале (путём установки флагов A и L в опции PIO сообщений DIO). Узел A владеет префиксом A::/64, B - B::/64 и т. д. Узел B автоматически настраивает адреса узлов на канале к узлу A - A::B. Узлы C и D аналогично настраивают адреса из префикса B - B::C и B::D. Узлы могут установить флаг R и опубликовать свои адреса в поле Prefix опции PIO.

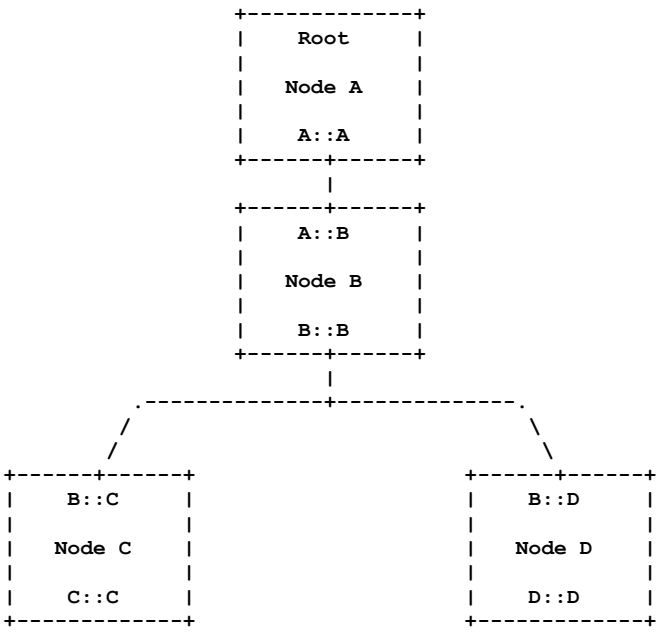


Рисунок 34. Режим Non-Storing с префиксами узлов.

A.3.1. Сообщения DIO и PIO

Опцию PIO в сообщениях DIO режима Non-Storing с принадлежащими узлу префиксами можно считать идентичной опции в режиме Storing (Приложение A.1.1).

A.3.2. Сообщения DAO

Узел B передаёт узлу A сообщения DAO, содержащие:

Target B::/64, Transit A::B

Узел C передаёт узлу A сообщения DAO, содержащие:

Target C::/64, Transit B::C

Узел D передаёт узлу A сообщения DAO, содержащие:

Target D::/64, Transit B::D

A.3.3. База маршрутной информации

Узел A собирает в свою базу RIB ниже сведения, позволяющие создать source route путём рекурсивного поиска в RIB:

A::/64 подключён
B::/64 через A::B
C::/64 через B::C
D::/64 через B::D

Узел B будет собирать в свою базу RIB следующие сведения:

::/0 через link-local узла A
B::/64 подключён

Узел C будет собирать в свою базу RIB следующие сведения:

::/0 через link-local узла B
C::/64 подключён

Узел D будет собирать в свою базу RIB следующие сведения:

::/0 через link-local узла B
D::/64 подключён

A.4. Пример работы в режиме Non-Storing с префиксом масштаба подсети

На рисунке 35 показана логическая архитектура адресации простой сети RPL, работающей в режиме Non-Storing. Корневой узел A является источником префикса, используемого для автоматической настройки адресов во всей подсети RPL (это выполняется путём установки флага A и сброса флага L в опции PIO сообщений DIO). Узлы A, B, C, D автоматически получают адреса из префикса A::/64. Узлы могут установить флаг R и опубликовать свои адреса в поле Prefix опции PIO.

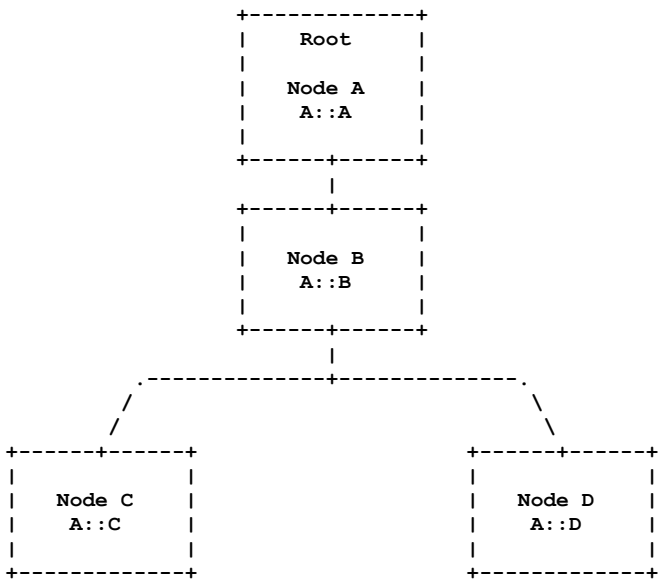


Рисунок 35. Режим Non-Storing с префиксом подсети.

A.4.1. Сообщения DIO и PIO

Узел A будет передавать сообщения DIO с опцией PIO вида:

флаг A установлен
флаг L сброшен
флаг R установлен
размер префикса 64
префикс A::A

Узел B будет передавать сообщения DIO с опцией PIO вида:

флаг A установлен
флаг L сброшен
флаг R установлен
размер префикса 64
префикс A::B

Узел C будет передавать сообщения DIO с опцией PIO вида:

флаг A установлен
флаг L сброшен
флаг R установлен
размер префикса 64
префикс A::C

Узел D будет передавать сообщения DIO с опцией PIO вида:

флаг A установлен
флаг L сброшен
флаг R установлен
размер префикса 64
префикс A::D

A.4.2. Сообщения DAO

Узел B передаёт узлу A сообщения DAO, содержащие:

Target A::B/128, Transit A::A

Узел C передаёт узлу A сообщения DAO, содержащие:

Target A::C/128, Transit A::B

Узел D передаёт узлу A сообщения DAO, содержащие:

Target A::D/128, Transit A::B

A.4.3. База маршрутных данных

Узел A собирает в свою базу RIB ниже сведения, позволяющие создать source route путём рекурсивного поиска в RIB:

A::A/128 подключён
A::B/128 через A::A
A::C/128 через A::B
A::D/128 через A::B

Узел B будет собирать в свою базу RIB следующие сведения:

::/0 через link-local узла A
A::B/128 подключён

Узел B будет собирать в свою базу RIB следующие сведения:

::/0 через link-local узла B
A::C/128 подключён

Узел B будет собирать в свою базу RIB следующие сведения:

::/0 через link-local узла B
A::D/128 подключён

A.5. Пример с внешними префиксами

Рассмотрим простой пример, показанный на рисунке 59, с группой маршрутизаторов, участвующих в сети RPL: корень DODAG, узлы A, O, Z. Корень DODAG и узел Z подключены также к внешним доменам (по отношению к сети RPL), которые могут быть сетями RPL или иного типа.

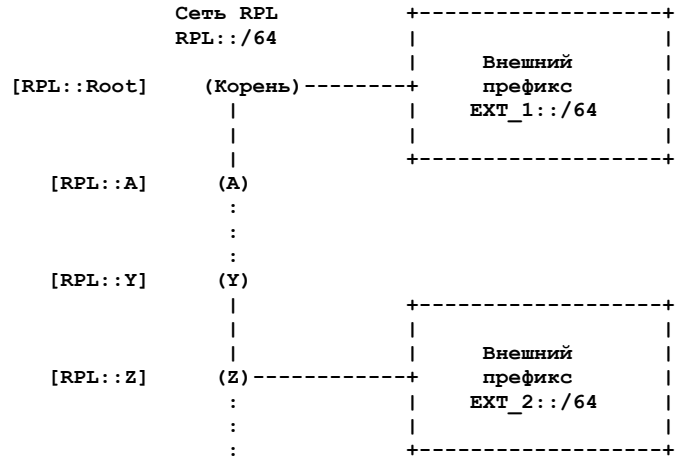


Рисунок 36. Пример простой сети.

В этом примере корень DODAG делает префикс доступным подсети RPL для автоматической настройки адресов. Вся подсеть RPL использует один префикс RPL::/64 для автоматической настройки адресов, хотя в других реализациях могут применяться более сложные или гибридные схемы.

Корень DODAG подключён ко внешнему (относительно RPL) префиксу EXT_1::/64 и может узнать об этом префиксе, например, через явную настройку или IPv6 ND на внешнем (не RPL) интерфейсе. Корень DODAG настроен на анонсирование сведений о подключении к этому префиксу. Аналогично, узел Z имеет подключение к внешнему префиксу EXT_2::/64. Кроме того, под узлом Z размещается субграф DODAG.

1. Корень DODAG добавляет в свои сообщения DIO опцию RIO, содержащую внешний префикс EXT_1::/64. Эта информация может повторятся в сообщениях DIO, выдаваемых другими узлами DODAG. Таким способом сведения о доступности префикса EXT_1::/64 распространяются вниз по DODAG.

2. Узел Z может анонсировать доступность Target-сети EXT_2::/64, передавая сообщения DAO с EXT_2::/64 в качестве Target в одноименной опции и себя (узел Z) как родителя в опции Transit Information (в режиме Storing эта опция Transit Information может не включать адрес узла Z). Корень сети Non-Storing узнает о канале 1-hop (Node Z - EXT_2::/64) и может включать его в задаваемый источником маршрут. Дополнительно узел Z может анонсировать доступность EXT_2::/64 узлам субграфа DODAG, передавая сообщения DIO, где в опции PIO, флаг A сброшен.

Адреса авторов

Tim Winter (редактор)
EMail: wintert@acm.org

Pascal Thubert (редактор)
Cisco Systems
Village d'Entreprises Green Side
400, Avenue de Roumanille
Batiment T3
Biot - Sophia Antipolis 06410
France
Phone: +33 497 23 26 34
EMail: pthubert@cisco.com

Anders Brandt
Sigma Designs
Emdrupvej 26A, 1.
Copenhagen DK-2100
Denmark
EMail: abr@sdesigns.dk

Jonathan W. Hui
Arch Rock Corporation
501 2nd St., Suite 410
San Francisco, CA 94107
USA
EMail: jhui@archrock.com

Richard Kelsey
Ember Corporation
Boston, MA
USA
Phone: +1 617 951 1225
EMail: kelsey@ember.com

Philip Levis
Stanford University
358 Gates Hall, Stanford University
Stanford, CA 94305-9030
USA
EMail: pal@cs.stanford.edu

Kris Pister
Dust Networks
30695 Huntwood Ave.
Hayward, CA 94544
USA
EMail: kpister@dustnetworks.com

Rene Struik
Struik Security Consultancy
EMail: rstruik.ext@gmail.com

JP. Vasseur
Cisco Systems
11, Rue Camille Desmoulins
Issy Les Moulineaux 92782
France
EMail: jpv@cisco.com

Roger K. Alexander
Cooper Power Systems
20201 Century Blvd., Suite 250
Germantown, MD 20874
USA
Phone: +1 240 454 9817
EMail: roger.alexander@cooperindustries.com

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru