

Networking Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 13, 2012

R. Alexander
T. Tsao
Cooper Power Systems
July 12, 2011

Adapted Multimedia Internet KEYing (AMiKEY): An extension of Multimedia
Internet KEYing (MIKEY) Methods for Generic LLN Environments
draft-alexander-roll-mikey-lln-key-mgmt-01

Abstract

Multimedia Internet Keying (MIKEY) is a key management protocol used for real-time applications. As standardized within RFC3830 it defines four key distribution methods, including pre-shared keys, public-key encryption, and Diffie-Hellman key exchange, with allowances for ready protocol extension. A number of additional methods have been developed and continue to be built from the base protocol (see for example, RFC4442, RFC4563, RFC4650, RFC4738, RFC5410, RFC6043 and RFC6267). However, in spite of its extensibility and more general applicability, MIKEY and its related extensions have primarily focused on the support of the Secure Real-time Transport Protocol (SRTP).

This document specifies a simple adaptation of the MIKEY specification to allow the base protocol and its various key management mode extensions to be readily applied in more general environments beyond the multimedia SRTP domain. In particular, the document defines a repurposing of the MIKEY multimedia crypto sessions structure and introduces a set of message extensions to the base specification to allow the MIKEY key management methods to be applied within Low-power and Lossy networks (LLNs) and other general constrained-device networks.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute

working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 13, 2012.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	5
1.1.	Motivation	5
1.2.	MIKEY Key Management Methods Background	6
1.3.	Adapting MIKEY to General LLNs	7
1.4.	Terminology and Definitions	7
1.5.	Document Outline	9
1.6.	Section Headings Notation	10
2.	AMIKEY Overview	10
3.	AMIKEY Extension Elements	14
3.1.	[RFC3830] Pre-shared key	14
3.2.	[RFC3830] Public-Key Encryption	14
3.3.	[RFC3830] Diffie-Hellman Key Exchange	14
4.	[RFC3830] Selected Key Management Functions	14
4.1.	[RFC3830] Key Calculation	14
4.1.1.	[RFC3830] Assumptions	14
4.1.2.	[RFC3830] Default PRF Description	14
4.1.3.	[RFC3830] Generating Keys from TGK	14
4.1.4.	[RFC3830] Generating Keys for MIKEY Messages from an Envelope/Pre-shared Key	14
4.2.	[RFC3830] Pre-defined Transforms and Timestamp Formats	15
4.2.1.	[RFC3830] Hash Functions	15
4.2.2.	[RFC3830] Pseudo-Random Number Generator	15
4.2.3.	[RFC3830] Key Data Transport Encryption	15
4.2.4.	[RFC3830] MAC Verification Message Function	15
4.3.	[RFC3830] Certificates, Policies and Authorization	15
4.4.	[RFC3830] Retrieving the Data SA	15
5.	[RFC3830] Behavior and Message Handling	15
6.	[RFC3830] Payload Encoding	15
6.1.	[RFC3830] Common Header Payload (HDR)	16
6.1.1.	[RFC3830] SRTP ID	18
6.1.2.	The Generic_LLN-ID Map Type	18
6.2.	[RFC3830] Key Data Transport Payload (KEMAC)	20
6.3.	[RFC3830] Envelope Data Payload (PKE)	20
6.4.	[RFC3830] DH Data Payload (DH)	20
6.5.	[RFC3830] Signature Payload (SIGN)	21
6.6.	[RFC3830] Timestamp Payload (T)	21
6.7.	[RFC3830] ID Payload (ID)	21
6.8.	[RFC3830] Cert Hash Payload (CHASH)	21
6.9.	[RFC3830] Ver msg payload (V)	21
6.10.	[RFC3830] Security Policy (SP) Payload	21
6.10.1.	[RFC3830] SRTP Policy	23
6.10.2.	AMIKEY Generic_LLN Policy	23
6.11.	[RFC3830] RAND Payload (RAND)	24
6.12.	[RFC3830] Error Payload (ERR)	24
6.13.	[RFC3830] Key Data Sub-Payload	24
6.14.	[RFC3830] Key Validity Data	25

6.15. [RFC3830] General Extension Payload	25
6.16. Key Index Payload	25
6.17. Key Source Identifier Payload	25
6.18. Key Activation Time Payload	26
7. [RFC3830] Transport Protocols	27
8. Security Considerations	27
9. [RFC3830] Groups	27
10. Additional Specification Considerations	27
11. IANA Considerations	27
12. Acknowledgments	29
13. References	29
13.1. Normative References	29
13.2. Informative References	29
Authors' Addresses	31

1. Introduction

Any sufficiently large scale network offering security services requires an automated key management mechanism for the exchange of keys and the update of related security credentials [RFC4107]. Key management may be needed for individual session exchanges or for the long-term control and update of security parameters from which session keys may be derived. In many Low-power and Lossy networks (LLN) and other constrained-device environments, key management emphasis is often on the management of long-term keys. This may automatically follow network associations based on device pre-configuration or may be based on specified key lifetimes or administrative or event-driven need for key credential changes. This would apply to the case of a network routing protocol like RPL ([I-D.ietf-roll-rpl]) that employs security as well as to other secured communications layer protocols.

Multimedia Internet Keying (MIKEY) is a key management protocol that has been used for real-time applications both for peer-to-peer and group communications. The capabilities of the protocol lend themselves just as readily to the management of long-term keys as to per-session or per association key control. MIKEY [RFC3830] defines four key distribution methods including pre-shared keys, public-key encryption, and Diffie-Hellman key exchange. Given its design simplicity, efficiency and flexibility a number of additional modes and extensions have indeed been developed and continue to be built from the base protocol (see for example, [RFC4442], [RFC4563], [RFC4650], [RFC4738], [RFC5410], [RFC6043] and [RFC6267]). MIKEY and its related RFC extensions have however primarily focused on the support of the SRTP and related Session Initiation Protocol (SIP) call scenarios [RFC3711].

This document specifies an adaptation of the MIKEY protocol specification to allow the base protocol and its various key management mode extensions to be more generally applied to LLN environments. In particular, the document defines a repurposing of the MIKEY multimedia crypto sessions structure to allow optional support for simultaneous management of multiple protocol or device interface key. The specification also introduces a set of message extensions to the base MIKEY protocol to allow its key management methods to be applied within generic LLN and constrained-device networks.

1.1. Motivation

Key distribution describes the process of delivering cryptographic keys to the required communicating parties. The MIKEY protocol has defined the mechanisms for establishing the security context used by

SRTP however the mechanisms for security parameter negotiation and update is just as readily extended to LLN protocols.

The flexibility to employ different key distribution methods according to available network infrastructure and particular operating scenarios together with the compact efficiency of its binary specification makes MIKEY well suited for general LLN use. The wide range of key management support extending from light-weight, low latency half round-trip pre-shared key distribution methods to multi-exchange Diffie-Hellman key agreements protected with digital signatures or pre-shared keys offers great flexibility to meet the needs of diverse LLN application environments.

The option to embed the MIKEY key management messages within an existing network signaling protocol or to be directly transported over UDP or TCP (using port 2269) also increases the ability to apply the methods in more general LLN domains.

MIKEY has met its original stated design goals [RFC3830] of end-to-end security, simplicity, efficiency, tunneling (even beyond integration with Session Description Protocol (SDP) [RFC4566] or RTCP [RFC3605]), and independence of underlying transport. In so doing it offers an excellent base for a generic key management protocol for Low-power Lossy Network (LLN) application. Key management protocols are also difficult to design and validate (see [RFC4107] guidelines) providing a further motivation for reliance on an established protocol like MIKEY that has had the benefit of wider operational deployment and evaluation.

1.2. MIKEY Key Management Methods Background

As noted in [RFC5197], several key distribution methods have been described for MIKEY, including:

- o Symmetric key distribution as defined in [RFC3830] (MIKEY-PSK)
- o Asymmetric key distribution as defined in [RFC3830] (MIKEY-RSA)
- o Diffie-Hellman key agreement protected by digital signatures as defined in [RFC3830] (MIKEY-DHSIGN)
- o Diffie-Hellman key agreement protected by symmetric pre-shared keys as defined in [RFC4650] (MIKEY-DHMAC)
- o Asymmetric key distribution (based on asymmetric encryption) with in-band certificate provision as defined in [RFC4738] (MIKEY-RSA-R)

Further extensions to MIKEY comprising algorithm enhancements and new payload definitions have since been defined generally motivated by the specific problems associated with SIP signaling and associated multimedia use case scenarios (see [RFC5197] for an earlier assessment). This specification proposes a new extension that is focused on a new domain of application.

1.3. Adapting MIKEY to General LLNs

This document specifies a set of additional message information elements to the base MIKEY protocol that provide both algorithm and message payload extensions. These additions allow the adapted protocol to be used directly for key transport and security policy specification between communications generic network entities. Furthermore, through integration within the base MIKEY specification it will allow current and future key methods and extensions to be utilized outside of the current multimedia environment.

The developed protocol adaption includes the specification of alternative default algorithms (in particular AES-based) and configurations that are particular to more constrained communications devices and using MIKEY's general extensibility to define new elements applicable to the LLN environment.

An important element of the protocol extension is the re-use of the MIKEY crypto-session structure to apply to individual device communications protocol layers or interfaces instead of applying to multimedia streams. By maintaining this base protocol structure and re-purposing associated message identifiers, the specification minimizes the protocol changes needed for network adaptation.

As with the original specification the intent is to allow MIKEY messages to be embedded into existing communications signaling protocols or to be independently transported between communicating entities over UDP or TCP transport connections.

Note: While MIKEY and its extensions provide a variety of choices in terms of modes of operation, implementations for a given LLN application domain will be able to simplify node behavior by operating in a single mode. To ensure necessary interoperability within the LLN environment, mandatory methods within the Adapted MIKEY protocol (AMIKEY), akin to those of MIKEY, shall be specified.

1.4. Terminology and Definitions

The following definitions have been taken from [RFC3830] with necessary augmentation for AMIKEY as indicated:

(Data) Security Protocol

The security protocol used to protect the actual data traffic. Examples of security protocols are IPsec and SRTP. For generic LLNs, security protocols may include secure versions of protocols such as RPL [I-D.ietf-roll-rpl].

Data SA

Data Security Association information for the security protocol, including a TEK and a set of parameters/policies.

CS Crypto Session, uni- or bidirectional data stream(s) protected by a single instance of a security protocol. For AMIKEY the concept of a crypto-session is expanded to allow definition of a particular protocol layer, logical device interface, or other communications association for which key management support is provided.

CSB Crypto Session Bundle, collection of one or more Crypto Sessions, which can have common TGKs (see below) and security parameters.

CS ID Crypto Session ID, unique identifier for the CS within a CSB. For AMIKEY the CS ID is used to identify a specific protocol layer, logical device interface or other communications association for which AMIKEY is being used to support key management (establishment of re-keying update).

CSB ID

Crypto Session Bundle ID, unique identifier for the CSB. For AMIKEY the CSB ID in conjunction with the Timestamp field is used as a unique key management exchange message reference identifier. This identifier will allow for the acknowledged key management message exchanges where applicable. The ID plus timestamp will also support the filtering of repeated or redundant AMIKEY messages when key management occurs over an unreliable transport network.

TGK TEK Generation Key, a bit-string agreed upon by two or more parties, associated with CSB. From the TGK, Traffic-Encrypting Keys can then be generated without needing further communication.

TEK Traffic-Encrypting Key, the key used by the security protocol to protect the CS (this key may be used directly by the security protocol or may be used to derive further keys depending on the security protocol). The TEKs are derived from the CSB's TGK.

The following definitions have been added to the ones from [RFC3830] specifically related to supporting AMIKEY:

Key Index

The Key Index (KI) is used as identifier to allow for reference to the key(s) that are associated with a given CS. Where TEKs may be updated over time a TKG can be associated with a KI that is transported as a payload within the AMIKEY message from the Initiator. Any TEK generated from the AMIKEY TKG shall be assigned the key index value associated with the TKG. Within general LLN protocol communications related to a given CS (device layer protocol or interface), to ensure security association synchronization reference can be made to the key index that is being applied for the given protocol security. Following successfully TKG key establishment communicating devices can verify security contexts through reference to maintained KI (see Section 6.16).

Key Source Identifier

The Key Source Identifier (KSI) is used as a logical identifier to allow for reference to the entity associated with the origination of a given TKG. Where TEKs are dynamically generated or updated, each TKG can be associated with a specific key source. The KSI, when used, is transported as a payload within the AMIKEY message from the entity responsible for the TKG origination (see Section 6.17).

1.5. Document Outline

Section 2 provides a brief general system overview of key management as introduced in MIKEY specification. This section generalizes the context in which the Adapted MIKEY (AMIKEY) protocol extension is applied. It also provides a reference to the common key management operating base of MIKEY and AMIKEY.

Sections 3 to 4 go into further detail by identifying the specific section and subsection extensions and enhancements needed to support the MIKEY protocol adaptation. These Sections mirror those of MIKEY [RFC3830] and are used to show the necessary commonality and make reference to specific changes would be required for AMIKEY. Reference is made only to the applicable Sections and Subsections of [RFC3830] for which special changes are proposed.

Section 6 includes the specific protocol specification elements that are needed to extend MIKEY for the support of the generic LLN key management requirements.

The remaining document sections are place-holders for standard RFC

draft sections.

1.6. Section Headings Notation

This document is written as a delta document to [RFC3830]. For ease of cross-reference and to maintain consistency with the MIKEY specification document structure, Section heading and Table and Figure numbers are maintained consistent with the [RFC3830] usage.

The notation of Section number followed by [RFC3830] "x.x.[RFC3830]" is used in this document for Sections specifically meant to align with [RFC3830]. Section numbers followed by [RFC3830] with additional heading text indicates some new element or clarification introduced by this specification. Section numbers followed by [RFC3830] without further heading text implies no change to [RFC3830] and is used only to align and maintain the current document headings structure.

The new parameters introduced in this specification are made consistent with the MIKEY recommendations (see Section 4.2.9 [RFC3830]).

2. AMIKEY Overview

This section provides an overview of AMIKEY. Material from MIKEY [RFC3830] is also repeated to clearly establish the common context in which MIKEY can be applied to LLN environments with the simple extension to the Adapted MIKEY (AMIKEY) specification.

The objective of the AMIKEY extension is exactly the same as that of MIKEY - "to produce a data security association (SA) for a security protocol, including a Traffic-Encrypting Key (TEK), which is derived from a TEK Generation Key (TGM), and used as input for the security protocol." In the case of AMIKEY the objective is support generic security protocols and particularly those that may be associated with LLNs.

AMIKEY uses the specified MIKEY mechanisms and features to "support the possibility of establishing keys and parameters for more than one security protocol (or for several instances of the same security protocol) at the same time." In MIKEY the Crypto Session Bundle (CSB), which derives from the multimedia (multi-stream) context, is used to denote this collection of one or more Crypto Sessions that can have a common TGM and security parameters, but that obtain distinct TEKs from MIKEY.

In the AMIKEY extension, the concept of CSB is used to provide the

option of simultaneously establishing multiple SAs on a given device. The individual Crypto Session (CS) SAs may be associated with different device layer or device interface security protocols. AMIKEY further uses the flexibility of the MIKEY specification to allow separate security policies to be defined in the SA established for each security protocol. The distribution mechanisms defined by MIKEY for re-keying and updating of established security associations is hence also directly applied. The ability to establish and maintain multiple SAs through a single key management association provides an important efficiency element in LLN domains.

As specified in [RFC3830], Section 2.3, the procedure of setting up a CSB and creating a TEK (and Data SA), is done in accordance with Figure 1:

1. A set of security parameters and TGK(s) are agreed upon for the Crypto Session Bundle. This is done by one of many alternative key transport/exchange mechanisms (see [RFC3830], Section 3, as well as subsequent extension RFCs).
2. The TGK(s) is used to derive (in a cryptographically secure way) a TEK for each Crypto Session or associated security protocol.
3. The TEK, together with the security protocol parameters, represent the Data SA, which is used as the input to the security protocol(s).

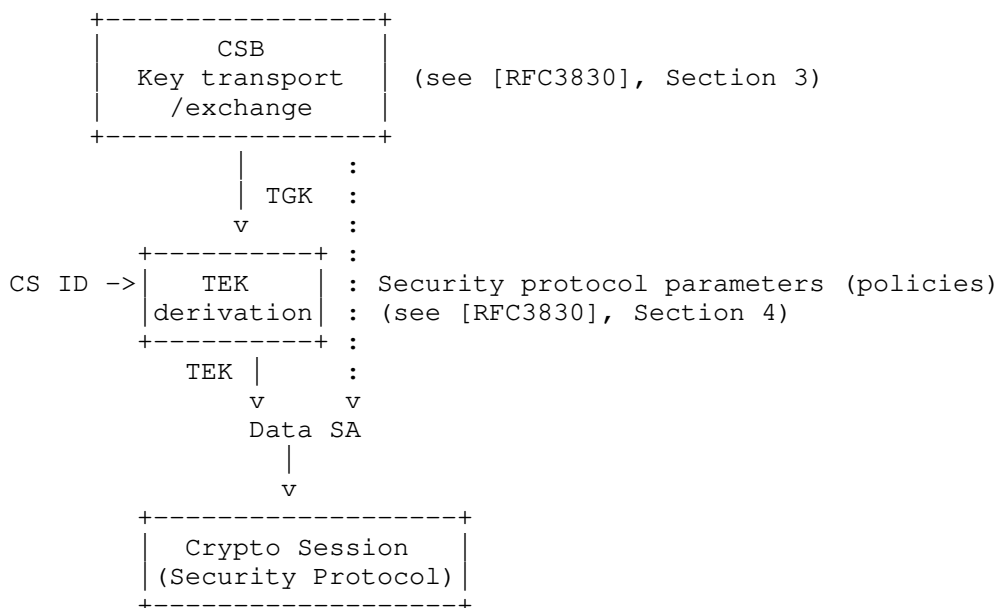


Figure 1: Overview of MIKEY (and AMIKEY extension) key management procedure

For generic LLNs that are the focus of this document, the default algorithms applied in the generation of the TEK for each protocol is defined within this AMIKEY specification. An additional MIKEY message extension is also specified to define the security protocol parameters (policies) for generic LLNs.

Whereas MIKEY CS IDs are associated with multimedia streams and have no intrinsic designation, in this specification the CS IDs are assigned values (public or private/vendor-specific) that are used to identify security protocols associated with specific device protocol layers or device interfaces.

As considered for the device security model discussed in [I-D.ietf-roll-security-framework], Section 6.5, Figure 2 provides an overview of the key management context introduced by the AMIKEY extension defined in this specification. The multi-protocol key management capability (through the particular use of the MIKEY CS-IDs) allows for the efficient, simultaneous management and update of one or more protocol layer security parameters.

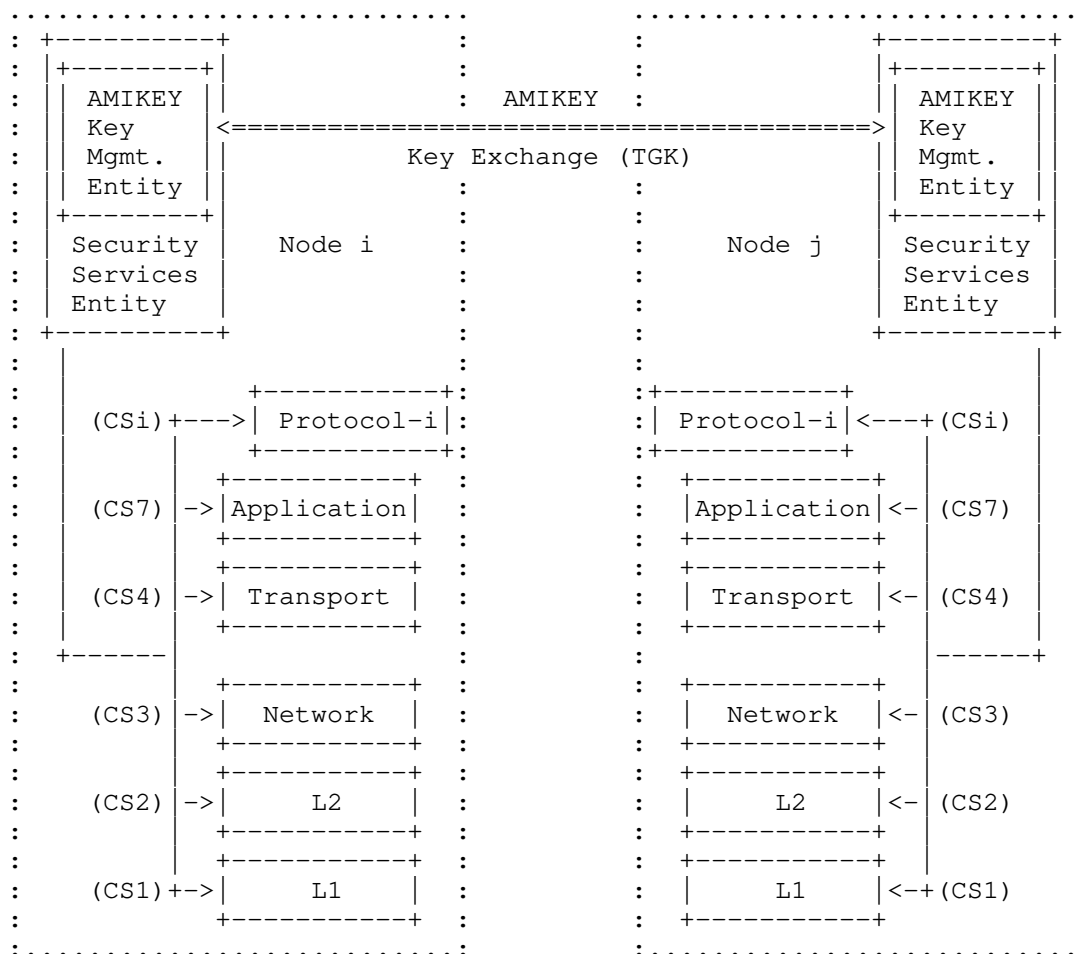


Figure 2: Overview of AMIKEY Multi-protocol Key Management Context

As in the base MIKEY specification, the security protocol can either use the TEK directly, or, if supported, derive further session keys from the TEK. It is however up to the targeted security protocol and the associated security policy to define how the TEK is used.

MIKEY can be used to update TEKs and the Crypto Sessions in a current Crypto Session Bundle (see [RFC3830], Section 4.5). This is done by executing the transport/exchange phase once again to obtain a new TGK (and consequently derive new TEKs) or to update some other specific CS parameters.

3. AMIKEY Extension Elements

The following Section and Subsections detail the proposed additions to the MIKEY specification [RFC3830] to support the AMIKEY extension. The Section heading outline of the MIKEY specification is used to indicate the delta changes made by the AMIKEY extension.

3.1. [RFC3830] Pre-shared key

3.2. [RFC3830] Public-Key Encryption

3.3. [RFC3830] Diffie-Hellman Key Exchange

4. [RFC3830] Selected Key Management Functions

For AMIKEY all the key derivation functionality defined in MIKEY shall be based on a new default Pseudo-Random Function (PRF) given by the AES-based, AES-XCBC-PRF-128 algorithm as specified in [RFC4434].

4.1. [RFC3830] Key Calculation

4.1.1. [RFC3830] Assumptions

For AMIKEY `cs_id` is defined so that session represents a protocol layer, logical device interface, or communications association. The `cs-id` values shall be as defined in this specification (see Section 6.1.2) and may be public or private/vendor-specific.

4.1.2. [RFC3830] Default PRF Description

For AMIKEY the default pseudo random function shall be AES-XCBC-PRF-128 [RFC4434]. Note: AES-XCBC-PRF-128 aligns with HMAC-SHA1 and HMAC-MD5 as PRFs.

4.1.3. [RFC3830] Generating Keys from TGK

For AMIKEY the `cs-id` values shall be as defined in this specification (see Section 6.1.2).

4.1.4. [RFC3830] Generating Keys for MIKEY Messages from an Envelope/ Pre-shared Key

Change from default PRF to the default AMIKEY PRF given in Section 4.1.2 of this specification.

Note: For AMIKEY, the Authentication key constant SHALL be used for generating the single TEK in the case of authenticated encryption

algorithms (such as AES-CCM).

4.2. [RFC3830] Pre-defined Transforms and Timestamp Formats

4.2.1. [RFC3830] Hash Functions

For AMIKEY the default hash function shall be AES-XCBC-PRF-128 [RFC4434].

4.2.2. [RFC3830] Pseudo-Random Number Generator

For AMIKEY it shall be MANDATORY to implement the new default AES-XCBC-PRF-128 PRF specified in [RFC4434] (See Section 4.1.2 of this specification).

4.2.3. [RFC3830] Key Data Transport Encryption

As in MIKEY the default and mandatory-to-implement key transport encryption shall be AES in Counter mode using a 128-bit key (derived as defined in Section 4.1.4 above). The applied Counter shall be the IV defined in [RFC3830], Section 4.2.3.

4.2.4. [RFC3830] MAC Verification Message Function

For AMIKEY AES-CCM-64 shall be the defined default for key message authentication. The Counter used shall be the IV defined in [RFC3830], Section 4.2.3.

4.3. [RFC3830] Certificates, Policies and Authorization

4.4. [RFC3830] Retrieving the Data SA

For AMIKEY the retrieval of a Data SA will depend on the security protocol. The support for different security protocols shall be explicitly identified through the use of public CS ID values (see Section 6.1.2 of this specification).

5. [RFC3830] Behavior and Message Handling

6. [RFC3830] Payload Encoding

The generic LLN security protocol parameters may be transported between peers as part of a key establishment or re-keying exchange. Based on IANA registration, MIKEY currently only defines two payloads for transporting the security policy information (see Section 6.10 of [RFC3830] and [RFC4442]). This section describes the extension of

MIKEY to allow the transport of Generic LLN security policy information and associated key(s) as well as applicable PRF used for key derivation.

This section describes, in detail, the payload for support of the Generic LLN security protocol(s) specified by the Adapted MIKEY protocol. As in RFC3830, for all encoding, network byte order is always used, and the sign ~ indicates a variable length field.

6.1. [RFC3830] Common Header Payload (HDR)

The Common Header payload MUST always be present as the first payload in each message. The Common Header includes a general description of the exchange message.

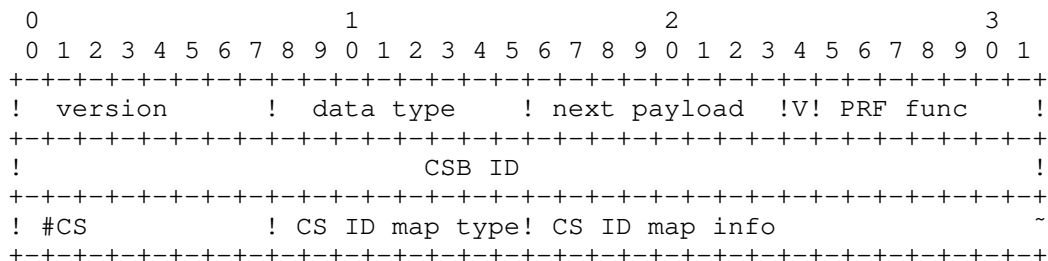


Figure 3: Common Header [RFC3830]

- o version (8 bits): the version number of MIKEY.
- o version = 0x01 refers to MIKEY as defined and maintained in [RFC3830].
- o version = 0x03 (to be assigned by IANA) shall be used to refer to AMIKEY as defined and maintained in this document.
- o data type (8 bits): describes the type of message (e.g., public-key transport message, verification message, error message). See latest IANA registered values. No additional values are specified for AMIKEY (TBD).
- o next payload (8 bits): identifies the payload that is added after this payload. See latest IANA registered values.

For AMIKEY a new next payload value is assigned to carry the Key Index parameter (see also Section 6.16).

Next Payload	Value	Section
Last payload	0	-
...		
Key Index	n	Section 6.16 as given by the AMIKEY specification (value to be assigned by IANA).
Key act time	m	Section 6.18 as given by the AMIKEY specification (value to be assigned by IANA)

Table 6.1.b

- o V (1 bit): flag to indicate whether a verification message is expected or not (this only has meaning when it is set by the Initiator).
- o PRF func (7 bits): indicates the PRF function that has been/will be used for key derivation; for AMIKEY a new value, 2, has been specified to indicate the PRF that must be supported for LLNs.

PRF Function	Value	Comments
AES-XCBC-PRF-128	2	As specified in [RFC4434] and that shall be mandatory for AMIKEY

Table 6.1.c

(AMIKEY value to be assigned by IANA)

- o CSB ID (32 bits): identifies the CSB (generated as specified in [RFC3830]); for AMIKEY this field is used as a message reference identifier to allow for duplicate detection where message exchanges occur over an unreliable transport network.
- o #CS (8 bits): indicates the number of Crypto Sessions that will be handled within the CBS; for AMIKEY this field indicates the number of protocol layers, logical device interfaces, or other communications associations that are being configured or managed within the current key management message exchange.
- o CS ID map type (8 bits): specifies the method of uniquely mapping. Crypto Sessions to the security protocol sessions; for AMIKEY a new value, 3, has been specified to indicate the Generic-LLN map

type that must be supported for LLNs.

CS ID Map Type	Value	Comments
Generic_LLN-ID	3	As specified in this document and as mandatory for AMIKEY

Table 6.1.d

(AMIKEY value to be assigned by IANA)

- o CS ID map info (variable length): identifies the crypto session(s) for which the SA should be created. For AMIKEY the GENERIC_LLN map type (defined in Section 6.1.2 below) is used to specify the security association for the individual protocol layers, logical device interfaces, or other communications associations for which key management is being provided.

6.1.1. [RFC3830] SRTP ID

6.1.2. The Generic_LLN-ID Map Type

For the Generic_LLN map type, the CS ID map info consists of #CS (see Section 6.1) number of blocks or segments, where each segment maps policies (and a key) to a specific protocol layer, logical device interface or other communications association security protocol.

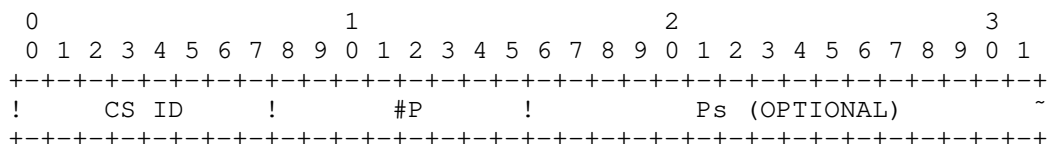


Figure 4: Generic_LLN-ID Map Type

- o CS ID (8 bits): specifies the CS ID used to identify a given security protocol; for AMIKEY, when used in conjunction with the Generic-LLN map type, values 0-127 shall be reserved for assignment (by IANA) to specific protocol layer, logical device interface, or other communications association security protocols while values 128-255 shall be Reserved for Private Use.

Note: A combination of public and private CS IDs can be specified

within a given CSB when combined key management is being applied.

The following values are currently specified in this document (for example, with values to be assigned by IANA):

CS ID	Value	Comments
Reserved	0	
Generic PHY Layer	1	
Generic Link Layer	2	
Generic Network Layer	3	
Generic Transport Layer	4	
Generic Application Layer	7	
RPL Protocol	20	
...		
Reserved values	128-255	Reserved for private use

Table 6.1.e

- o #P (8 bits): indicates the number of security policies provided for the crypto session (given by the CS ID) for which key management is being provided. In response messages, #P SHALL always be exactly 1. So if #P = 0 in an initial message, a security profile MUST be provided in the response message. If #P > 0, one of the suggested policies SHOULD be chosen in the response message. If needed, the suggested policies MAY be changed.
- o Ps (variable length): lists the policies for the crypto session for which key management is being provided. It SHALL contain exactly #P policies, each having the specified Prot type (see Section 6.10).

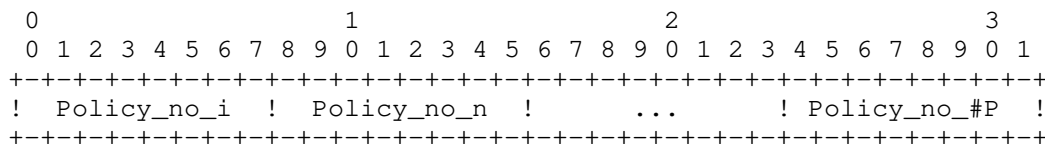


Figure 5: Policies

- o Policy_no_i (8 bits): a policy_no that corresponds to the policy_no of a SP payload. In response messages, the policy_no may refer to a SP payload in the initial message. The policy numbers should be listed in increasing order.

6.2. [RFC3830] Key Data Transport Payload (KEMAC)

This section shall apply entirely as specified for MIKEY in [RFC3830] with the addition of the specific message authentication code algorithms given below for AMIKEY.

- o MAC alg (8 bits): specifies the authentication algorithm used.

MAC alg	Value	Comments	Length (bits)
NULL	0	restricted usage [RFC3830], Section 4.2.4	0
HMAC-SHA-1-160	1	Mandatory, [RFC3830], Section 4.2.4	160
HMAC-SHA-256-256	2	Mandatory, [RFC3830], Section 4.2.4	256
AES-CBC-MAC-32	3	Mandatory for AMIKEY, see Section 4.2.4	32
AES-CBC-MAC-64	4	Mandatory for AMIKEY, see Section 4.2.4	64
AES-CBC-MAC-128	5	Mandatory for AMIKEY, see Section 4.2.4	128

Table 6.2.b

(Values for AMIKEY to be assigned by IANA)

- o MAC (variable length): the message authentication code of the entire message.

For AMIKEY the use of AES-CBC-MAC-n may be applied in conjunction with the AES-CM encryption as given by the Encr alg field. This authenticated encryption shall be applied using an AES-CCM-n implementation.

6.3. [RFC3830] Envelope Data Payload (PKE)

6.4. [RFC3830] DH Data Payload (DH)

6.5. [RFC3830] Signature Payload (SIGN)

6.6. [RFC3830] Timestamp Payload (T)

6.7. [RFC3830] ID Payload (ID)

For AMIKEY the range of ID types shall be extended to allow for an expanded array of communications protocol entities that may be key management participants. The IDs are carried within the key management message ID payload field with the TLV format as specified in [RFC3830], Section 6.7.

ID Type	Value	Comments
IPv6 Address	4	As specified for AMIKEY
Device MAC Address	5	As specified for AMIKEY
Other (TBD)	n	As specified for AMIKEY

Table 6.7.a

The IPv6 Address ID type is used to allow an IPv6 Address to be referenced as the unique entity identifier of the key management correspondents. To directly reference the IPv6 Address of the exchanged packets, the ID len value will be set to zero and no ID data included in the value field.

The Device MAC Address is used to allow a MAC address to be referenced as the unique entity identifier for correspondents in a key management exchange.

6.8. [RFC3830] Cert Hash Payload (CHASH)

6.9. [RFC3830] Ver msg payload (V)

6.10. [RFC3830] Security Policy (SP) Payload

The Security Policy payload defines a set of policies that apply to a specific security protocol.

For AMIKEY the definition is based on the same security policy payload definition in [RFC3830], Section 6.10, with a new security protocol (Generic-LLN) as defined below.

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+
! Next payload ! Policy no   ! Prot type   ! Policy param ~
+-----+-----+-----+-----+
~ length (cont) ! Policy param                                     ~
+-----+-----+-----+-----+

```

- o Next payload (8 bits): Identifies the payload that is added after this payload. See Section 6.1 of [RFC3830] for more details.
- o Policy no (8 bits): Each security policy payload must be given a distinct number for the current MIKEY session by the local peer. This number is used to map a cryptographic session to a specific policy (see also Section 6.1.1 of [RFC3830]).
- o Prot type (8 bits): This value defines the security protocol; For AMIKEY an additional value shall be assigned as given below.

Prot Type	Value	Comments
Generic_LLN	3	As specified for AMIKEY

Table 6.10

- o Policy param length (16 bits): This field defines the total length of the policy parameters for the selected security protocol.
- o Policy param (variable length): This field defines the policy for the specific security protocol. The Policy param part is built up by a set of Type/Length/Value (TLV) payloads. For each security protocol, a set of possible type/value pairs can be negotiated as defined.

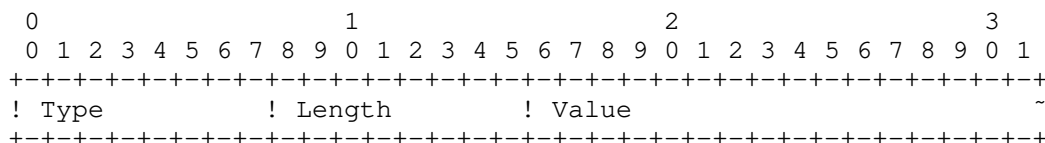


Figure 6: Policy Parameter

- o Type (8 bits): Specifies the type of the parameter.
- o Length (8 bits): Specifies the length of the Value field (in bytes).
- o Value (variable length): Specifies the value of the parameter.

6.10.1. [RFC3830] SRTP Policy

6.10.2. AMIKEY Generic_LLN Policy

This policy specifies the parameters for the Generic_LLN (G_LLN) protocol for which key management is being provided. The types/values that can be negotiated are defined by the following table for the known, assigned CS ID values. For Vendor-specific, private CS ID values the applicable policy specification for a given crypto session will be left to the communicating parties.

Type	Meaning	Possible Values
0	Encryption algorithm	See below
1	Encryption key length	Depends on cipher used
2	Authentication algorithm	See below
3	Authentication key length	Depends on MAC used
4	Generic LLN PRF	See below
5	Encryption off/on	0 if off, 1 if on

Table 6.10.2.a

For the Encryption algorithm, a one byte length is sufficient. For AMIKEY the currently defined possible Values are:

G_LLN encr alg	Value
NULL	0
AES-CM-128	1

Table 6.10.2.b

For the Authentication algorithm, a one byte length is sufficient. For AMIKEY the currently defined possible Values are:

G_LLN auth alg	Value	Comments
NULL	0	Not recommended for operational use
AES-CBC-MAC-32	1	
AES-CBC-MAC-64	2	
AES-CBC-MAC-128	3	
RSA-SHA-256 Sig	4	

Table 6.10.2.c

Note: Since authentication is mandatory for operational protocol security, where Encryption is set "on" by the Generic_LLN policy, authenticated encryption, AES-CCM-n, with the MAC size given by the selected authentication algorithm, or AES-CM with authentication given by the identified Signature algorithm, shall be applied.

For the Generic_LLN pseudo-random function, a one byte length is also sufficient. For AMIKEY the currently defined possible Values are:

Generic_LLN PRF	Value
AES-XCBC-PRF-128	0

Table 6.10.2.d

6.11. [RFC3830] RAND Payload (RAND)

6.12. [RFC3830] Error Payload (ERR)

6.13. [RFC3830] Key Data Sub-Payload

6.14. [RFC3830] Key Validity Data

6.15. [RFC3830] General Extension Payload

6.16. Key Index Payload

For AMIKEY the Key Index (KI) payload is used to specify the value of the key index associated with a given TKG.

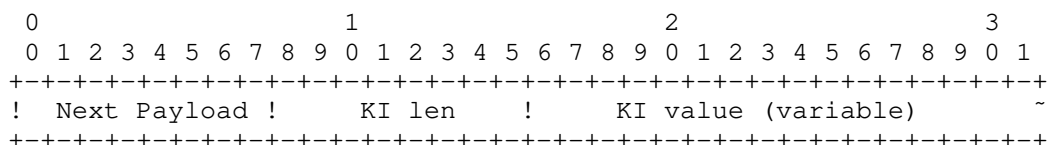


Figure 7: Key Index

- o Next payload (8 bits): identifies the payload that is added after this payload. See Section 6.1 [RFC3830] for values.
- o KI len (8 bits): indicates the length of the key source identifier field.
- o KI value (variable length): indicates the value of the key index to be assigned to any CS TEK generated from the transported TKG.

6.17. Key Source Identifier Payload

For AMIKEY the Key Source Identifier payload is used to provide a logical reference to the entity associated with the origination of a given TGK. The specification of the Key Source Identifier (KSI) shall be given by the supported security protocol (for example, the secured RPL routing protocol [I-D.ietf-roll-rpl] specifies the use of an 8-byte KSI).

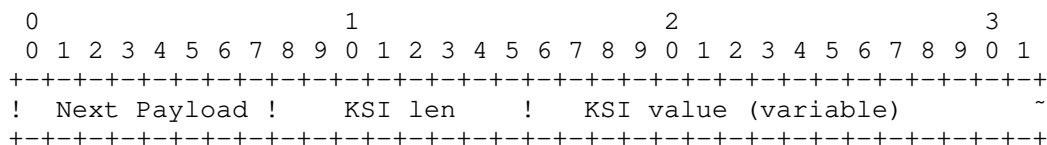


Figure 8: Key Source Identifier

- o Next payload (8 bits): identifies the payload that is added after this payload. See Section 6.1 [RFC3830] for values.
- o KSI len (8 bits): indicates the length of the key source identifier field.
- o KSI value (variable length): specifies the logical identifier assigned to the Source or Originator of a given TGK.

6.18. Key Activation Time Payload

For AMIKEY the Key Activation time payload is used to specify the time at which a new key derived from a communicated TGK shall become active for the associated device protocol or interface. The Key Activation time is used only when needed to specify a delay or future activation of an updated key. The format of this AMIKEY information element type shall be the same as that of the Timestamp payload (T) [RFC3830].

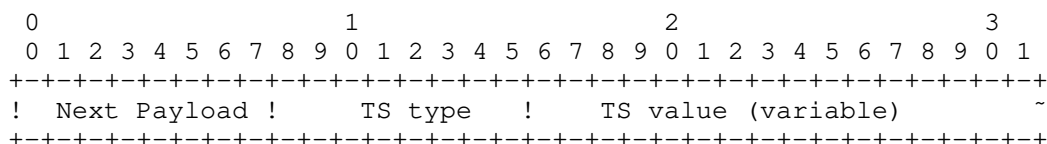


Figure 9: Key Activation Time Payload

- o Next payload (8 bits): identifies the payload that is added after this payload. See Section 6.1 [RFC3830] for values.
- o TS type (8 bits): indicates the timestamp type use to convey the time at which a new derived TEK shall become active (See Section 6.6 [RFC3830]).

- o TS value (variable length): the timestamp value of the specified TS type (See Section 6.6 [RFC3830]).

7. [RFC3830] Transport Protocols

As in [RFC3830], AMIKEY may be integrated within session establishment or other system signaling protocols or may be directly transported over UDP or TCP. Where AMIKEY messages are integrated into other LLN-related signaling protocols its transport shall be defined as part of those protocols.

8. Security Considerations

A primary motivation for this RFC is the security that comes from a re-use of the key management methods and framework developed for MIKEY. The extensive deployment and on-going development provides the benefit of much wider vetting and validation essential to assuring greater security.

9. [RFC3830] Groups

10. Additional Specification Considerations

Work had been previously initiated in developing support for an ECC-based asymmetric key management method ([I-D.ietf-msec-mikey-ecc], expired). In the context of LLNs application and subject to IPR considerations, related AMIKEY requirements may be developed.

11. IANA Considerations

This document defines several new name spaces associated with the AMIKEY payloads. This section summarizes the name spaces for which IANA is requested to manage the allocation of values. IANA is requested to record the pre-defined values defined in the given sections for each name space. IANA is also requested to manage the definition of additional values in the future. Unless explicitly stated otherwise, values in the range 0-240 for each name space SHOULD be approved by the process of IETF consensus and values in the range 241-255 are reserved for Private Use, according to [RFC2434].

The name spaces for the new fields identified in this document are requested to be managed by IANA (in bracket is the reference to the table with the initially registered values):

- o Common Header payload (6.1.)
 - * Version
- o Next payload (6.1.b)
 - * Key index
 - * Key source identifier
 - * Key activation time
- o Prf func (6.1.c)
 - * AES-XCBC-PRF-128
- o CS ID map type (6.1.d)
 - * Generic_LLN-ID
- o MAC alg (6.2.b)
 - * AES-CBC-MAC-32
 - * AES-CBC-MAC-64
 - * AES-CBC-MAC-128
- o ID payload (6.7.a)
 - * IPv6 Address
 - * Device MAC Address
- o Proto type (6.10)
 - * Generic_LLN
- o Generic_LLN policy (6.10.2)
 - * Policy parameters (6.10.2.a)
 - * G_LLN encr alg (6.10.2.b)
 - * G_LLN auth alg (6.10.2.c)
 - * G_LLN prf (6.10.2.d)

12. Acknowledgments

The authors would like to acknowledge the review and comments from Rene Struik.

13. References

13.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.
- [RFC3830] Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and K. Norrman, "MIKEY: Multimedia Internet KEYing", RFC 3830, August 2004.
- [RFC4434] Hoffman, P., "The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol (IKE)", RFC 4434, February 2006.

13.2. Informative References

- [I-D.ietf-msec-mikey-ecc] Milne, A., "ECC Algorithms for MIKEY", draft-ietf-msec-mikey-ecc-03 (work in progress), June 2007.
- [I-D.ietf-roll-rpl] Winter, T., Thubert, P., Brandt, A., Clausen, T., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., and J. Vasseur, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks", draft-ietf-roll-rpl-19 (work in progress), March 2011.
- [I-D.ietf-roll-security-framework] Tsao, T., Alexander, R., Dohler, M., Daza, V., and A. Lozano, "A Security Framework for Routing over Low Power and Lossy Networks", draft-ietf-roll-security-framework-06 (work in progress), June 2011.
- [RFC3605] Huitema, C., "Real Time Control Protocol (RTCP) attribute in Session Description Protocol (SDP)", RFC 3605, October 2003.

- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.
- [RFC4107] Bellovin, S. and R. Housley, "Guidelines for Cryptographic Key Management", BCP 107, RFC 4107, June 2005.
- [RFC4442] Fries, S. and H. Tschofenig, "Bootstrapping Timed Efficient Stream Loss-Tolerant Authentication (TESLA)", RFC 4442, March 2006.
- [RFC4563] Carrara, E., Lehtovirta, V., and K. Norrman, "The Key ID Information Type for the General Extension Payload in Multimedia Internet KEYing (MIKEY)", RFC 4563, June 2006.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.
- [RFC4650] Euchner, M., "HMAC-Authenticated Diffie-Hellman for Multimedia Internet KEYing (MIKEY)", RFC 4650, September 2006.
- [RFC4738] Ignjatic, D., Dondeti, L., Audet, F., and P. Lin, "MIKEY-RSA-R: An Additional Mode of Key Distribution in Multimedia Internet KEYing (MIKEY)", RFC 4738, November 2006.
- [RFC5197] Fries, S. and D. Ignjatic, "On the Applicability of Various Multimedia Internet KEYing (MIKEY) Modes and Extensions", RFC 5197, June 2008.
- [RFC5410] Jerichow, A. and L. Piron, "Multimedia Internet KEYing (MIKEY) General Extension Payload for Open Mobile Alliance BCAS 1.0", RFC 5410, January 2009.
- [RFC6043] Mattsson, J. and T. Tian, "MIKEY-TICKET: Ticket-Based Modes of Key Distribution in Multimedia Internet KEYing (MIKEY)", RFC 6043, March 2011.
- [RFC6267] Cakulev, V. and G. Sundaram, "MIKEY-IBAKE: Identity-Based Authenticated Key Exchange (IBAKE) Mode of", RFC 6267, June 2011.

Authors' Addresses

Roger K. Alexander
Cooper Power Systems
20201 Century Blvd. Suite 250
Germantown, Maryland 20874
USA

Email: roger.alexander@cooperindustries.com

Tzeta Tsao
Cooper Power Systems
20201 Century Blvd. Suite 250
Germantown, Maryland 20874
USA

Email: tzeta.tsao@cooperindustries.com

Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: August 27, 2011

M. Goyal, Ed.
University of Wisconsin Milwaukee
E. Baccelli
INRIA
J. Martocci
Johnson Controls
February 23, 2011

Identifying Defunct DAGs in RPL
draft-goyal-roll-defunct-dags-00

Abstract

This document specifies a mechanism for an RPL node to identify defunct directed acyclic graphs.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 27, 2011.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Defunct DAGs	3
2. Terminology	4
3. The No Inconsistency Flag in the DIS Base Object	5
4. Identifying A Defunct DAG	6
5. Security Considerations	7
6. IANA Considerations	7
7. Acknowledgements	7
8. References	7
8.1. Normative References	7
8.2. Informative References	7
Authors' Addresses	8

1. Introduction

RPL [I-D.ietf-roll-rpl], an IPv6 routing protocol for low power and lossy networks (LLNs), allows the formation of directed acyclic graphs (DAGs) in an LLN. These DAGs are used for routing data traffic within the LLN as well as to reach destinations outside the LLN via the DAG root(s). A DAG can be categorized as "grounded" or "floating" based on whether joining the DAG allows a node to meet an application specific goal or not. A DAG can be categorized as "global" or "local" depending on whether the RPL Instance (identified by the RPLInstanceID), to which the DAG belongs, is globally unique or not. A DAG may be permanent in nature or exist temporarily [I-D.ietf-roll-rpl] [I-D.ietf-roll-p2p-rpl]. A DAG is uniquely identified by the combination of its RPLInstanceID, DODAGID and DODAGVersionNumber. A node, running RPL, can join at most one DAG within an RPL Instance.

As described in Section 17.4.2 in [I-D.ietf-roll-rpl], an RPL node needs to maintain a certain state about each DAG it belongs to. This state includes the tuple (RPLInstanceID, DODAGID, DODAGVersionNumber) to identify the DAG, the node's current Rank as well as the minimum Rank (L) the node has had in this DAG, the set of parents the node has in the DAG and the Trickle timers that govern the sending of DODAG Information Object (DIO) messages by the node for the DAG [I-D.ietf-roll-trickle]. This state, except the Trickle timers, needs to be maintained for a certain time duration even when the node has no parent left in the DAG. This is done to ensure that the node does not join an earlier version of the DAG and it does not rejoin the DAG version represented by the DODAGVersionNumber value at a rank higher than $L + \text{DAGMaxRankIncrease}$, where DAGMaxRankIncrease is a configurable RPL parameter [I-D.ietf-roll-rpl].

Given the strict memory constraints faced by nodes in an LLN [RFC5548] [RFC5673] [RFC5826] [RFC5867], it is imperative that RPL protocol has a mechanism that allows a node to identify defunct DAGs and delete the state it maintains for such DAGs. This document specifies such a mechanism.

1.1. Defunct DAGs

An RPL node removes a neighbor from its parent set for a DAG:

- o If the neighbor is no longer reachable, as determined using a mechanism such as Neighbor Unreachability Detection (NUD) [RFC4861], Bidirectional Forwarding Detection (BFD) [RFC5881] or L2 triggers [RFC5184]; or

- o If the neighbor advertises in its DIO an infinite rank in the DAG;
or
- o If keeping the neighbor as a parent would required the node to increase its rank beyond $L + \text{DAGMaxRankIncrease}$; or
- o If the neighbor advertises in its DIO membership in a different DAG within the same RPL Instance, where a different DAG is recognised by a different DODAGID or a different DODAGVersionNumber.

Even if the conditions listed above exist, an RPL node may fail to remove a neighbor from its parent set because:

- o The node fails to receive the neighbor's DIOs advertising an increased rank or the neighbor's membership in a different DAG;
- o The node may not check, and hence may not detect, the neighbor's unreachability for a long time. For example, the node may not have any data to send to this neighbor and hence may not encounter any event (such as failure to send data to this neighbor) that would trigger a check for the neighbor's reachability.

In such cases, a node would continue to consider itself attached to a DAG even if all its parents in the DAG are unreachable or have moved to different DAGs. Such a DAG can be characterized as being defunct from the node's perspective. If the node maintains state about a large number of defunct DAGs, such state may prevent a considerable portion of the total memory in the node from being available for more useful purposes.

To alleviate the problem described above, this document specifies a mechanism for an RPL node to identify the defunct DAGs and delete the state it maintains for such DAGs. Note that, given the proactive nature of RPL protocol, the lack of data traffic using a DAG can not be considered a reliable indication of the DAG's defunction. Further, the Trickle timer based control of DIO transmissions means the possibility of an indefinite delay in the receipt of a new DIO from a functional DAG parent. Hence, the mechanism specified in this document is based on the use of a multicast DODAG Information Solicitation (DIS) message to solicit DIOs about a DAG suspected of defunction.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and

"OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Additionally, this document uses terminology from [I-D.ietf-roll-terminology] and [I-D.ietf-roll-rpl]. Specifically, the term RPL node refers to an RPL router or an RPL host as defined in [I-D.ietf-roll-rpl].

3. The No Inconsistency Flag in the DIS Base Object

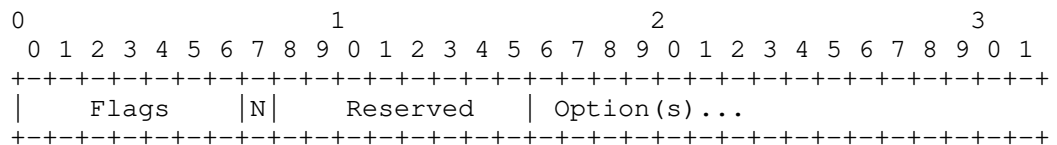


Figure 1: The No Inconsistency Flag in DIS Base Object

An RPL node can use a DODAG Information Solicitation (DIS) message to solicit DODAG Information Object (DIO) messages from its neighbors. A DIS may carry a Solicited Information option that specifies the predicates of the DAG(s) the node is interested in. In the absence of a Solicited Information option, it is assumed that the node generating the DIS is interested in receiving DIOs for all the DAGs. In the following discussion, we use the term "DIS predicates" to refer to both cases. If the DIS does not contain a Solicited Information option, all DAGs will match the DIS predicates; otherwise only those DAGs match the DIS predicates that satisfy the predicates specified in the Solicited Information option contained in the DIS.

A DIS can be multicast to all the in-range neighbors or it can be unicast to a specific neighbor. Unless restricted by a DIS flag, an RPL node must consider the receipt of a multicast DIS as an inconsistency and hence reset its Trickle timers [I-D.ietf-roll-trickle] for the DAGs that match the DIS predicates. The receipt of a unicast DIS causes an RPL node to generate the DIOs for all the DAGs matching the DIS predicates without resetting the Trickle timers.

This document defines a "No Inconsistency" (N) flag inside the DIS base object. The modified DIS base object format is shown in Figure 1. An RPL node, generating a DIS, MUST set this flag if it solicits DIOs for the purpose of identifying the defunct DAGs as specified in this document. On receiving a unicast/multicast DIS with N flag set, an RPL node MUST NOT reset the trickle timers for the DAGs that match the DIS predicates. For each DAG matching the

predicates of a multicast DIS received with N flag set, an RPL node MUST schedule a DIO transmission after a time duration between $I_{min}/2$ and I_{min} , where I_{min} is the minimum Trickle interval size [I-D.ietf-roll-trickle] associated with the DAG. For each DAG matching the predicates of a unicast DIS received with N flag set, an RPL node MUST immediately generate a DIO.

4. Identifying A Defunct DAG

When an RPL node has not received a DIO from any of its parents in a DAG for more than $MaxSilence * I_{max}$ seconds, where $MaxSilence$ is a configurable parameter greater than 1 and I_{max} is the maximum Trickle interval size [I-D.ietf-roll-trickle] associated with the DAG:

- o The node MUST generate a multicast DIS message that carries a Solicited Information option and has N flag set. The Solicited Information option MUST have the I and D flags set and the RPLInstanceID/DODAGID fields MUST be set to values identifying the DAG. The V flag inside the Solicited Information option SHOULD NOT be set so as to allow neighbors to send DIOs advertising the latest version of the DAG.
- o After sending the DIS, the node MUST wait for I_{min} duration, where I_{min} is the minimum Trickle interval size associated with the DAG, to receive the DIOs generated by its neighbors.
- o At the conclusion of the wait period:
 - * If the node has received one or more DIOs advertising newer version(s) of the DAG, it MUST join the latest version of the DAG, select a new parent set among the neighbors advertising the latest DAG version and mark the DAG status as functional.
 - * Otherwise, if the node has not received a DIO advertising the current version of the DAG from a neighbor in the parent set, it MUST remove that neighbor from the parent set. As a result, if the node has no parent left in the DAG, it MUST mark the DAG as defunct and schedule the deletion of the state it has maintained for the DAG after DAGHoldTime duration, a configurable parameter.

An RPL node SHOULD check the functional status of a DAG it belongs to in the manner described above at least once during a CheckDAGStatusTime interval, which is a configurable parameter.

5. Security Considerations

TBA

6. IANA Considerations

TBA

7. Acknowledgements

We gratefully acknowledge Thomas Clausen for motivating this draft.

8. References

8.1. Normative References

[I-D.ietf-roll-rpl]

Winter, T., Thubert, P., Brandt, A., Clausen, T., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., and J. Vasseur, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks", draft-ietf-roll-rpl-18 (work in progress), February 2011.

[I-D.ietf-roll-trickle]

Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko, "The Trickle Algorithm", draft-ietf-roll-trickle-08 (work in progress), January 2011.

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

8.2. Informative References

[I-D.ietf-roll-p2p-rpl]

Goyal, M., Baccelli, E., Brandt, A., Cragie, R., Martocci, J., and C. Perkins, "Reactive Discovery of Point-to-Point Routes in Low Power and Lossy Networks", draft-ietf-roll-p2p-rpl-02 (work in progress), February 2011.

[I-D.ietf-roll-terminology]

Vasseur, J., "Terminology in Low power And Lossy Networks", draft-ietf-roll-terminology-04 (work in progress), September 2010.

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

Authors' Addresses

Mukul Goyal (editor)
University of Wisconsin Milwaukee
3200 N Cramer St
Milwaukee, WI 53201
USA

Phone: +1 414 2295001
Email: mukul@uwm.edu

Emmanuel Baccelli
INRIA

Phone: +33-169-335-511
Email: Emmanuel.Baccelli@inria.fr
URI: <http://www.emmanuelbaccelli.org/>

Jerald Martocci
Johnson Controls
507 E Michigan St
Milwaukee, WI 53202
USA

Phone: +1 414-524-4010
Email: jerald.p.martocci@jci.com

Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: August 27, 2011

M. Goyal, Ed.
University of Wisconsin Milwaukee
E. Baccelli
INRIA
J. Martocci
Johnson Controls
February 23, 2011

The Direction Field in Routing Metric/Constraint Objects Used in RPL
draft-goyal-roll-metrics-direction-00

Abstract

This document specifies a Direction field in the Routing Metric/Constraint objects used in RPL operation in low power and lossy networks.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 27, 2011.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as

described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Terminology	3
3. The Direction Field	4
4. Security Considerations	5
5. IANA Considerations	5
6. References	5
6.1. Normative References	5
6.2. Informative References	5
Authors' Addresses	6

1. Introduction

Asymmetric links are a common observation in low power and lossy networks (LLNs) [sang_2010]. Many link-level routing metrics have a directional aspect. Although such routing metrics can be defined in a bidirectional manner so as to account for the link properties in both directions, this is not always desirable. In the context of RPL [I-D.ietf-roll-rpl], the IPv6 routing protocol for LLNs, it may be necessary to measure a link-level routing metric in a particular direction. For example, if the intent is to build a directional acyclic graph (DAG) specifically for the purpose of low latency communication to the DAG root, the routing metric must measure the link latency in Up direction, i.e., towards the DAG root, as defined in [I-D.ietf-roll-rpl]. Similarly, if a temporary DAG is being constructed to discover a point-to-point route towards a destination [I-D.ietf-roll-p2p-rpl], the routing metric must calculate the relevant link characteristic in Down direction, i.e., away from the DAG root, as defined in [I-D.ietf-roll-rpl]. Thus, there is a need to specify the directional aspect of a link-level routing metric.

Accordingly, this document defines a Direction field inside the Routing Metric/Constraint object header, defined in [I-D.ietf-roll-routing-metrics]. The Direction field is defined in two previously reserved bits inside the Routing Metric/Constraint object header. The modified Routing Metric/Constraint object header is backward compatible with its definition in [I-D.ietf-roll-routing-metrics].

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Additionally, this document uses terminology from [I-D.ietf-roll-terminology] and [I-D.ietf-roll-rpl]. Specifically, the term RPL node refers to an RPL router or an RPL host as defined in [I-D.ietf-roll-rpl].

3. The Direction Field

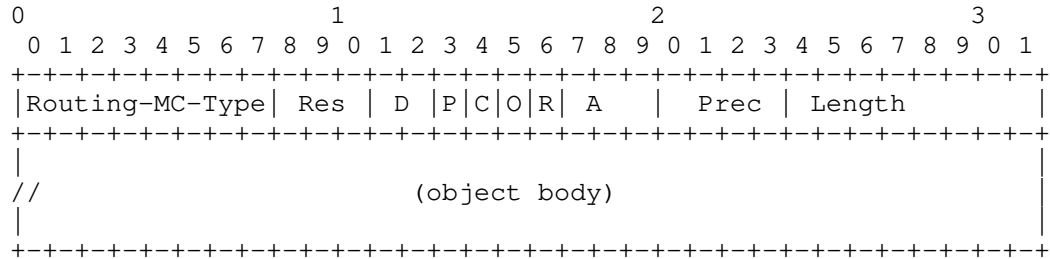


Figure 1: Routing Metric/Constraint object generic format

The modified Routing Metric/Constraint object header is illustrated in Figure 1. The Direction (or D) field is a 2-bit field that indicates the direction associated with the routing metric/constraint:

- o D = 0x00: undefined;
- o D = 0x01: Up;
- o D = 0x02: Down;
- o D = 0x03: Bidirectional.

If the D field has value 0x00, the direction associated with the routing metric/constraint is undefined as in [I-D.ietf-roll-routing-metrics]. A value 0x00 for the D field may be suitable for node-level routing metrics/constraints defined in [I-D.ietf-roll-routing-metrics]. The D field value in link-level routing metrics/constraints SHOULD NOT be set to 0x00.

This document does not specify how to measure/evaluate a routing metric/constraint object in the direction specified by the D field. The measurement/evaluation methodology for specific routing metrics/constraints, taking in account the D field, may be specified in a separate document.

A routing metric/constraint object MUST be measured/evaluated in accordance with its D field value if defined. In case, an RPL node can not measure/evaluate the routing metric/constraint object in the specified direction, the following rules MUST be applied:

- o If the object is a recorded metric, i.e., has C=0 and R=1 fields, the RPL node MUST set the P flag inside the object, thereby

indicating the partial nature of the recorded metric.

- o If the object is an aggregated metric, i.e., has C=0 and R=0 fields, the RPL node MUST drop the DIO containing the object.
- o If the object is a mandatory constraint, i.e., has C=1 and O=0 fields, the RPL node MUST drop the DIO containing the object.
- o If the object is an optional constraint, i.e., has C=1 and O=1 fields, the RPL node MAY drop the DIO containing the object or it MAY continue processing rest of the DIO ignoring this object.

4. Security Considerations

TBA

5. IANA Considerations

This document does not have any IANA considerations.

6. References

6.1. Normative References

- [I-D.ietf-roll-routing-metrics]
Vasseur, J., Kim, M., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics used for Path Calculation in Low Power and Lossy Networks", draft-ietf-roll-routing-metrics-17 (work in progress), January 2011.
- [I-D.ietf-roll-rpl]
Winter, T., Thubert, P., Brandt, A., Clausen, T., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., and J. Vasseur, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks", draft-ietf-roll-rpl-18 (work in progress), February 2011.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

6.2. Informative References

- [I-D.ietf-roll-p2p-rpl]
Goyal, M., Baccelli, E., Brandt, A., Cragie, R., Martocci,

J., and C. Perkins, "Reactive Discovery of Point-to-Point Routes in Low Power and Lossy Networks", draft-ietf-roll-p2p-rpl-02 (work in progress), February 2011.

[I-D.ietf-roll-terminology]

Vasseur, J., "Terminology in Low power And Lossy Networks", draft-ietf-roll-terminology-04 (work in progress), September 2010.

[sang_2010]

Sang, L., Arora, A., and H. Zhang, "On Link Asymmetry and One-way Estimation in Wireless Sensor Networks", ACM Transactions on Sensor Networks Volume 6, Number 2, February 2010.

Authors' Addresses

Mukul Goyal (editor)
University of Wisconsin Milwaukee
3200 N Cramer St
Milwaukee, WI 53201
USA

Phone: +1 414 2295001
Email: mukul@uwm.edu

Emmanuel Baccelli
INRIA

Phone: +33-169-335-511
Email: Emmanuel.Baccelli@inria.fr
URI: <http://www.emmanuelbaccelli.org/>

Jerald Martocci
Johnson Controls
507 E Michigan St
Milwaukee, WI 53202
USA

Phone: +1 414-524-4010
Email: jerald.p.martocci@jci.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: November 29, 2011

M. Goyal, Ed.
University of Wisconsin
Milwaukee
E. Baccelli
M. Philipp
INRIA
A. Brandt
Sigma Designs
J. Martocci
Johnson Controls
May 28, 2011

A Compression Format for RPL Control Messages
draft-goyal-roll-rpl-compression-00

Abstract

This document specifies a compression format for RPL ICMPv6 RPL control messages.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 29, 2011.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Terminology	3
2. RPL ICMPv6 Message Compression	3
3. RPL Base Object Compression	4
3.1. Compressing the DODAG Information Object	4
4. Compressing the RPL Options	6
4.1. DODAG Configuration Option	6
4.2. Metric/Constraint Objects	8
4.2.1. Compressed Node State and Attributes Object	9
4.2.2. Compressed Node Energy Object	9
4.2.3. Compressed Hop Count Object	10
4.2.4. Compressed Throughput Object	10
4.2.5. Compressed Latency Object	10
4.2.6. Compressed ETX Object	10
5. Examples	11
5.1. A DIO With A Configuration Option, A Route Information Option and A Metric Container	11
5.2. A DIO With A Configuration Option, A Route Discovery Option and A Metric Container	12
6. Security Considerations	14
7. IANA Considerations	14
8. References	14
8.1. Normative References	14
8.2. Informative References	14
Authors' Addresses	15

1. Introduction

RPL [I-D.ietf-roll-rpl] is an IPv6 routing protocol for low power and lossy networks. It defines a number of ICMPv6 control messages for its operation. These messages are susceptible to fragmentation when RPL is deployed over a link layer with a small payload (e.g. IEEE 802.15.4, where the MAC payload can be as small as 81 bytes). This document specifies a compression format for ICMPv6 RPL control messages to minimize such fragmentation. This document currently defines the compression format for RPL's DODAG Information Object (DIO) base object, DODAG Configuration Option and some of the Routing Metric/Constraint objects. Later versions of this document may include the compression formats for other RPL messages and options.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Additionally, this document uses terminology from [I-D.ietf-roll-rpl], [I-D.ietf-roll-routing-metrics], [I-D.ietf-roll-p2p-rpl] and [I-D.ietf-6lowpan-hc].

2. RPL ICMPv6 Message Compression

Various fields of a compressed ICMPv6 messages are as follows:

- o Type: 155, as specified in [I-D.ietf-roll-rpl];
- o Code: The Code value of the compressed version of an RPL ICMPv6 message is obtained by setting the 7th bit in the Code value associated with the corresponding uncompressed message. For example, the Code associated with a compressed DODAG Information Object is 0x40.
- o Checksum: The 16-bit Checksum for a compressed RPL message is calculated in the same manner as for the uncompressed message.
- o Base: The Base object carried in the message is compressed in the manner described in Section 3.
- o Option(s): An option carried in a compressed RPL ICMPv6 message MAY be compressed as described in Section 4 or it MAY be carried uncompressed as in an uncompressed message.

3. RPL Base Object Compression

This section defines the compression format for various Base objects associated with RPL ICMPv6 messages.

3.1. Compressing the DODAG Information Object

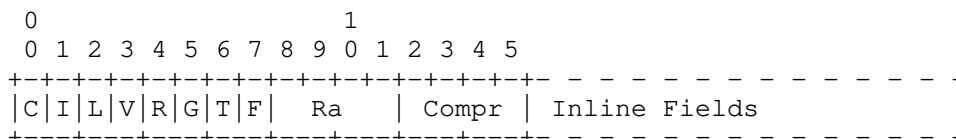


Figure 1: The Compression Format for DODAG Information Object

The format of a compressed DODAG Information Object (DIO) base object is shown in Figure 1 and consists of the following fields:

- o C: This flag is set to 1 if the 3rd byte of the DIO contains an 8-bit identifier of the "context" that provides values of all elided fields in the DIO base object and the options it contains. Otherwise, this flag is set to 0. The description of the contexts in use in an LLN and how RPL nodes come to know of these contexts is out of scope for this document.
- o I: This flag indicates whether the RPLInstanceID field is elided or not. This flag is set to 1 if the RPLInstanceID field is present inline in the compressed DIO. This flag is set to 0 if the RPLInstanceID field is elided. In this case, the implicit value of the RPLInstanceID depends on the context if present. In the absence of the context, the implicit value of the RPLInstanceID depends on the value of the L flag as discussed next.
- o L: This flag is meaningful only when both C and I flags are 0, i.e. the RPLInstanceID field is elided but no context is identified. In this case, the L flag is set to 1 if the elided RPLInstanceID is local and has implicit value 128. The L flag is set to 0 if the elided RPLInstanceID field is global and has implicit value 0. If either of C and I flag is set to 1, the L flag MUST be set to 0 on transmission and ignored on reception.
- o V: This flag is set to 1 if the Version Number is carried inline in the DIO message. The flag is set to 0 if the Version Number is elided. In this case, the implicit value of the Version Number depends on the context if present. If no context is present, the implicit value of the Version Number is assumed to be zero.

- o R: This flag indicates whether the Rank field in the DIO is shortened or not. This flag is set to 1 if the full 16-bit Rank is present inline in the compressed DIO. The flag is set to 0 if the 4-bit long Ra field contains the rank value.
- o G: This flag indicates whether the byte containing the Grounded, Mode of Operation and DODAG Preference fields is elided or not. This flag is set to 1 if the above-mentioned byte is carried inline. The flag is set to 0, if this byte is elided. In this case, the implicit values of Grounded, Mode of Operation and DODAGPreference fields depend on the context if present. If no context is present, the implicit values of these fields are as follows:
 - * The Grounded flag has implicit value 0, i.e., the DODAG is not grounded.
 - * The Mode of Operation field has implicit value 0, i.e., the DODAG does not maintain any downward routes.
 - * The DODAG Preference field has implicit value 0, i.e., least preferred.
- o T: This flag is set to 1 if the DTSN field is carried inline. The flag is set to 0, if the DTSN field is elided. In this case, the implicit value of the DTSN field depends on the context if present. If no context is present, the implicit value of this field is assumed to be zero.
- o F: This flag indicates whether the Flags and Reserved fields in the DIO are elided or not. This flag is set to 1 if these fields are carried inline. The flag is set to 0, if these fields are elided. In this case, the values of these fields depend on the context if present. If no context is present, both fields are assumed to be zero.
- o Ra: This field contains the 4-bit rank value if the R flag is set to 0.
- o Compr: This field contains the number of prefix octets that are elided from the DODAGID field. For example, the Compr value will be zero if full 16-octet DODAGID field is carried inline in the compressed DIO.
- o Inline Fields: The context identifier, if present, occupies the 3rd byte of the compressed DIO base object. Any inline fields in the compressed DIO appear in the same order as in the uncompressed format defined in [I-D.ietf-roll-rpl].

4. Compressing the RPL Options

This section defines the compression format for some of the RPL options that may be carried inside an RPL control message. These RPL options SHOULD be compressed when carried inside an RPL control message compressed in the manner described in this document. The other RPL options, for which a compression format is not specified in this document, MUST follow the format in which they are defined when carried inside an RPL control message compressed as described in this document.

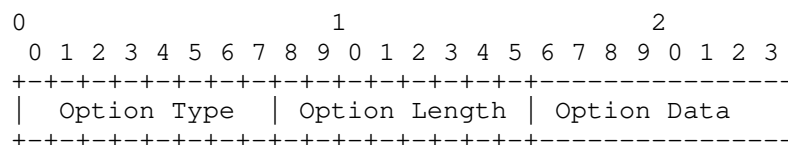


Figure 2: Format of a Compressed RPL Option

The compression format of an RPL option is shown in Figure 2. It consists of:

- o Option Type: The Option Type value for a compressed RPL option is same as that of the uncompressed option with the most significant bit (MSB) set to 1.
- o Option Length: The Option Length is 8 bits long as in case of an uncompressed RPL option.

4.1. DODAG Configuration Option

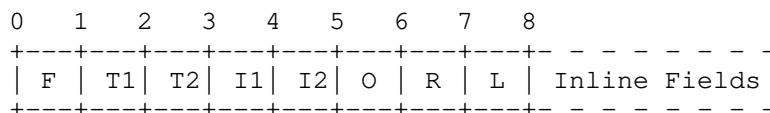


Figure 3: Format of a Compressed DODAG Configuration Option

The format of the compressed DODAG Configuration Option is shown in Figure 3. The compressed DODAG Configuration option begins with an octet consisting of flags that specify whether the individual fields in the option are elided or not. The implicit value of an elided field depends on the context identified in the DIO base object. If DIO base object does not identify a context, the implicit value of an elided field is as specified below:

- o F: This flag indicates whether the byte in the uncompressed DODAG Configuration option, consisting of the Flags, A and PCS fields, is elided or not. This flag is set to 1 if this byte is carried inline. The flag is set to 0, if this byte is elided. If the DIO base object does not contain a context, the implicit values of elided A and PCS fields are zero and DEFAULT_PATH_CONTROL_SIZE (as defined in [I-D.ietf-roll-rpl]) respectively.
- o T1: This flag indicates whether the DIOIntervalDoublings and DIOIntervalMin fields are elided or not. This flag is set to 1 if these fields are carried inline. The flag is set to 0, if these fields are elided. If DIO base object does not contain a context identifier, these fields, if elided, assume their default values as defined in [I-D.ietf-roll-rpl].
- o T2: This flag indicates whether the DIORedundancyConstant field is elided or not. This flag is set to 1 if DIORedundancyConstant is carried inline. The flag is set to 0, if this field is elided. In this case, the field assumes its default value as defined in [I-D.ietf-roll-rpl] if the DIO base object does not identify a context.
- o I1: This flag indicates whether the MaxRankIncrease field is elided or not. This flag is set to 1 if this field is carried inline. The flag is set to 0 if this field is elided. In this case, the MaxRankIncrease field assumes its default value (as defined in [I-D.ietf-roll-rpl]) if the DIO base object does not identify a context.
- o I2: This flag indicates whether the MinHopRankInc field is elided or not. This flag is set to 1 if this field is carried inline. The flag is set to 0 if this field is elided. In this case, the MinHopRankInc field assumes its default value (as defined in [I-D.ietf-roll-rpl]) if the DIO base object does not identify a context.
- o O: This flag indicates whether the OCP field is elided or not. This flag is set to 1 if this field is carried inline. The flag is set to 0 if this field is elided. In this case, if the DIO base object does not identify a context, RPL Objective Function 0 [I-D.ietf-roll-of0] is the OCP in effect.
- o R: This flag indicates whether the byte marked as Reserved in the uncompressed format is elided or not. This flag is set to 1 if this byte is carried inline. The flag is set to 0 if this byte is elided. In this case, the Reserved byte is assumed to have value 0.

- o L: This flag indicates whether the Default Lifetime and Lifetime Unit fields are elided or not. This flag is set to 1 if these fields are carried inline. The flag is set to 0 if these fields are elided. In this case, the life time of the routes associates with this DODAG is infinity unless another value is specified in the context identified in the DIO base object.
- o Inline fields: Any inline fields in the compressed DODAG Configuration option appear in the same order as in the uncompressed format.

Note that a compressed DODAG Configuration Option can be as small as 3 bytes, whereas an uncompressed DODAG Configuration Option is 16 bytes long.

4.2. Metric/Constraint Objects

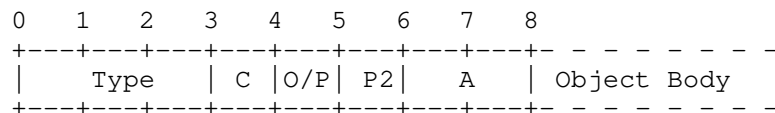


Figure 4: Generic Format of a Compressed Routing Metric/Constraint Object

A compressed Metric Container Option contains compressed Routing Metric/Constraint objects as defined in this document. A compressed Metric Container Option MUST NOT contain uncompressed Routing Metric/Constraint objects defined in [I-D.ietf-roll-routing-metrics]. The generic format of a compressed Routing Metric/Constraint Object is shown in Figure 4. A compressed Routing Metric/Constraint Object always has a fixed size as defined in this document. Thus, "recorded" metrics and sub-objects/TLV options within a metric object are not allowed. Various fields inside a compressed Routing Metric/Container Object header are as follows:

- o Type: The type of the routing metric/constraint object.
- o C: This flag is set to one if the object represents a constraint. This flag is set to zero if the object represents a metric.
- o O/P: If the object represents a constraint, this flag is set to one if the constraint is optional. Otherwise, the flag is set to zero. If the object represents a metric, this bit represents, along with P2 bit, a 2-bit "precedence" field.
- o P2: This bit is relevant only when the object represents a metric. Along with the O/P bit, this bit forms a 2-bit "precedence" field.

to indicate the precedence of this metric relative to other metrics in the container. The precedence values range from 0 to 3, 0 being the highest precedence.

- o A: This field is relevant only for metrics and indicates the manner in which the routing metric must be aggregated:

- * A=0x00: The routing metric is additive
- * A=0x01: The routing metric reports a maximum
- * A=0x02: The routing metric reports a minimum
- * A=0x03: The routing metric is multiplicative

4.2.1. Compressed Node State and Attributes Object

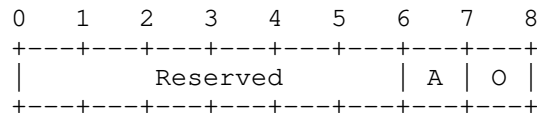


Figure 5: Compressed Node State and Attributes Object

The compressed Node State and Attributes object has Type value 0 and is shown in Figure 5. The A and O flags in the object have same meaning and function as the corresponding flags in the uncompressed object defined in [I-D.ietf-roll-routing-metrics].

4.2.2. Compressed Node Energy Object

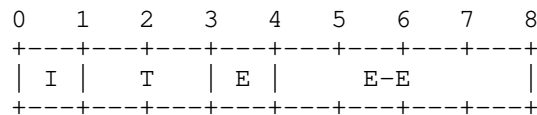


Figure 6: Compressed Node Energy Object

The compressed Node Energy object has Type value 1 and is shown in Figure 6. Various fields in the object have same meaning and function as the corresponding fields in Node Energy sub-object defined in [I-D.ietf-roll-routing-metrics]. Note that the E-E field has been reduced from 8 bits to 4 bits.

4.2.3. Compressed Hop Count Object

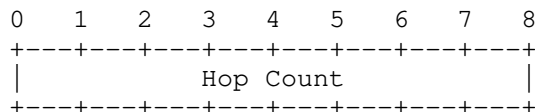


Figure 7: Compressed Hop Count Object

The compressed Hop Count object has Type value 2 and is shown in Figure 7. It consists of a 8-bit hop count value.

4.2.4. Compressed Throughput Object

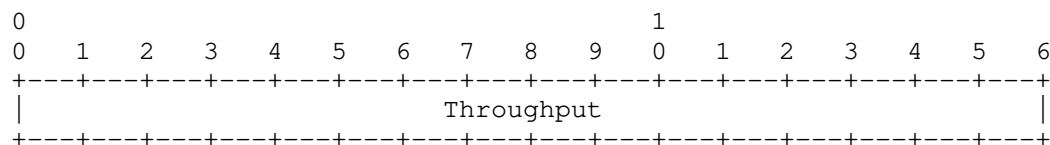


Figure 8: Compressed Throughput Object

The compressed Throughput object has Type value 3 and is shown in Figure 8. It consists of a 16-bit value expressed in units of kilo bytes per second.

4.2.5. Compressed Latency Object

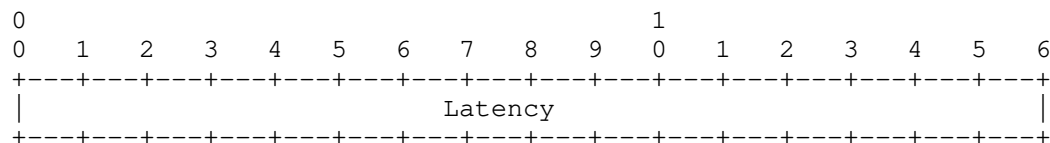


Figure 9: Compressed Latency Object

The compressed Latency object has Type value 4 and is shown in Figure 9. It consists of a 16-bit value expressed in units of milliseconds.

4.2.6. Compressed ETX Object

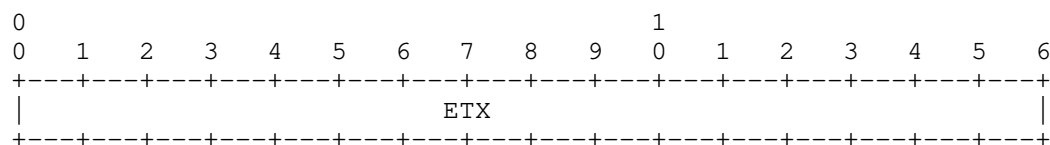


Figure 10: Compressed ETX Object

The compressed ETX object has Type value 5 and is shown in Figure 10. It consists of a 16-bit value, which has the same meaning and function as the ETX field inside ETX sub-object inside ETX object defined in [I-D.ietf-roll-routing-metrics].

5. Examples

In this section, we compare the sizes of RPL control messages with and without the compression mechanism specified in this document.

5.1. A DIO With A Configuration Option, A Route Information Option and A Metric Container

Consider an uncompressed multicast DIO message that has a Configuration Option, a Route Information Option and a Metric Container with one ETX metric object and one ETX constraint object. This message consists of the following components:

- o IPv6 header: A typical IPv6 header, compressed as per [I-D.ietf-6lowpan-hc], for a multicast DIO message consists of 5 bytes as follows:
 - * 2 byte LOWPAN_IPHC Base Encoding
 - * 1 byte Context Identifier Extension
 - * 1 byte Next Header
 - * 1 byte Group ID to identify all-RPL-nodes multicast address.
- o 4 bytes for ICMP Type, Code and Checksum fields;
- o 24 bytes for DIO Base Object;
- o 16 bytes for DODAG Configuration Option;
- o 24 byte Route Information Option;
- o 14 bytes for Metric Container consisting of:
 - * 2 bytes for Type and Option Length fields;
 - * 6 bytes for ETX metric object (4 bytes header + 2 bytes body);

- * 6 bytes for ETX constraint object (4 bytes header + 2 bytes body).

Thus, the total length of such a DIO is 87 bytes.

The same message, when compressed in the manner described in this document, consists of:

- o 5 bytes of IPv6 header compressed as per [I-D.ietf-6lowpan-hc] in the manner described in the previous paragraph;
- o 4 bytes for ICMP Type, Code and Checksum fields;
- o 4 bytes of compressed DIO Base Object consisting of 2 bytes of header and 2 bytes for DODAGID (the best case scenario);
- o 3 bytes of compressed DODAG Configuration Option, including 2 bytes for Type and Option Length fields;
- o 24 bytes of Route Information Option;
- o 8 bytes for Metric Container consisting of:
 - * 2 bytes for Type and Option Length fields;
 - * 3 bytes for ETX metric object (1 byte header + 2 bytes body);
 - * 3 bytes for ETX constraint object (1 byte header + 2 bytes body).

Thus, the total length of the compressed DIO is 48 bytes.

5.2. A DIO With A Configuration Option, A Route Discovery Option and A Metric Container

Consider an uncompressed multicast DIO message that has a Configuration Option, a Route Discovery Option (defined in [I-D.ietf-roll-p2p-rpl]) and a Metric Container with one ETX metric object and one ETX constraint object. This message consists of the following components:

- o 5 bytes of IPv6 header compressed as per [I-D.ietf-6lowpan-hc] in the manner described in the previous section;
- o 4 bytes for ICMP Type, Code and Checksum fields;
- o 24 bytes for DIO Base Object;

- o 16 bytes for DODAG Configuration Option;
- o 26 bytes for Route Discovery Option consisting of:
 - * 4 bytes for Type, Option Length and other fixed length fields;
 - * 2 bytes for the Target address field;
 - * 20 bytes for the Address vector (assuming 10 2-byte long elements).
- o 14 bytes for Metric Container consisting of:
 - * 2 bytes for Type and Option Length fields;
 - * 6 bytes for ETX metric object (4 bytes header + 2 bytes body);
 - * 6 bytes for ETX constraint object (4 bytes header + 2 bytes body).

Thus, the total length of such a DIO is 89 bytes.

The same message, when compressed in the manner described in this document, consists of:

- o 5 bytes of IPv6 header compressed as per [I-D.ietf-6lowpan-hc];
- o 4 bytes for ICMP Type, Code and Checksum fields;
- o 4 bytes of compressed DIO Base Object consisting of 2 bytes of header and 2 bytes for DODAGID (the best case scenario);
- o 3 bytes of compressed DODAG Configuration Option, including 2 bytes for Type and Option Length fields;
- o 26 bytes of Route Information Option;
- o 8 bytes for Metric Container consisting of:
 - * 2 bytes for Type and Option Length fields;
 - * 3 bytes for ETX metric object (1 byte header + 2 bytes body);
 - * 3 bytes for ETX constraint object (1 byte header + 2 bytes body).

Thus, the total length of the compressed DIO is 50 bytes.

6. Security Considerations

TBA

7. IANA Considerations

TBA

8. References

8.1. Normative References

[I-D.ietf-roll-routing-metrics]

Vasseur, J., Kim, M., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics used for Path Calculation in Low Power and Lossy Networks", draft-ietf-roll-routing-metrics-19 (work in progress), March 2011.

[I-D.ietf-roll-rpl]

Winter, T., Thubert, P., Brandt, A., Clausen, T., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., and J. Vasseur, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks", draft-ietf-roll-rpl-19 (work in progress), March 2011.

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

8.2. Informative References

[I-D.ietf-6lowpan-hc]

Hui, J. and P. Thubert, "Compression Format for IPv6 Datagrams in Low Power and Lossy Networks (6LoWPAN)", draft-ietf-6lowpan-hc-15 (work in progress), February 2011.

[I-D.ietf-roll-of0]

Thubert, P., "RPL Objective Function 0", draft-ietf-roll-of0-12 (work in progress), May 2011.

[I-D.ietf-roll-p2p-rpl]

Goyal, M., Baccelli, E., Brandt, A., Cragie, R., and J. Martocci, "Reactive Discovery of Point-to-Point Routes in Low Power and Lossy Networks", draft-ietf-roll-p2p-rpl-03 (work in progress), May 2011.

[I-D.ietf-roll-terminology]

Vasseur, J., "Terminology in Low power And Lossy Networks", draft-ietf-roll-terminology-05 (work in progress), March 2011.

Authors' Addresses

Mukul Goyal (editor)
University of Wisconsin Milwaukee
3200 N Cramer St
Milwaukee, WI 53211
USA

Phone: +1 414 2295001
Email: mukul@uwm.edu

Emmanuel Baccelli
INRIA

Phone: +33-169-335-511
Email: Emmanuel.Baccelli@inria.fr
URI: <http://www.emmanuelbaccelli.org/>

Matthias Philipp
INRIA

Email: matthias.philipp@inria.fr

Anders Brandt
Sigma Designs

Phone: +45-29609501
Email: abr@sdesigns.dk

Jerald Martocci
Johnson Controls

Phone: +1-414-524-4010
Email: jerald.p.martocci@jci.com

Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: October 3, 2013

M. Goyal, Ed.
University of Wisconsin
Milwaukee
E. Baccelli
INRIA
A. Brandt
Sigma Designs
J. Martocci
Johnson Controls
April 1, 2013

A Mechanism to Measure the Routing Metrics along a Point-to-point Route
in a Low Power and Lossy Network
draft-ietf-roll-p2p-measurement-10

Abstract

This document specifies a mechanism that enables an RPL router to measure the aggregated values of given routing metrics along an existing route towards another RPL router, thereby allowing the router to decide if it wants to initiate the discovery of a better route.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 3, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Terminology	4
2. Overview	4
3. The Measurement Object (MO)	6
3.1. Format of the base MO	6
3.2. Secure MO	11
4. Originating a Measurement Request	11
4.1. When Measuring A Hop-by-hop Route with a Global RPLInstanceID	12
4.2. When Measuring A Hop-by-hop Route with a Local RPLInstanceID With Route Accumulation Off	13
4.3. When Measuring A Hop-by-hop Route with a Local RPLInstanceID With Route Accumulation On	14
4.4. When Measuring A Source Route	16
5. Processing a Measurement Request at an Intermediate Point	17
5.1. When Measuring A Hop-by-hop Route with a Global RPLInstanceID	18
5.2. When Measuring A Hop-by-hop Route with a Local RPLInstanceID With Route Accumulation Off	19
5.3. When Measuring A Hop-by-hop Route with a Local RPLInstanceID With Route Accumulation On	20
5.4. When Measuring A Source Route	21
5.5. Final Processing	21
6. Processing a Measurement Request at the End Point	22
6.1. Generating the Measurement Reply	23
7. Processing a Measurement Reply at the Start Point	23
8. Security Considerations	24
9. IANA Considerations	25
10. Acknowledgements	26
11. References	26
11.1. Normative References	26
11.2. Informative References	27
Authors' Addresses	27

1. Introduction

Point to point (P2P) communication between arbitrary routers in a Low power and Lossy Network (LLN) is a key requirement for many applications [RFC5826][RFC5867]. The IPv6 Routing Protocol for LLNs (RPL) [RFC6550] constrains the LLN topology to a Directed Acyclic Graph (DAG) built to optimize the routing costs to reach the DAG's root. The P2P routing functionality, available under RPL, has the following key limitations:

- o The P2P routes are restricted to use the DAG links only. Such P2P routes may potentially be suboptimal and may lead to traffic congestion near the DAG root.
- o RPL is a proactive routing protocol and hence requires all P2P routes to be established ahead of the time they are used. Many LLN applications require the ability to establish P2P routes "on demand".

To ameliorate situations where the core RPL's P2P routing functionality does not meet the application requirements [I-D.ietf-roll-p2p-rpl] describes P2P-RPL, an extension to core RPL. P2P-RPL provides a reactive mechanism to discover P2P routes that meet the specified routing constraints [RFC6551]. In some cases, the application requirements or the LLN's topological features allow a router to infer these routing constraints implicitly. For example, the application may require the end-to-end loss rate and/or latency along the route to be below certain thresholds or the LLN topology may be such that a router can safely assume its destination to be less than a certain number of hops away from itself.

When the existing routes are deemed unsatisfactory but the router does not implicitly know the routing constraints to be used in P2P-RPL route discovery, it may be necessary for the router to measure the aggregated values of the routing metrics along the existing route. This knowledge will allow the router to frame reasonable routing constraints to discover a better route using P2P-RPL. For example, if the router determines the aggregate ETX (Expected Number of Transmissions) [RFC6551] along an existing route to be "x", it can use " $ETX < x*y$ ", where y is a certain fraction, as the routing constraint for use in P2P-RPL route discovery. Note that it is important that the routing constraints not be overly strict; otherwise, the P2P-RPL route discovery may fail even though a route exists that is much better than the one currently being used.

This document specifies a mechanism that enables an RPL router to measure the aggregated values of the routing metrics along an existing route to another RPL router in an LLN, thereby allowing the

router to decide if it wants to discover a better route using P2P-RPL and determine the routing constraints to be used for this purpose. Thus, the utility of this mechanism is dependent on the existence of P2P-RPL, which is targeting publication as an Experimental RFC. It makes sense, therefore, for this document also to target publication as an Experimental RFC. The hope is that experiments with P2P-RPL and the mechanism defined in this document will result in feedback on the utility and benefits of this document and it will be revised and progressed on the Standards Track based on this feedback.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

This document uses terminology from [RFC6550], [I-D.ietf-roll-terminology] and [I-D.ietf-roll-p2p-rpl]. Additionally, this document defines the following terms.

Start Point: The Start Point refers to the RPL router that initiates the measurement process defined in this document and is the start point of the P2P route being measured.

End Point: The End Point refers to the RPL router at the end point of the P2P route being measured.

Intermediate Point: An RPL router, other than the Start Point and the End Point, on the P2P route being measured.

The following terms, already defined in [I-D.ietf-roll-p2p-rpl], have been redefined in this document in the following manner.

Forward direction: The direction from the Start Point to the End Point.

Reverse direction: The direction from the End Point to the Start Point.

2. Overview

The mechanism described in this document can be used by a Start Point in an LLN to measure the aggregated values of selected routing metrics along a P2P route to an End Point within the LLN. The route is measured in the Forward direction. Such a route could be a Source Route or a Hop-by-hop Route established using RPL [RFC6550] or P2P-

RPL [I-D.ietf-roll-p2p-rpl]. Such a route could also be a "mixed" route with the initial part consisting of hop-by-hop ascent to the root of a non-storing DAG [RFC6550] and the final part consisting of a source-routed descent to the End Point. The Start Point decides what metrics to measure and sends a Measurement Request message, carrying the desired routing metric objects, along the route. If a Source Route is being measured, the Measurement Request carries the route inside an Address vector. If a Hop-by-hop Route is being measured, the Measurement Request identifies the route by its RPLInstanceID [RFC6550] (and, in case the RPLInstanceID is a local value, the Start Point's IPv6 address associated with the route). On receiving a Measurement Request, an Intermediate Point updates the routing metric values inside the message and forwards it to the next hop on the route. Thus, the Measurement Request accumulates the values of the routing metrics for the complete route as it travels towards the End Point. Upon receiving the Measurement Request, the End Point unicasts a Measurement Reply message, carrying the accumulated values of the routing metrics, back to the Start Point. Optionally, the Start Point may allow an Intermediate Point to generate the Measurement Reply if the Intermediate Point already knows the relevant routing metric values along rest of the route.

The Measurement Request may include an Address vector that serves one of the following functions:

- o To accumulate a Source Route for End Point's use: If a Hop-by-hop Route with a local RPLInstanceID is being measured, the Start Point may require each Intermediate Point to add its global or unique local IPv6 address to an Address vector inside the Measurement Request. The Source Route, thus accumulated, can be used by the End Point to reach the Start Point. In particular, the End Point may use the accumulated Source Route to send the Measurement Reply back to the Start Point. In this case, the Start Point includes a suitably-sized Address vector in the Measurement Request. The size of the Address vector puts a hard limit on the length of the accumulated route. An Intermediate Point is not allowed to modify the size of the Address vector and must discard a received Measurement Request if the Address vector is not large enough to contain the complete route.
- o To carry the Source Route being measured: The Start Point may insert an Address vector inside the Measurement Request to carry the Source Route being measured. Also, the root of a global non-storing DAG may insert an Address vector, carrying a Source Route from itself to the End Point, inside a Measurement Request message if this message had been traveling along this DAG so far. This Source Route must consist of global or unique local IPv6 addresses. An Intermediate Point is not allowed to modify an

existing Address vector before forwarding the Measurement Request further. In other words, an Intermediate Point must not modify the Source Route along which the Measurement Request is currently traveling.

3. The Measurement Object (MO)

This document defines two new RPL Control Message types, the Measurement Object (MO), with code TBD1, and the Secure MO, with code TBD2. An MO serves as both Measurement Request and Measurement Reply.

3.1. Format of the base MO

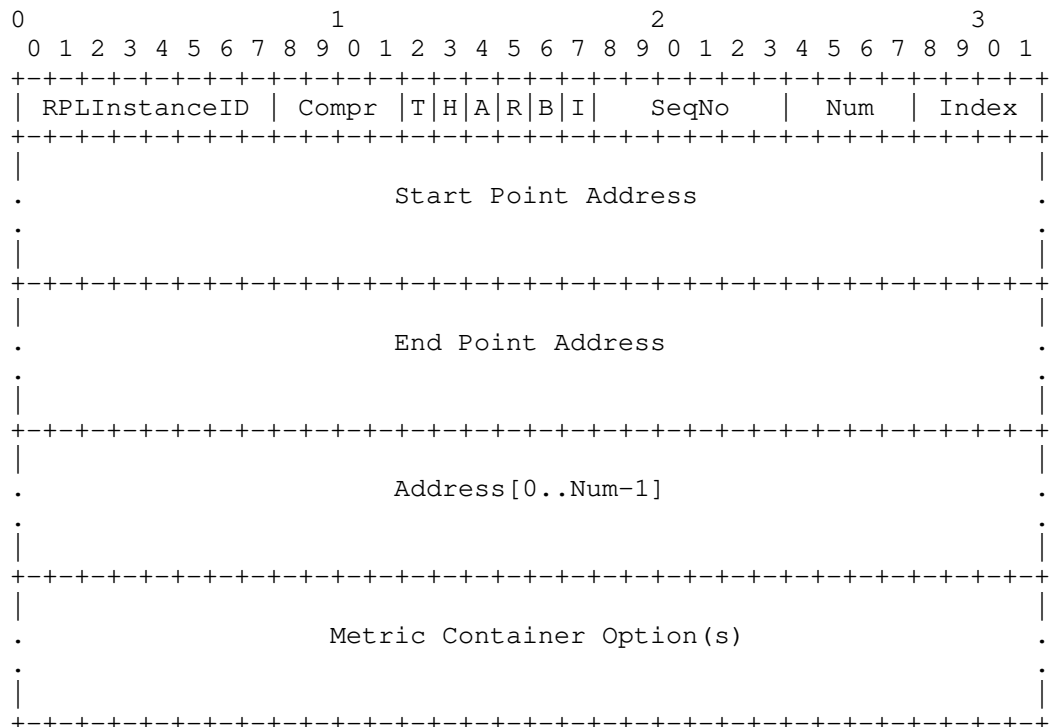


Figure 1: Format of the base Measurement Object (MO)

The format of a base MO is shown in Figure 1. A base MO consists of the following fields:

- o RPLInstanceID: This field specifies the RPLInstanceID of the Hop-by-hop Route along which the Measurement Request travels (or traveled initially until it switched over to a Source Route).
- o Compr: In many LLN deployments, IPv6 addresses share a well known, common prefix. In such cases, the common prefix can be elided when specifying IPv6 addresses in the Start Point/End Point Address fields and the Address vector. The "Compr" field, a 4-bit unsigned integer, is set by the Start Point to specify the number of prefix octets that are elided from the IPv6 addresses in Start Point/End Point Address fields and the Address vector. The Start Point will set the Compr value to zero if full IPv6 addresses are to be carried in the Start Point Address/End Point Address fields and the Address vector.
- o Type (T): This flag is set to one if the MO represents a Measurement Request. The flag is set to zero if the MO is a Measurement Reply.
- o Hop-by-hop (H): The Start Point MUST set this flag to one if (at least the initial part of) the route being measured is hop-by-hop. In that case, the Hop-by-hop Route is identified by the RPLInstanceID, the End Point Address and, if the RPLInstanceID is a local value, the Start Point Address fields inside the Measurement Request. Here, the Start Point Address field is required to be same as the DODAGID (the identifier of the destination-oriented DAG root) [RFC6550] of the route being measured. The Start Point MUST set the H flag to zero if the route being measured is a Source Route specified in the Address vector. An Intermediate Point MUST set the H flag in an outgoing Measurement Request to the same value that it had in the corresponding incoming Measurement Request unless it is the root of the non-storing global DAG, identified by the RPLInstanceID, along which the Measurement Request had been traveling so far and the Intermediate Point intends to insert a Source Route inside the Address vector to direct it towards the End Point. In that case, the Intermediate Point MUST set the H flag to zero.
- o Accumulate Route (A): A value 1 in this flag indicates that the Measurement Request is accumulating a Source Route for use by the End Point to send the Measurement Reply back to the Start Point. Route accumulation MUST NOT be used (i.e., this flag MUST NOT be set to 1) inside a Measurement Request unless it travels along a Hop-by-hop Route represented by a local RPLInstanceID (i.e., H = 1, RPLInstanceID has a local value). Route accumulation MAY be used (i.e., this flag MAY be set to 1) if the Measurement Request is traveling along a Hop-by-hop Route with a local RPLInstanceID. In this case if the route accumulation is on, an Intermediate

Point adds its unicast global/unique-local IPv6 address (after eliding Compr number of prefix octets) to the Address vector in the manner specified in Section 5.3. In other cases, this flag MUST be set to zero on transmission and ignored on reception. Route accumulation is not allowed when the Measurement Request travels along a Hop-by-hop Route with a global RPLInstanceID, i.e., along a global DAG, because:

- * The DAG's root may need the Address vector to insert a Source Route to the End Point; and
 - * The End Point can presumably reach the Start Point along this global DAG (identified by the RPLInstanceID field).
- o Reverse (R): A value 1 in this flag inside a Measurement Request indicates that the Address vector contains a complete Source Route from the Start Point to the End Point, which can be used, after reversal, by the End Point to send the Measurement Reply back to the Start Point. This flag MAY be set to one inside a Measurement Request only if a Source Route, from the Start Point to the End Point, is being measured. Otherwise, this flag MUST be set to zero on transmission and ignored on reception.
 - o Back Request (B): A value 1 in this flag serves as a request to the End Point to send a Measurement Request towards the Start Point. On receiving a Measurement Request with the B flag set to one, the End Point SHOULD generate a Measurement Request to measure the cost of its current (or the most preferred) route to the Start Point. Receipt of this Measurement Request would allow the Start Point to know the cost of the back route from the End Point to itself and thus determine the round-trip cost of reaching the End Point.
 - o Intermediate Reply (I): A value 1 in this flag serves as a permission to an Intermediate Point to generate a Measurement Reply if it knows the aggregated values of the routing metrics being measured for the rest of the route. Setting this flag to one may be useful in scenarios where the Hop Count [RFC6551] is the routing metric of interest and an Intermediate Point (e.g. the root of a non-storing global DAG or a common ancestor of the Start Point and the End Point in a storing global DAG) may know the Hop Count of the remainder of the route to the End Point. This flag MAY be set to one only if a Hop-by-hop Route with a global RPLInstanceID is being measured (i.e., H = 1, RPLInstanceID has a global value). Otherwise, this flag MUST be set to zero on transmission and ignored on reception.

- o SeqNo: A 6-bit sequence number, assigned by the Start Point, that allows the Start Point to uniquely identify a Measurement Request and the corresponding Measurement Reply.
- o Num: This field indicates the number of elements, each (16 - Compr) octets in size, inside the Address vector. If the value of this field is zero, the Address vector is not present in the MO.
- o Index: If the Measurement Request is traveling along a Source Route contained in the Address vector (i.e., H = 0), this field indicates the index in the Address vector of the next hop on the route. If the Measurement Request is traveling along a Hop-by-hop Route with a local RPLInstanceID and the Route Accumulation is on (i.e., H = 1, RPLInstanceID has a local value, A = 1), this field indicates the index in the Address vector where an Intermediate Point receiving the Measurement Request must store its IPv6 address. Otherwise, this field MUST be set to zero on transmission and ignored on reception.
- o Start Point Address: A unicast global or unique local IPv6 address of the Start Point after eliding Compr number of prefix octets. If the Measurement Request is traveling along a Hop-by-hop Route and the RPLInstanceID field indicates a local value, the Start Point Address field MUST specify the DODAGID value that, along with the RPLInstanceID and the End Point Address, uniquely identifies the Hop-by-hop Route being measured.
- o End Point Address: A unicast global or unique local IPv6 address of the End Point after eliding Compr number of prefix octets.
- o Address[0..Num-1]: A vector of unicast global or unique local IPv6 addresses (with Compr number of prefix octets elided) representing a Source Route:
 - * Each element in the vector has size (16 - Compr) octets.
 - * The total number of elements inside the Address vector is given by the Num field.
 - * The Start Point and End Point addresses MUST NOT be included in the Address vector.
 - * The Address vector MUST NOT contain any multicast addresses.
 - * If the Start Point wants to measure a Hop-by-hop Route with a local RPLInstanceID and accumulate a Source Route for the End Point's use (i.e., the Measurement Request has the H flag set to 1, RPLInstanceID set to a local value and the A flag set to

1), it MUST include a suitably-sized Address vector in the Measurement Request. As the Measurement Request travels over the route being measured, the Address vector accumulates a Source Route that can be used by the End Point, after reversal, to reach (and, in particular, to send the Measurement Reply back to) the Start Point. The route MUST be accumulated in the Forward direction but the IPv6 addresses in the accumulated route MUST be reachable in the Reverse direction. An Intermediate Point MUST add only a global or unique local IPv6 address to the Address vector and MUST NOT modify the size of the Address vector.

- * If the Start Point wants to measure a Source Route, it MUST include an Address vector, containing the route being measured, inside the Measurement Request. Similarly, if the Measurement Request had been traveling along a global non-storing DAG so far, the root of this DAG may insert an Address vector, containing a Source Route from itself to the End Point, inside the Measurement Request. In both cases, the Source Route inside the Address vector MUST consist only of global or unique local IPv6 addresses that are reachable in the Forward direction. Further, in both cases, an Intermediate Point MUST NOT modify the contents of the existing Address vector before forwarding the Measurement Request further. In other words, an Intermediate Point MUST NOT modify the Source Route along which the Measurement Request is currently traveling. The Start Point MAY set the R flag in the Measurement Request to one if the Source Route inside the Address vector can be used by the End Point, after reversal, to reach (and, in particular, to send the Measurement Reply back to) the Start Point. In other words, the Start Point MAY set the R flag to one only if all the IPv6 addresses in the Address vector are reachable in the Reverse direction.
- o Metric Container Options: A Measurement Request MUST contain one or more Metric Container options [RFC6550] to accumulate the values of the selected routing metrics in the manner described in [RFC6551] for the route being measured.

Section 4 describes how a Start Point sets various fields inside a Measurement Request in different cases. Section 5 describes how an Intermediate Point processes a received Measurement Request before forwarding it further. Section 6 describes how the End Point processes a received Measurement Request and generate a Measurement Reply. Finally, Section 7 describes how the Start Point processes a received Measurement Reply. In the following discussion, any reference to discarding a received Measurement Request/Reply with "no further processing" does not preclude updating the appropriate error

counters or any similar actions.

3.2. Secure MO

A Secure MO follows the format in Figure 7 of [RFC6550], where the base format is the base MO shown in Figure 1. Sections 6.1, 10 and 19 of [RFC6550] describe RPL security framework. These sections are applicable to the use of Secure MO messages as well except as constrained in this section. An LLN deployment MUST support the use of Secure MO messages so that it has the ability to invoke RPL-provided security mechanisms and prevent misuse of the measurement mechanism by unauthorized routers.

The Start Point determines whether Secure MO messages are to be used in a particular route measurement and if yes the Security Configuration (see definition in [I-D.ietf-roll-p2p-rpl]) to be used for the purpose. The Start Point MUST NOT set the "Key Identifier Mode" field to value 1 inside this Security Configuration since this setting indicates the use of a per-pair key which is not suitable for securing the Measurement Request messages that travel over multiple hops. A router (an Intermediate Point or the End Point), participating in a particular route measurement,

- o MUST generate a Secure MO message (a Measurement Request or a Measurement Reply) if the received Measurement Request is a Secure MO. The Security Configuration used in generating a Secure MO message MUST be same as the one used in the received message.
- o MUST NOT generate a Secure MO message if the received Measurement Request is not a Secure MO.

A router MUST discard a received Measurement Request if it cannot follow the above mentioned rules. If the Start Point sends a Measurement Request in a Secure MO message using a particular Security Configuration, it MUST discard the corresponding Measurement Reply it receives with no further processing unless the Measurement Reply is received in a Secure MO message generated with same Security Configuration as the one used in the Measurement Request.

In the following discussion, any reference to an MO message is also applicable to a Secure MO message unless noted otherwise.

4. Originating a Measurement Request

A Start Point sets various fields inside the Measurement Request it generates in the manner described below. The Start Point MUST also include the routing metric objects [RFC6551] of interest inside one

or more Metric Container options inside the Measurement Request. The Start Point then determines the next hop on the route being measured. If a Hop-by-hop route is being measured (i.e., $H = 1$), the next hop is determined using the RPLInstanceID, the End Point Address and, if RPLInstanceID is a local value, the Start Point Address fields in the Measurement Request. If a Source Route is being measured (i.e., $H = 0$), the Address[0] element inside the Measurement Request contains the next hop address. The Start Point MUST ensure that

- o the next hop address is a unicast address; and
- o the next hop is on-link; and
- o the next hop is in the same RPL routing domain [I-D.ietf-roll-terminology] as the Start Point;

failing which the Start Point MUST discard the Measurement Request without sending. Depending on the routing metrics, the Start Point must initiate the routing metric objects inside the Metric Container options by including the routing metric values for the first hop on the route being measured. Finally, the Start Point MUST unicast the Measurement Request to the next hop on the route being measured.

The Start Point MUST maintain state for just transmitted Measurement Request for a life time duration that is large enough to allow the corresponding Measurement Reply to return. This state consists of the RPLInstanceID, the SeqNo and the End Point Address fields of the Measurement Request. The life time duration for this state is locally determined by the Start Point and may be deployment specific. This state expires when the corresponding Measurement Reply is received or when the life time is over, whichever occurs first. Failure to receive the corresponding Measurement Reply before the expiry of a state may occur due to a number of reasons including unwillingness on part of an Intermediate Point or the End Point to process the Measurement Request. The Start Point should take such possibilities in account when deciding whether to generate another Measurement Request for this route. The Start Point MUST discard a received Measurement Reply with no further processing if the state for the corresponding Measurement Request has already expired.

4.1. When Measuring A Hop-by-hop Route with a Global RPLInstanceID

If a Hop-by-hop Route with a global RPLInstanceID is being measured (i.e., $H = 1$, RPLInstanceID has a global value), the MO MUST NOT contain an Address vector and various MO fields MUST be set in the following manner:

- o RPLInstanceID: MUST be set to the RPLInstanceID of the route being measured.
 - o Compr: MUST be set to specify the number of prefix octets that are elided from the IPv6 addresses in Start Point/End Point Address fields.
 - o Type (T): MUST be set to one since the MO represents a Measurement Request.
 - o Hop-by-hop (H): MUST be set to one.
 - o Accumulate Route (A): This flag MUST be set to zero.
 - o Reverse (R): This flag MUST be set to zero.
 - o Back Request (B): This flag MAY be set to one to request the End Point to send a Measurement Request to the Start Point.
 - o Intermediate Reply (I): This flag MAY be set to one if the Start Point expects an Intermediate Point to know the values of the routing metrics being measured for the remainder of the route.
 - o SeqNo: Assigned by the Start Point so that it can uniquely identify the Measurement Request and the corresponding Measurement Reply.
 - o Num: This field MUST be set to zero.
 - o Index: This field MUST be set to zero.
 - o Start Point Address: MUST be set to a unicast global/unique-local IPv6 address of the Start Point after eliding Compr number of prefix octets.
 - o End Point Address: MUST be set to a unicast global/unique-local IPv6 address of the End Point after eliding Compr number of prefix octets.
- 4.2. When Measuring A Hop-by-hop Route with a Local RPLInstanceID With Route Accumulation Off

If a Hop-by-hop Route with a local RPLInstanceID is being measured and the Start Point does not want the MO to accumulate a Source Route for the End Point's use, the MO MUST NOT contain the Address vector and various MO fields MUST be set in the following manner:

- o RPLInstanceID: MUST be set to the RPLInstanceID of the route being measured.
- o Compr: MUST be set to specify the number of prefix octets that are elided from the IPv6 addresses in Start Point/End Point Address fields.
- o Type (T): MUST be set to one since the MO represents a Measurement Request.
- o Hop-by-hop (H): MUST be set to one.
- o Accumulate Route (A): This flag MUST be set to zero.
- o Reverse (R): This flag MUST be set to zero.
- o Back Request (B): This flag MAY be set to one to request the End Point to send a Measurement Request to the Start Point.
- o Intermediate Reply (I): This flag MUST be set to zero.
- o SeqNo: Assigned by the Start Point so that it can uniquely identify the Measurement Request and the corresponding Measurement Reply.
- o Num: This field MUST be set to zero.
- o Index: This field MUST be set to zero.
- o Start Point Address: This field MUST contain the DODAGID value (after eliding Compr number of prefix octets) associated with the route being measured. This DODAGID MUST also be a global or unique local IPv6 address of the Start Point.
- o End Point Address: MUST be set to a unicast global or unique local IPv6 address of the End Point after eliding Compr number of prefix octets.

4.3. When Measuring A Hop-by-hop Route with a Local RPLInstanceID With Route Accumulation On

If a Hop-by-hop Route with a local RPLInstanceID is being measured and the Start Point desires the MO to accumulate a Source Route for the End Point to send the Measurement Reply message back, the MO MUST contain a suitably-sized Address vector and various MO fields MUST be set in the following manner:

- o RPLInstanceID: MUST be set to the RPLInstanceID of the route being measured.
- o Compr: MUST be set to specify the number of prefix octets that are elided from the IPv6 addresses in Start Point/End Point Address fields and the Address vector.
- o Type (T): MUST be set to one since the MO represents a Measurement Request.
- o Hop-by-hop (H): MUST be set to one.
- o Accumulate Route (A): This flag MUST be set to one.
- o Reverse (R): This flag MUST be set to zero.
- o Back Request (B): This flag MAY be set to one to request the End Point to send a Measurement Request to the Start Point.
- o Intermediate Reply (I): This flag MUST be set to zero.
- o SeqNo: Assigned by the Start Point so that it can uniquely identify the Measurement Request and the corresponding Measurement Reply.
- o Num: This field MUST specify the number of address elements, each (16 - Compr) octets in size, that can fit inside the Address vector.
- o Index: This field MUST be set to zero to indicate the position in the Address vector where the next hop must store its IPv6 address.
- o Start Point Address: This field MUST contain the DODAGID value (after eliding Compr number of prefix octets) associated with the route being measured. This DODAGID MUST also be a global or unique local IPv6 address of the Start Point.
- o End Point Address: MUST be set to a unicast global or unique local IPv6 address of the End Point after eliding Compr number of prefix octets.
- o Address vector: The Address vector must be large enough to accommodate a complete Source Route from the End Point to the Start Point. All the bits in the Address vector field MUST be set to zero.

4.4. When Measuring A Source Route

If a Source Route is being measured, the Start Point MUST set various MO fields in the following manner:

- o RPLInstanceID: This field does not have any significance when a Source Route is being measured and hence can be set to any value.
- o Compr: MUST be set to specify the number of prefix octets that are elided from the IPv6 addresses in Start Point/End Point Address fields and the Address vector.
- o Type (T): MUST be set to one since the MO represents a Measurement Request.
- o Hop-by-hop (H): MUST be set to zero.
- o Accumulate Route (A): This flag MUST be set to zero.
- o Reverse (R): This flag SHOULD be set to one if the Source Route in the Address vector can be reversed and used by the End Point to send the Measurement Reply message back to the Start Point. Otherwise, this flag MUST be set to zero.
- o Back Request (B): This flag MAY be set to one to request the End Point to send a Measurement Request to the Start Point.
- o Intermediate Reply (I): This flag MUST be set to zero.
- o SeqNo: Assigned by the Start Point so that it can uniquely identify the Measurement Request and the corresponding Measurement Reply.
- o Num: This field MUST specify the number of address elements, each (16 - Compr) octets in size, inside the Address vector.
- o Index: This field MUST be set to zero to indicate the position in the Address vector of the next hop on the route.
- o Start Point Address: MUST be set to a unicast global or unique local IPv6 address of the Start Point after eliding Compr number of prefix octets.
- o End Point Address: MUST be set to a unicast global or unique local IPv6 address of the End Point after eliding Compr number of prefix octets.

- o Address vector:

- * The Address vector MUST contain a complete Source Route from the Start Point to the End Point (excluding the Start Point and the End Point).
- * Each address appearing in the Address vector MUST be a unicast global or unique local IPv6 address. Further, each address MUST have the same prefix as the Start Point Address and the End Point Address. This prefix, whose length in octets is specified in the Compr field, MUST be elided from each address.
- * The IPv6 addresses in the Address vector MUST be reachable in the Forward direction.
- * If the R flag is set to one, the IPv6 addresses in the Address vector MUST also be reachable in the Reverse direction.

5. Processing a Measurement Request at an Intermediate Point

A router (an Intermediate Point or the End Point) MAY discard a received MO with no processing to meet any policy-related goal. Such policy goals may include the need to reduce the router's CPU load or to enhance its battery life or to prevent misuse of this mechanism by unauthorized nodes.

A router MUST discard a received MO with no further processing if the value in the Compr field inside the received message is more than what the router considers the length of the common prefix used in IPv6 addresses in the LLN to be.

On receiving an MO, if a router chooses to process the packet further, it MUST check if one of its IPv6 addresses is listed as either the Start Point or the End Point Address. If neither, the router considers itself an Intermediate Point and MUST process the received MO in the following manner.

An Intermediate Point MUST discard the packet with no further processing if the received MO is not a Measurement Request (i.e., T = 0). This is because the End Point unicasts a Measurement Reply directly to the Start Point. So, the Intermediate Point treats a transiting Measurement Reply as a data packet and not an RPL control message.

Next, the Intermediate Point determines the type of the route being measured (by checking the values of the H flag and the RPLInstanceID field) and processes the received MO accordingly in the manner

specified next.

5.1. When Measuring A Hop-by-hop Route with a Global RPLInstanceID

If a Hop-by-hop Route with a global RPLInstanceID is being measured (i.e. $H = 1$ and RPLInstanceID has a global value), the Intermediate Point MUST process the received Measurement Request in the following manner.

If the Num field inside the received Measurement Request is not set to zero, thereby implying that an Address vector is present, the Intermediate Point MUST discard the received message with no further processing.

If the Intermediate Reply (I) flag is set to one in the received Measurement Request and the Intermediate Point knows the values of the routing metrics, specified in the Metric Container options, for the remainder of the route, it MAY generate a Measurement Reply on the End Point's behalf in the manner specified in Section 6.1 (after including in the Measurement Reply the relevant routing metric values for the complete route being measured). Otherwise, the Intermediate Point MUST process the received message in the following manner.

The Intermediate Point MUST determine the next hop on the route being measured using the RPLInstanceID and the End Point Address. If the Intermediate Point is the root of the non-storing global DAG along which the received Measurement Request had been traveling so far, it MUST process the received Measurement Request in the following manner:

- o If the router does not know how to reach the End Point, it MUST discard the Measurement Request with no further processing and MAY send an ICMPv6 Destination Unreachable (with Code 0 - No Route To Destination) error message [RFC4443] to the Start Point.
- o Otherwise, unless the router determines the End Point itself to be the next hop, the router MUST make the following changes in the received Measurement Request:
 - * Set the H, A, R and I flags to zero (the A and R flags should already be zero in the received message).
 - * Leave remaining fields unchanged (the Num field would be modified in next steps). Note that the RPLInstanceID field identifies the non-storing global DAG along which the Measurement Request traveled so far. This information MUST be preserved so that the End Point may use this DAG to send the Measurement Reply back to the Start Point.

- * Insert a new Address vector inside the Measurement Request and specify a Source Route to the End Point inside the Address vector as per the following rules:
 - + The Address vector MUST contain a complete route from the router to the End Point (excluding the router and the End Point);
 - + Each address appearing in the Address vector MUST be a unicast global or unique local IPv6 address. Further, each address MUST have the same prefix as the Start Point Address and the End Point Address. This prefix, whose length in octets is specified in the Compr field, MUST be elided from each address.
 - + The IPv6 addresses in the Address vector MUST be reachable in the Forward direction;

If the router cannot insert an Address vector satisfying the rules mentioned above, it MUST discard the Measurement Request with no further processing and MAY send an ICMPv6 Destination Unreachable (with Code 0 - No Route To Destination) error message [RFC4443] to the Start Point.

- * Specify in the Num field the number of address elements in the Address vector.
- * Set the Index field to zero to indicate the position in the Address vector of the next hop on the route. Thus, Address[0] element contains the address of the next hop on the route.

The Intermediate Point MUST then complete the processing of the received Measurement Request as specified in Section 5.5.

5.2. When Measuring A Hop-by-hop Route with a Local RPLInstanceID With Route Accumulation Off

If a Hop-by-hop Route with a local RPLInstanceID is being measured and the route accumulation is off (i.e., $H = 1$, RPLInstanceID has a local value, $A = 0$), the Intermediate Point MUST process the received Measurement Request in the following manner.

If the Num field inside the received Measurement Request is not set to zero, thereby implying that an Address vector is present, the Intermediate Point MUST discard the received message with no further processing.

The Intermediate Point MUST then determine the next hop on the route

being measured using the RPLInstanceID, the End Point Address and the Start Point Address (which represents the DODAGID of the route being measured). If the Intermediate Point can not determine the next hop, it MUST discard the Measurement Request with no further processing and MAY send an ICMPv6 Destination Unreachable (with Code 0 - No Route To Destination) error message [RFC4443] to the Start Point. Otherwise, the Intermediate Point MUST complete the processing of the received Measurement Request as specified in Section 5.5.

5.3. When Measuring A Hop-by-hop Route with a Local RPLInstanceID With Route Accumulation On

If a Hop-by-hop Route with a local RPLInstanceID is being measured and the route accumulation is on (i.e., $H = 1$, RPLInstanceID has a local value, $A = 1$), the Intermediate Point MUST process the received Measurement Request in the following manner.

If the Num field inside the received Measurement Request is set to zero, thereby implying that an Address vector is not present, the Intermediate Point MUST discard the received message with no further processing.

The Intermediate Point MUST then determine the next hop on the route being measured using the RPLInstanceID, the End Point Address and the Start Point Address (which represents the DODAGID of the route being measured). If the Intermediate Point can not determine the next hop, it MUST discard the Measurement Request with no further processing and MAY send an ICMPv6 Destination Unreachable (with Code 0 - No Route To Destination) error message [RFC4443] to the Start Point. If the index field has value Num - 1 and the next hop is not same as the End Point, the Intermediate Point MUST drop the received Measurement Request with no further processing. In this case, the next hop would have no space left in the Address vector to store its address. Otherwise, the router MUST store one of its IPv6 addresses at location Address[Index] and then increment the Index field. The IPv6 address added to the Address vector MUST have the following properties:

- o This address MUST be a unicast global or unique local address.
- o This address MUST have the same prefix as the Start Point Address and the End Point Address. This prefix, whose length in octets is specified in the Compr field, MUST be elided before the address is added to the Address vector.
- o This address MUST be reachable in the Reverse direction.

If the router does not have an IPv6 address that satisfies the

properties mentioned above, it MUST discard the Measurement Request with no further processing.

The Intermediate Point MUST then complete the processing of the received Measurement Request as specified in Section 5.5.

5.4. When Measuring A Source Route

If a Source Route is being measured (i.e., $H = 0$), the Intermediate Point MUST process the received Measurement Request in the following manner.

If the Num field inside the received Measurement Request is set to zero, thereby implying that an Address vector is not present, the Intermediate Point MUST discard the received message with no further processing.

The Intermediate Point MUST verify that the Address[Index] element lists one of its unicast global or unique local IPv6 addresses (minus the prefix whose length in octets is specified in the Compr field), failing which it MUST discard the Measurement Request with no further processing. The Intermediate Point MUST then increment the Index field and use the Address[Index] element as the next hop (unless Index value is now Num). If the Index value is now Num, the Intermediate Point MUST use the End Point Address as the next hop.

The Intermediate Point MUST then complete the processing of the received Measurement Request as specified in Section 5.5.

5.5. Final Processing

The Intermediate Point MUST drop the received Measurement Request with no further processing:

- o If the next hop address is not a unicast address; or
- o If the next hop is not on-link; or
- o If the next hop is not in the same RPL routing domain as the Intermediate Point.

Next, the Intermediate Point MUST update the routing metric objects, inside the Metric Container option(s) inside the Measurement Request, either by updating the aggregated value for the routing metric or by attaching the local values for the metric inside the object. An Intermediate Point can only update the existing metric objects and MUST NOT add any new routing metric object to the Metric Container. An Intermediate Point MUST drop the Measurement Request with no

further processing if it cannot update a routing metric object specified inside the Metric Container.

Finally, the Intermediate Point MUST unicast the Measurement Request to the next hop.

6. Processing a Measurement Request at the End Point

On receiving an MO, if a router chooses to process the message further and finds one of its unicast global or unique local IPv6 addresses (minus the prefix whose length in octets is specified in the Compr field) listed as the End Point Address, the router considers itself the End Point and MUST process the received MO in the following manner.

The End Point MUST discard the received message with no further processing if it is not a Measurement Request (i.e., T = 0).

If the received Measurement Request traveled on a Hop-by-hop Route with a local RPLInstanceID with route accumulation on (i.e., H = 1, RPLInstanceID has a local value and A = 1), elements Address[0] through Address[Index - 1] in the Address vector contain a complete Source Route from the Start Point to the End Point, which the End Point MAY use, after reversal, to reach the Start Point. Note that the Source Route in the Address vector does not include the Start Point and the End Point addresses and the individual addresses do not include the common prefix whose length in octets is specified in the Compr field.

If the received Measurement Request traveled on a Source Route and the Reverse flag is set to one (i.e., H = 0, R = 1), elements Address[0] through Address[Num - 1] in the Address vector contain a complete Source Route from the Start Point to the End Point, which the End Point MAY use, after reversal, to reach the Start Point. Again, the Source Route in the Address vector does not include the Start Point and the End Point addresses and the individual addresses do not include the common prefix whose length in octets is specified in the Compr field.

The End Point MUST update the routing metric objects in the Metric Container options if required and MAY note the measured values for the complete route (especially, if the received Measurement Request is likely a response to an earlier Measurement Request that the End Point had sent to the Start Point with B flag set to one).

The End Point MUST generate a Measurement Reply message as specified in Section 6.1. If the B flag is set to one in the received

Measurement Request, the End Point SHOULD generate a new Measurement Request to measure the cost of its current (or the most preferred) route to the Start Point. The routing metrics used in the new Measurement Request MUST include the routing metrics specified in the received Measurement Request.

6.1. Generating the Measurement Reply

A Measurement Reply MUST have the Type (T) flag set to zero and need not contain the Address vector. The following fields inside a Measurement Reply MUST have the same values as they had inside the corresponding Measurement Request: RPLInstanceID, Compr, SeqNo, Start Point Address, End Point Address and Metric Container Option(s). The remaining fields inside a Measurement Reply may have any value and MUST be ignored on reception at the Start Point; the received Measurement Request can, therefore, trivially be converted into a Measurement Reply by setting the Type (T) flag to zero.

A Measurement Reply MUST be unicast back to the Start Point:

- o If the Measurement Request traveled along a global DAG, identified by the RPLInstanceID field, the Measurement Reply MAY be unicast back to the Start Point along the same DAG.
- o If the Measurement Request traveled along a Hop-by-hop Route with a local RPLInstanceID and accumulated a Source Route from the Start Point to the End Point, this Source Route MAY be used after reversal to send the Measurement Reply back to the Start Point.
- o If the Measurement Request traveled along a Source Route and the R flag inside the received message is set to one, the End Point MAY reverse the Source Route contained in the Address vector and use it to send the Measurement Reply back to the Start Point.

7. Processing a Measurement Reply at the Start Point

When a router receives an MO, it examines if one of its unicast IPv6 addresses is listed as the Start Point Address. If yes, the router is the Start Point and MUST process the received message in the following manner.

If the Start Point discovers that the received MO is not a Measurement Reply or if it no longer maintains state for the corresponding Measurement Request, it MUST discard the received message with no further processing.

The Start Point can use the routing metric objects inside the Metric

Container to evaluate the metrics for the measured P2P route. If a routing metric object contains local metric values recorded by routers on the route, the Start Point can make use of these local values by aggregating them into an end-to-end metric according to the aggregation rules for the specific metric. A Start Point is then free to interpret the metrics for the route according to its local policy.

8. Security Considerations

In general, the security considerations for the route measurement mechanism described in this document are similar to the ones for RPL (as described in Section 19 of [RFC6550]). Sections 6.1 and 10 of RPL specification [RFC6550] describe RPL's security framework that provides data confidentiality, authentication, replay protection and delay protection services. This security framework is applicable to the route measurement mechanism described here as well after taking in account the constraints specified in Section 3.2.

This document requires all routers participating in a secure invocation of the route measurement process to use the Security Configuration decided by the Start Point. The intention is to avoid compromising the overall security of the route measurement due to some routers using a weaker Security Configuration. A router is allowed to participate in a "secure" route measurement only if it can support the Security Configuration in use, which also specifies the key in use. It does not matter whether the key is pre-installed or dynamically acquired after proper authentication. The router must have the key in use before it can process or generate Secure MO messages. Hence, from the perspective of the route measurement mechanism, there is no distinction between the "preinstalled" and "authenticated" security modes described in RPL specification [RFC6550]. Ofcourse if a compromised router has the key being used, it could cause the route measurement to fail, or worse, insert wrong information in Secure MO messages.

A rogue router acting as the Start Point could use the route measurement mechanism defined in this document to measure routes from itself to other routers and thus find out key information about the LLN, e.g., the topological features of the LLN (such as the identity of the key routers in the topology) or the remaining energy levels [RFC6551] in the routers. This information can potentially be used to attack the LLN. A rogue router could also use this mechanism to send bogus Measurement Requests to arbitrary End Points. If sufficient Measurement Requests are sent, then it may cause CPU overload in the routers in the network, drain their batteries and cause traffic congestion in the network. Note that some of these

problems would occur even if the compromised router were to generate bogus data traffic to arbitrary destinations.

To protect against such misuse, this document allows RPL routers implementing this mechanism to not process MO messages (or process such messages selectively) based on a local policy. For example, an LLN deployment might require the use of Secure MO messages generated using a key that could be obtained only after proper authentication. Note that this document requires an LLN deployment to support Secure MO messages so that such policies can be enforced where considered essential.

Since a Measurement Request can travel along a Source Route specified in the Address vector, some of the security concerns that led to the deprecation of Type 0 routing header [RFC5095] may be valid here. To address such concerns, the mechanism described in this document includes several remedies:

- o This document requires that a route inserted inside the Address vector must be a strict Source Route and must not include any multicast addresses.
- o This document requires that an MO message must not cross the boundaries of the RPL routing domain where it originated. A router must not forward a received MO message further if the next hop belongs to a different RPL routing domain. Hence, any security problems associated with the mechanism would be limited to one RPL routing domain.
- o This document requires that a router must drop a received Measurement Request if the next hop address is not on-link or if it is not a unicast address.

9. IANA Considerations

This document defines two new RPL messages:

- o "Measurement Object" (see Section 3.1), assigned a value TBD1 from the "RPL Control Codes" space [to be removed upon publication: <http://www.iana.org/assignments/rpl/rpl.xml#control-codes>] [RFC6550]. IANA is requested to allocate TBD1 from the range 0x00-0x7F to indicate a message without security enabled. The string TBD1 in this document should be replaced by the allocated value. These last two sentences should be removed before publication.

- o "Secure Measurement Object" (see Section 3.2), assigned a value TBD2 from the "RPL Control Codes" space [to be removed upon publication:
<http://www.iana.org/assignments/rpl/rpl.xml#control-codes>]
[RFC6550]. IANA is requested to allocate TBD2 from the range 0x80-0xFF to indicate a message with security enabled. The string TBD2 in this document should be replaced by the allocated value. These last two sentences should be removed before publication.

Code	Description	Reference
TBD1	Measurement Object	This document
TBD2	Secure Measurement Object	This document

RPL Control Codes

10. Acknowledgements

Authors gratefully acknowledge the contributions of Ralph Droms, Adrian Farrel, Joel Halpern, Matthias Philipp, Pascal Thubert, Richard Kelsey and Zach Shelby in the development of this document.

11. References

11.1. Normative References

- [I-D.ietf-roll-p2p-rpl]
Goyal, M., Baccelli, E., Philipp, M., Brandt, A., and J. Martocci, "Reactive Discovery of Point-to-Point Routes in Low Power and Lossy Networks", draft-ietf-roll-p2p-rpl-17 (work in progress), March 2013.
- [I-D.ietf-roll-terminology]
Vasseur, J., "Terminology in Low power And Lossy Networks", draft-ietf-roll-terminology-12 (work in progress), March 2013.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [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.

- [RFC6550] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, March 2012.

11.2. Informative References

- [RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, December 2007.
- [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.
- [RFC6551] Vasseur, JP., Kim, M., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks", RFC 6551, March 2012.

Authors' Addresses

Mukul Goyal (editor)
University of Wisconsin Milwaukee
3200 N Cramer St
Milwaukee, WI 53211
USA

Phone: +1 414 2295001
Email: mukul@uwm.edu

Emmanuel Baccelli
INRIA

Phone: +33-169-335-511
Email: Emmanuel.Baccelli@inria.fr
URI: <http://www.emmanuelbaccelli.org/>

Anders Brandt
Sigma Designs
Emdrupvej 26A, 1.
Copenhagen, Dk-2100
Denmark

Phone: +45 29609501
Email: abr@sdesigns.dk

Jerald Martocci
Johnson Controls
507 E Michigan Street
Milwaukee 53202
USA

Phone: +1 414 524 4010
Email: jerald.p.martocci@jci.com

Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: September 21, 2013

M. Goyal, Ed.
University of Wisconsin
Milwaukee
E. Baccelli
M. Philipp
INRIA
A. Brandt
Sigma Designs
J. Martocci
Johnson Controls
March 20, 2013

Reactive Discovery of Point-to-Point Routes in Low Power and Lossy
Networks
draft-ietf-roll-p2p-rpl-17

Abstract

This document specifies a point-to-point route discovery mechanism, complementary to the RPL core functionality. This mechanism allows an IPv6 router to discover "on demand" routes to one or more IPv6 routers in the LLN such that the discovered routes meet specified metrics constraints.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 21, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal

Provisions Relating to IETF Documents
(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	4
1.1. Known Issues/Future Work	4
2. The Use Cases	5
3. Terminology	5
4. Applicability	6
5. Functional Overview	8
6. P2P Route Discovery Mode Of Operation	10
6.1. Setting a P2P Mode DIO	11
7. P2P Route Discovery Option (P2P-RDO)	14
8. The P2P Discovery Reply Object (P2P-DRO)	18
8.1. Secure P2P-DRO	20
8.2. Setting a P2P-RDO Carried in a P2P Discovery Reply Object	20
9. P2P-RPL Route Discovery By Creating a Temporary DAG	21
9.1. Joining a Temporary DAG	21
9.2. Trickle Operation For P2P Mode DIOs	22
9.3. Processing a P2P Mode DIO	24
9.4. Additional Processing of a P2P Mode DIO At An Intermediate Router	25
9.5. Additional Processing of a P2P Mode DIO At The Target	26
9.6. Processing a P2P-DRO At An Intermediate Router	27
9.7. Processing a P2P-DRO At The Origin	29
10. The P2P Discovery Reply Object Acknowledgement (P2P-DRO-ACK)	30
11. Secure P2P-RPL Operation	31
12. Packet Forwarding Along a Route Discovered Using P2P-RPL	32
13. Interoperability with Core RPL	33
14. Security Considerations	33
15. IANA Considerations	35
15.1. Additions to Mode of Operation	35
15.2. Additions to RPL Control Message Options	35
15.3. Additions to RPL Control Codes	36
16. Acknowledgements	37
17. References	37
17.1. Normative References	37
17.2. Informative References	38

Authors' Addresses

38

1. Introduction

Targeting Low power and Lossy Networks (LLNs), the IPv6 Routing Protocol for LLNs (RPL) [RFC6550] provides paths along a Directed Acyclic Graph (DAG) rooted at a single router in the network. Establishment and maintenance of a DAG is performed by routers in the LLN using Destination-Oriented DAG (DODAG) Information Object (DIO) messages. When two arbitrary routers (neither of which is the DAG's root) need to communicate, the data packets are restricted to travel only along the links in the DAG. Such point-to-point (P2P) routing functionality may not be sufficient for several Home and Building Automation applications [RFC5826] [RFC5867] due to the following reasons:

- o The need to pre-establish routes: each potential destination in the network must declare itself as such ahead of the time a source needs to reach it.
- o The need to route only along the links in the DAG: A DAG is built to optimize the routing cost to reach the root. Restricting P2P routes to use only the in-DAG links may result in significantly suboptimal routes and severe traffic congestion near the DAG root.

This document describes an extension to core RPL (i.e., the RPL functionality described in [RFC6550]) that enables an IPv6 router in the LLN to discover routes to one or more IPv6 routers in the LLN "on demand". The discovered routes may not be the best available but are guaranteed to meet the specified routing metric constraints. Thus, such routes are considered "good enough" from the application's perspective. This reactive P2P route discovery mechanism is henceforth referred to as P2P-RPL.

A mechanism to measure the end-to-end cost of an existing route is specified in [I-D.ietf-roll-p2p-measurement]. As discussed in Section 4, measuring the end-to-end cost of an existing route may help decide whether to initiate the discovery of a better route using P2P-RPL and the metric constraints to be used for this purpose.

1.1. Known Issues/Future Work

This document is presented as an Experimental specification to facilitate P2P-RPL's deployment in LLN scenarios where reactive P2P route discovery is considered useful or necessary. It is anticipated that, once sufficient operational experience has been gained, this specification will be revised to progress it on to the Standards Track. Experience reports regarding P2P-RPL implementation and deployment are encouraged particularly with respect to:

- o Secure P2P-RPL operation (Section 11);
- o Rules governing Trickle operation (Section 9.2);
- o Values in the default DODAG Configuration Option (Section 6.1);
- o The RPLInstanceID reuse policy (Section 6.1);
- o Utility and implementation complexity of allowing multiple Target addresses in a P2P-RPL route discovery.

2. The Use Cases

One use case, common in home [RFC5826] and commercial building [RFC5867] environments, involves a device (say a remote control) that suddenly needs to communicate with another device (say a lamp) to which it does not already have a route (and whose network address it knows apriori). In this case, the remote control must be able to discover a route to the lamp "on demand".

Another use case, common in a commercial building environment, involves a large LLN deployment where P2P communication along a particular DAG among hundreds (or thousands) of routers creates severe traffic congestion near that DAG's root. In this case, it is desirable to discover direct routes between various source-destination pairs that do not pass through the DAG's root.

Other use cases involve scenarios where energy or latency constraints are not satisfied by the P2P routes along an existing DAG because they involve traversing many more intermediate routers than necessary to reach the destination.

3. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Additionally, this document uses terminology from [RFC6550] and [I-D.ietf-roll-terminology]. This document introduces the following terms:

Origin : The IPv6 router initiating the P2P-RPL route discovery.

Target : The IPv6 router at the other end point of the P2P route(s)

to be discovered. A P2P-RPL route discovery can discover routes to multiple Targets at the same time.

Intermediate Router: An IPv6 router that is neither the Origin nor a Target.

Forward direction: The direction from the Origin to the Target.

Reverse direction: The direction from the Target to the Origin.

Forward Route: A route in the Forward direction.

Reverse Route: A route in the Reverse direction.

Bidirectional Route: A route that can be used in both Forward and Reverse directions.

Ingress-only Interface: A network interface that can only receive packets.

Egress-only Interface: A network interface that can only send packets.

Source Route: A complete and ordered list of routers that can be used by a packet to travel from a source to a destination node.

Hop-by-hop Route: The route characterized by each router on the route using its routing table to determine the next hop on the route.

RPL Security Configuration: The values for Counter is Time, Security Algorithm, Key Identifier Mode and Security Level fields, defined in Section 6.1 of [RFC6550], inside the Security section of a secure RPL control message.

4. Applicability

A route discovery using P2P-RPL may be performed by an Origin when no route exists between itself and the Target(s) or when the existing routes do not satisfy the application requirements. P2P-RPL is designed to discover Hop-by-hop or Source Routes to one or more Targets such that the discovered routes meet the specified constraints. In some application contexts, the constraints that the discovered routes must satisfy are intrinsically known or can be specified by the application. For example, an Origin that expects its Targets to be less than 5 hops away may use "hop-count < 5" as the constraint. In other application contexts, the Origin may need to measure the cost of the existing route to a Target to determine

the constraints. For example, an Origin that measures the total ETX along its current route to a Target to be 20 may use "ETX < x*20", where x is a fraction that the Origin decides, as the constraint. A mechanism to measure the cost of an existing route between two IPv6 routers is specified in [I-D.ietf-roll-p2p-measurement]. If there is no existing route between the Origin and the Target(s) or the cost measurement for the existing routes fails, the Origin will have to guess the constraints to be used in the initial route discovery. Once, the initial route discovery succeeds or fails, the Origin will have a better estimate for the constraints to be used in the subsequent route discovery.

P2P-RPL may result in discovery of better P2P routes than the ones available along a global DAG designed to optimize routing cost to the DAG's root. The improvement in route quality depends on a number of factors including the network topology, the "distance" between the Origin and the Target (in terms of the routing metrics in use) and the prevalent conditions in the network. In general, a P2P-RPL route may be better than the one along a global DAG if the Origin and the Target are nearby. Similarly, a P2P-RPL route may not be much better than the one along a global DAG if the Origin and the Target are far apart. Note that, even when P2P-RPL routes are not much better than those along a global DAG, P2P-RPL routes may still be able to avoid congestion that might occur near the root if the routing takes place only along a global DAG. In general, the costs associated with a P2P-RPL route discovery (in terms of the control messages, mostly DIOs, generated) increases with the distance between the Origin and the Target. However, it is possible to limit the cost of route discovery by carefully setting the routing constraints, the Trickle parameters (that govern the DIO generation) and the time duration for which a router maintains its membership in the temporary DAG created for the route discovery. A network designer may take into consideration both the benefits (potentially better routes; no need to maintain routes proactively; avoid congestion near the global DAG's root) and costs when using P2P-RPL. The latency associated with a P2P-RPL route discovery again depends on the distance between the Origin and the Target and the Trickle parameters.

Like core RPL [RFC6550], P2P-RPL operation requires links to have bidirectional reachability. For this reason, the routers participating in a P2P-RPL route discovery must ensure that

- o Links that do not have bidirectional reachability do not become part of the route being discovered; and
- o IPv6 addresses belonging to Ingress-only (or Egress-only) Interfaces do not become part of the route being discovered.

5. Functional Overview

This section contains a high level description of P2P-RPL.

A P2P-RPL route discovery takes place by forming a DAG rooted at the Origin. As is the case with core RPL, P2P-RPL uses IPv6 link-local multicast DIO messages to establish a DAG. However, unlike core RPL, this DAG is temporary in nature. The routes are discovered and installed while the DAG is alive. Once the specified duration of their membership in the DAG is over, the routers leave the DAG and hence the DAG ceases to exist. However, the installed routes are retained for their specified life time (which is different than the specified duration of a router's membership in the DAG) even though the DAG that caused their installation no longer exists. In P2P-RPL, the sole purpose of DAG creation is to discover routes to the Target(s) and DIOs serve as the route discovery messages. Each router joining the DAG determines a rank for itself in the DAG and ignores the subsequent DIOs received from lower (higher in numerical value) ranked neighbors. Thus, the route discovery messages propagate away from the Origin rather than return back to it. As in core RPL, DIO generation at a router is controlled by a Trickle timer [RFC6206] that allows a router to avoid generating unnecessary messages while providing protection against packet loss. P2P-RPL also uses the routing metrics [RFC6551], objective functions and packet forwarding framework [RFC6554][RFC6553] developed for core RPL.

An Origin may use P2P-RPL to discover routes to one or more Target(s) identified by one or more unicast/multicast addresses. P2P-RPL allows for the discovery of one Hop-by-hop Route or up to four Source Routes per Target. The discovered routes are guaranteed to meet the specified routing metric constraints but may not be the best available. P2P-RPL may fail to discover any route if the specified routing constraints are overly strict.

The Origin initiates a P2P-RPL route discovery by forming a temporary DAG rooted at itself. The DIOs used to create the temporary DAG are identified by a new Mode of Operation (P2P Route Discovery mode defined in Section 6). The DIOs listing the P2P Route Discovery mode as the Mode of Operation are henceforth referred to as the P2P mode DIOs. A P2P mode DIO always carries exactly one P2P Route Discovery Option (P2P-RDO, defined in Section 7) in which the Origin specifies the following information:

- o The IPv6 address of a Target. This could be a unicast address or a multicast address. Any additional Targets may be specified by including one or more RPL Target Options [RFC6550] inside the DIO.

- o The nature of the route(s) to be discovered: Hop-by-hop or Source Routes. This specification allows for the discovery of one Hop-by-hop Route or up to four Source Routes per Target.
- o The desired number of routes (if Source Routes are being discovered).
- o Whether the Target(s) should send P2P Discovery Reply Object (P2P-DRO) messages (defined in Section 8) back to the Origin on receiving a DIO message. A P2P-DRO message carries a discovered Source Route back to the Origin or establishes a Hop-by-hop Route between the Origin and the Target.

A P2P-RDO also includes the best route from the Origin that the router, generating the P2P mode DIO, has seen so far.

A P2P mode DIO MAY also carry:

- o One or more Metric Container Options to specify:
 - * The relevant routing metrics.
 - * The constraints that the discovered route must satisfy. These constraints also limit how far the DIOs message may travel.
- o One or more RPL Target options to specify additional unicast or multicast Targets.

As the routers join the temporary DAG, they keep track of the best route(s) (so far from the Origin) they have seen and advertise these routes, along with the corresponding routing metrics, in their P2P mode DIOs. A router, including the Target(s), discards a received P2P mode DIO if the aggregated routing metrics on the route advertised by the DIO do not satisfy the listed constraints. These constraints can be used to limit the propagation of P2P mode DIO messages. A router may also discard a received P2P mode DIO if it does not wish to be a part of the discovered route due to limited resources or due to policy reasons.

When a Target receives a P2P mode DIO, it contains inside the P2P-RDO a complete Source Route from the Origin to this Target. Since the links in the discovered route have bidirectional reachability (Section 7), the Target may use the discovered route to reach the Origin. Thus, a router that provides a particular service in the LLN (e.g. an outside temperature server) could initiate a P2P-RPL route discovery listing all its potential clients as Targets, thereby allowing the clients to discover a Source Route back to the server. In this case, the Origin (the server) might want to disable the

generation of P2P-DRO messages by the Targets (the clients). If the Origin has requested P2P-DRO messages to be sent back, the Target may select the discovered route in the received DIO for further processing as described next. This document does not specify a particular method for the Target to use to select a route for further processing. Example methods include selecting any route that meets the constraints or selecting the best route(s) discovered over a certain time period.

If one or more Source Route(s) are being discovered, the Target sends the selected Source Route(s) to the Origin via P2P-DRO messages with one P2P-DRO message carrying one discovered route. On receiving a P2P-DRO message, the Origin stores the discovered route in its memory. This specification allows the Origin to discover up to four Source Routes per Target, thereby allowing the Origin to have sufficient ready-to-use alternatives should one or more of these routes fail. If a Hop-by-hop Route is being discovered, the Target sends a P2P-DRO message containing the selected route to the Origin. The P2P-DRO message travels back to the Origin along the selected route, establishing state for the Forward Route in the routers on the path.

The Target may request the Origin to acknowledge the receipt of a P2P-DRO message by sending back a P2P-DRO Acknowledgement (P2P-DRO-ACK) message (defined in Section 10). The Origin unicasts a P2P-DRO-ACK message to the Target. If the Target does not receive the requested P2P-DRO-ACK within a certain time interval of sending a P2P-DRO, it resends the P2P-DRO message (up to a certain number of times) carrying the same route as before.

The use of trickle timers to delay the propagation of DIO messages may cause some nodes to generate these messages even when the desired routes have already been discovered. In order to preempt the generation of such unnecessary messages, the Target may set a "Stop" flag in the P2P-DRO message to let the nodes in the LLN know about the completion of the route discovery process. The routers receiving such a P2P-DRO should not generate any more DIOs for this temporary DAG. Neither should they process any received DIOs for this temporary DAG in future. However, such routers must still process the P2P-DROs received for this temporary DAG.

6. P2P Route Discovery Mode Of Operation

This section specifies a new RPL Mode of Operation (MOP), P2P Route Discovery Mode (or P2P mode, for short), with value TBD1. A DIO message, listing P2P mode as the MOP, is identified as performing a P2P-RPL route discovery by creating a temporary DAG. A P2P mode DIO

MUST carry exactly one P2P Route Discovery Option (P2P-RDO, specified in Section 7).

6.1. Setting a P2P Mode DIO

The Base Object in a P2P mode DIO message MUST be set in the following manner:

- o RPLInstanceID: RPLInstanceID MUST be a local value as described in Section 5.1 of [RFC6550]. The Origin chooses the RPLInstanceID to be used for a particular route discovery in accordance with the following rules:
 - * The Origin SHOULD NOT reuse an RPLInstanceID for a route discovery if some routers might still maintain membership in the DAG the Origin had initiated for the previous route discovery using this RPLInstanceID. As described in Section 7, a router's membership in a DAG created for a P2P-RPL route discovery lasts for the time duration (say 'l' seconds) indicated by the L field inside the P2P-RDO. In general, there is no upper bound on the time duration by when all the routers have left the DAG created for a P2P-RPL route discovery. In the specific case where the discovered route must be at most 'n' hops in length, all the routers must have left the DAG "(n+1)*l" seconds after its initiation by the Origin. In practice, all the routers should have joined the DAG within 'l' seconds of its initiation (since the route discovery must complete while the Origin still belongs to the DAG) and hence all the routers should have left the DAG within "2*l" seconds of its initiation. Hence, it is usually sufficient that the Origin wait for twice the duration indicated by the L field inside the P2P-RDO used for the previous route discovery before reusing the RPLInstanceID for a new route discovery. Individual P2P-RPL deployments are encouraged to share their experience with various RPLInstanceID reuse policies to help guide the development of standards track version of the protocol.
 - * When initiating a new route discovery to a particular Target, the Origin MUST NOT reuse the RPLInstanceID used in a previous route discovery to this Target if the state created during the previous route discovery might still exist in some routers. Note that it is possible that the previous route discovery did not succeed yet some routers still ended up creating state. The Default Lifetime and Lifetime Unit parameters in the DODAG Configuration Option specify the lifetime of the state the routers, including the Origin and the Target, maintain for a Hop-by-hop or a Source Route discovered using P2P-RPL. Suppose

this lifetime is 'X' seconds. As discussed above, any state created during the previous route discovery was likely created within "2*1" seconds of its initiation. Hence, it is sufficient that the Origin lets a time duration equal to "X+2*1" seconds pass since the initiation of the previous route discovery before initiating a new route discovery to the same Target using the same RPLInstanceID.

- o Version Number: MUST be set to zero. The temporary DAG used for P2P-RPL route discovery does not exist long enough to have new versions.
- o Grounded (G) Flag: This flag MUST be set to one. Unlike a global RPL instance, the concept of a floating DAG, used to provide connectivity within a sub-DAG detached from a grounded DAG, does not apply to a local RPL instance. Hence, an Origin MUST always set the G flag to one when initiating a P2P-RPL route discovery. Further, clause 3 of Section 8.2.2.2 in [RFC6550] does not apply and a node MUST NOT initiate a new DAG if it does not have any parent left in a P2P-RPL DAG.
- o Mode of Operation (MOP): MUST be set to TBD1, corresponding to P2P Route Discovery mode.
- o DTSN: MUST be set to zero on transmission and ignored on reception.
- o DODAGPreference (Prf): This field MUST be set to zero (least preferred).
- o DODAGID: This field MUST be set to an IPv6 address of the Origin.
- o The other fields in the DIO Base Object can be set in the desired fashion as per the rules described in [RFC6550].

A received P2P mode DIO MUST be discarded if it does not follow the above-listed rules regarding the RPLInstanceID, Version Number, G flag, MOP and Prf fields inside the base object.

The DODAG Configuration Option, inside a P2P mode DIO MUST be set in the following manner:

- o The Origin MUST set the MaxRankIncrease parameter to zero to disable local repair of the temporary DAG. A received P2P mode DIO MUST be discarded if the MaxRankIncrease parameter inside the DODAG Configuration Option is not zero.

- o The Origin SHOULD set the Trickle parameters (DIOIntervalDoublings, DIOIntervalMin, DIORedundancyConstant) as recommended in Section 9.2.
- o The Origin sets the Default Lifetime and Lifetime Unit parameters to indicate the lifetime of the state the routers, including the Origin and the Target(s), maintain for a Hop-by-hop or a Source Route discovered using P2P-RPL.
- o The Origin sets the other fields in the DODAG Configuration Option, including the OCP identifying the Objective function, in the desired fashion as per the rules described in [RFC6550].
- o An Intermediate Router (or a Target) MUST set various fields in the DODAG Configuration Option in the outgoing P2P mode DIOs to the values they had in the incoming P2P mode DIOs for this DAG.

A default DODAG Configuration Option comes in effect if a P2P mode DIO does not carry an explicit one. The default DODAG Configuration Option has the following parameter values:

- o Authentication Enabled: 0
- o DIOIntervalMin: 6, which translates to 64ms as the value for Imin parameter in Trickle operation. This value is roughly one order of magnitude larger than the typical transmission delay on IEEE 802.15.4 links and corresponds to the recommendation in Section 9.2 for well-connected topologies.
- o DIORedundancyConstant: 1. See the discussion in Section 9.2.
- o MaxRankIncrease: 0 (to disable local repair of the temporary DAG).
- o Default Lifetime: 0xFF, to correspond to infinity.
- o Lifetime Unit: 0xFFFF, to correspond to infinity.
- o Objective Code Point: 0, i.e., OF0 [RFC6552] is the default objective function.
- o The remaining parameters have default values as specified in [RFC6550].

Individual P2P-RPL deployments are encouraged to share their experience with these default values to help guide the development of standards track version of the protocol.

The routing metrics and constraints [RFC6551] used in P2P-RPL route

discovery are included in one or more Metric Container Options [RFC6550] inside the P2P mode DIO. Note that a DIO need not include a Metric Container if OF0 is the objective function in effect. In that case, a P2P mode DIO may still specify an upper limit on the maximum rank, that a router may have in the temporary DAG, inside the P2P-RDO.

A P2P mode DIO:

- o MUST carry one (and only one) P2P-RDO. The P2P-RDO allows for the specification of one unicast or multicast address for the Target. A received P2P mode DIO MUST be discarded if it does not contain exactly one P2P-RDO.
- o MAY carry one or more RPL Target Options to specify additional unicast/multicast addresses for the Target. If a unicast address is specified, it MUST be a global address or a unique local address.
- o MAY carry one or more Metric Container Options to specify routing metrics and constraints.
- o MAY carry one or more Route Information Options [RFC6550]. In the context of P2P-RPL, a Route Information Option advertizes to the Target(s) the Origin's connectivity to the prefix specified in the option.

An RPL Option, besides the ones listed above, MUST be ignored when found inside a received P2P mode DIO and MUST NOT be included in the P2P mode DIOs the receiving router generates.

In accordance with core RPL, a P2P mode DIO MUST propagate via link-local multicast. The IPv6 source address in a P2P mode DIO MUST be a link-local address and the IPv6 destination address MUST be the link-local multicast address all-RPL-nodes [RFC6550]. A P2P mode DIO MUST be transmitted on all interfaces the router has in this RPL domain [I-D.ietf-roll-terminology].

7. P2P Route Discovery Option (P2P-RDO)

This section defines a new RPL control message option: the P2P Route Discovery Option (P2P-RDO).

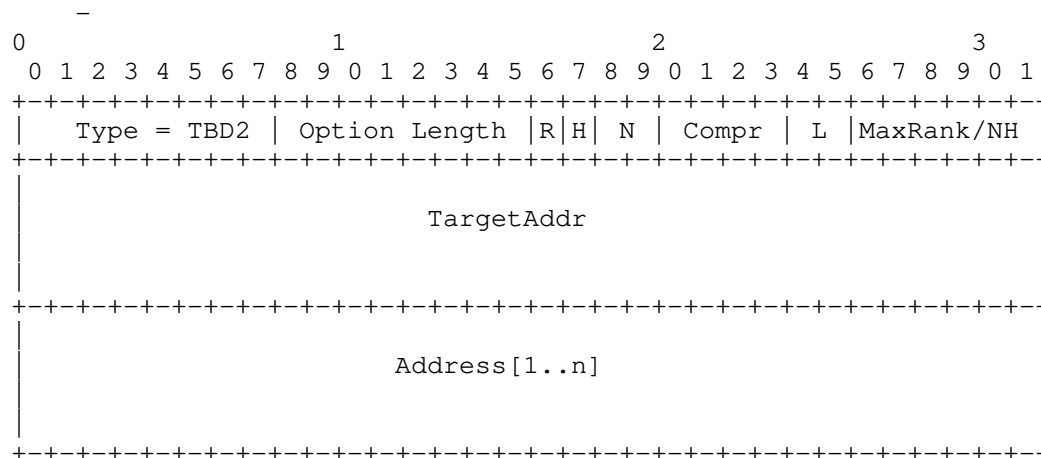


Figure 1: Format of P2P Route Discovery Option (P2P-RDO)

The format of a P2P Route Discovery Option (P2P-RDO) is illustrated in Figure 1. A P2P mode DIO and a P2P-DRO (defined in Section 8) message MUST carry exactly one P2P-RDO. A P2P-RDO consists of the following fields:

- o Option Type: TBD2.
- o Option Length: 8-bit unsigned integer, representing the length in octets of the option, not including the Option Type and Option Length fields.
- o Reply (R): The Origin sets this flag to one to allow the Target(s) to send P2P-DRO messages back to the Origin. If this flag is zero, a Target MUST NOT generate any P2P-DRO message.
- o Hop-by-hop (H): This flag is valid only if the R flag is set to one. The Origin sets this flag to one if it desires Hop-by-hop Routes. The Origin sets this flag to zero if it desires Source Routes. This specification allows for the establishment of one Hop-by-hop route or up to four Source Routes per Target. The Hop-by-hop Route is established in the Forward direction, i.e. from the Origin to the Target. This specification does not allow for the establishment of Hop-by-hop Routes in the Reverse direction.
- o Number of Routes (N): This field is valid only if the R flag is one and H flag is zero, i.e. the Targets are allowed to generate P2P-DRO messages carrying discovered Source Routes back to the Origin. In this case, the value in the N field plus one indicates

the number of Source Routes that each Target should convey to the Origin. When Hop-by-hop Routes are being discovered, the N field MUST be set to zero on transmission and ignored on reception.

- o Compr: 4-bit unsigned integer indicating the number of prefix octets that are elided from the Target field and the Address vector. For example, Compr value will be zero if full IPv6 addresses are carried in the Target field and the Address vector.
- o Life Time (L): A 2-bit field that indicates the exact duration a router joining the temporary DAG, including the Origin and the Target(s), MUST maintain its membership in the DAG. A router MUST leave the temporary DAG once the time elapsed since it joined reaches the value indicated by this field. The mapping between the value in this field and the duration of the router's membership in the temporary DAG is as follows:

- * 0x00: 1 second;
- * 0x01: 4 seconds;
- * 0x02: 16 seconds;
- * 0x03: 64 seconds;

The Origin sets this field based on its expectation regarding the time required for the route discovery to complete, which includes the time required for the DIOs to reach the Target(s) and the P2P-DROs to travel back to the Origin. The time required for the DIOs to reach the Target(s) would in turn depend on the Trickle parameters (Imin and the redundancy constant) as well as the expected distance (in terms of hops and/or ETX) to the Target(s). While deciding the value in this field, the Origin should also take in account the fact that all routers joining the temporary DAG would need to stay in the DAG for this much time.

- o MaxRank/NH:
 - * When a P2P-RDO is included in a P2P mode DIO, this field indicates the upper limit on the integer portion of the rank (calculated using the DAGRank() macro defined in [RFC6550]) that a router may have in the temporary DAG being created. An Intermediate Router MUST NOT join a temporary DAG being created by a P2P mode DIO if the integer portion of its rank would be equal to or higher (in numerical value) than the MaxRank limit. A Target can join the temporary DAG at a rank whose integer portion is equal to the MaxRank. A router MUST discard a received P2P mode DIO if the integer part of the advertized

rank equals or exceeds the MaxRank limit. A value 0 in this field indicates that the MaxRank is infinity.

- * When a P2P-RDO is included in a P2P-DRO message, this field indicates the index of the next hop address inside the Address vector.
- o TargetAddr: An IPv6 address of the Target after eliding Compr number of prefix octets. When the P2P-RDO is included in a P2P mode DIO, this field may contain a unicast address or a multicast address. If a unicast address is specified, it MUST be a global address or a unique local address. Any additional Target addresses can be specified by including one or more RPL Target Options [RFC6550] in the DIO. When the P2P-RDO is included in a P2P-DRO, this field MUST contain a unicast global or unique local IPv6 address of the Target generating the P2P-DRO.
- o Address[1..n]: A vector of IPv6 addresses representing a complete route so far in the Forward direction:
 - * Each element in the Address vector has size (16 - Compr) octets and MUST contain a valid global or unique local IPv6 address with first Compr octets elided.
 - * The total number of elements inside the Address vector is given by $n = (\text{Option Length} - 2 - (16 - \text{Compr})) / (16 - \text{Compr})$.
 - * The IPv6 address that a router adds to the vector MUST belong to the interface on which the router received the DIO containing this P2P-RDO. Further, this interface MUST NOT be an Ingress-only Interface. This allows the route accumulated in the Address vector to be a Bidirectional Route that can be used by a Target to send a P2P-DRO message to the Origin.
 - * The Address vector MUST carry the accumulated route in the Forward direction, i.e., the first element in the Address vector must contain the IPv6 address of the router next to the Origin and so on.
 - * The Origin and Target addresses MUST NOT be included in the Address vector.
 - * A router adding its address to the vector MUST ensure that any of its addresses do not already exist in the vector. A Target specifying a complete route in the Address vector MUST ensure that the vector does not contain any address more than once.

- * The Address vector MUST NOT contain any multicast addresses.

8. The P2P Discovery Reply Object (P2P-DRO)

This section defines two new RPL Control Message types, the P2P Discovery Reply Object (P2P-DRO), with code TBD3, and the Secure P2P-DRO, with code TBD4. A P2P-DRO serves one of the following functions:

- o Carry a discovered Source Route from a Target to the Origin;
- o Establish a Hop-by-hop Route as it travels from a Target to the Origin.

A P2P-DRO message can also serve the function of letting the routers in the LLN know that a P2P-RPL route discovery is complete and no more DIO messages need to be generated for the corresponding temporary DAG. A P2P-DRO message MUST carry one (and only one) P2P-RDO whose TargetAddr field MUST contain a unicast IPv6 address of the Target that generates the P2P-DRO. A P2P-DRO message MUST travel from the Target to the Origin via link-local multicast along the route specified inside the Address vector in the P2P-RDO, as included in the P2P-DRO. The IPv6 source address in a P2P-DRO message MUST be a link-local address and the IPv6 destination address MUST be the link-local multicast address all-RPL-nodes [RFC6550]. A P2P-DRO message MUST be transmitted on all interfaces the router has in this RPL domain [I-D.ietf-roll-terminology].

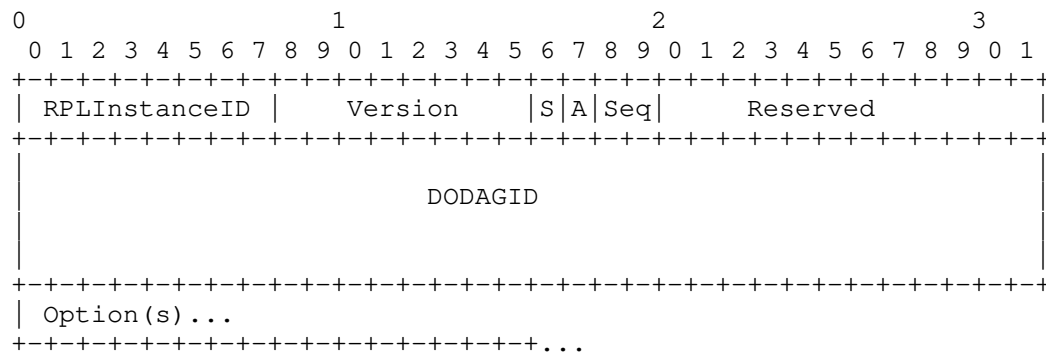


Figure 2: Format of the base P2P Discovery Reply Object (P2P-DRO)

The format of the base P2P Discovery Reply Object (P2P-DRO) is shown in Figure 2. A base P2P-DRO consists of the following fields:

- o RPLInstanceID: The RPLInstanceID of the temporary DAG used for route discovery.
- o Version: The Version of the temporary DAG used for route discovery. Since a temporary DAG always has value zero for the Version, this field MUST always be set to zero.
- o Stop (S): This flag, when set to one by a Target, indicates that the P2P-RPL route discovery is over. All the routers receiving such a P2P-DRO, including the ones not listed in the route carried inside P2P-RDO,
 - * SHOULD NOT process any more DIOs received for this temporary DAG;
 - * SHOULD NOT generate any more DIOs for this temporary DAG;
 - * SHOULD cancel any pending DIO transmission for this temporary DAG.

Note that the Stop flag serves to stop further DIO generation/processing for a P2P-RPL route discovery but it does not affect the processing of P2P-DRO messages at either the Origin or the Intermediate Routers. In other words, a router (the Origin or an Intermediate Router) MUST continue to process the P2P-DRO messages even if an earlier P2P-DRO message (with the same RPLInstanceID and DODAGID fields) had the Stop flag set to one. When set to zero, this flag does not imply any thing and MUST be ignored on reception.

- o Ack Required (A): This flag, when set to one by the Target, indicates that the Origin MUST unicast a P2P-DRO-ACK message (defined in Section 10) to the Target when it receives the P2P-DRO.
- o Sequence Number (Seq): This 2-bit field indicates the sequence number for the P2P-DRO. This field is relevant when the A flag is set to one, i.e., the Target requests an acknowledgement from the Origin for a received P2P-DRO. The Origin includes the RPLInstanceID, the DODAGID and the Sequence Number of the received P2P-DRO inside the P2P-DRO-ACK message it sends back to the Target.
- o Reserved: These bits are reserved for future use. These bits MUST be set to zero on transmission and MUST be ignored on reception.
- o DODAGID: The DODAGID of the temporary DAG used for route discovery. The DODAGID also identifies the Origin. The

RPLInstanceID, the Version and the DODAGID together uniquely identify the temporary DAG used for route discovery and can be copied from the DIO message advertizing the temporary DAG.

o Options: The P2P-DRO message:

- * MUST carry one (and only one) P2P-RDO that MUST specify a complete route between the Target and the Origin. A received P2P-DRO message MUST be discarded if it does not contain exactly one P2P-RDO.
- * MAY carry one or more Metric Container Options that contains the aggregated routing metrics values for the route specified in P2P-RDO.

An RPL Option, besides the ones listed above, MUST be ignored when found inside a received P2P-DRO message.

8.1. Secure P2P-DRO

A Secure P2P-DRO message follows the format in Figure 7 of [RFC6550], where the base format is the base P2P-DRO shown in Figure 2.

8.2. Setting a P2P-RDO Carried in a P2P Discovery Reply Object

A P2P Discovery Reply Object MUST carry one (and only one) P2P-RDO, which MUST be set as defined in Section 7. Specifically, the following fields MUST be set as specified next:

- o Reply (R): This flag MUST be set to zero on transmission and ignored on reception.
- o Hop-by-Hop (H): The H flag in the P2P-RDO included in a P2P-DRO message MUST have the same value as the H flag in the P2P-RDO inside the corresponding DIO message.
- o Number of Routes (N): This field MUST be set to zero on transmission and ignored on reception.
- o Life Time (L): This field MUST be set to zero on transmission and ignored on reception.
- o MaxRank/NH: This field indicates the index of the next hop address in the Address vector. When a Target generates a P2P-DRO message, the NH field is set to $n = (\text{Option Length} - 2 - (16 - \text{Compr})) / (16 - \text{Compr})$.

- o TargetAddr: This field MUST contain a unicast global or unique local IPv6 address of the Target generating the P2P-DRO.
- o Address[1..n]: The Address vector MUST contain a complete route between the Origin and the Target such that the first element in the vector contains the IPv6 address of the router next to the Origin and the last element contains the IPv6 address of the router next to the Target.

9. P2P-RPL Route Discovery By Creating a Temporary DAG

This section details the P2P-RPL route discovery operation.

9.1. Joining a Temporary DAG

All the routers participating in a P2P-RPL route discovery, including the Origin and the Target(s), MUST join the temporary DAG being created for the purpose. When a router joins a temporary DAG advertized by a P2P mode DIO, it MUST maintain its membership in the temporary DAG for the duration indicated by the L field inside the P2P-RDO. The only purpose of a temporary DAG's existence is to facilitate the P2P-RPL route discovery process. The temporary DAG MUST NOT be used to route data packets. In other words, joining a temporary DAG does not allow a router to provision routing table entries listing the router's parents in the temporary DAG as the next hops (i.e., the last bullet point in Section 3.2.8 of [RFC6550] is not applicable when the DAG is a temporary DAG created for the purpose of a P2P-RPL route discovery.).

Given the nature of a temporary DAG created for a P2P-RPL route discovery, this document disallows the solicitation of P2P mode DIOs using DODAG Information Solicitation (DIS) messages as described in [RFC6550]. A router participating in a P2P-RPL route discovery MUST NOT reset its Trickle timer that controls the transmission of P2P mode DIOs in response to a multicast DIS. Also, the router MUST NOT send a P2P mode DIO in response to a unicast DIS. In other words, the rules in Section 8.3 of [RFC6550] regarding a router's response to a multicast/unicast DIS are not applicable for P2P mode DIOs.

A router MUST detach from the temporary DAG created for a P2P-RPL route discovery once the duration of its membership in the DAG has reached the value indicated by the L field inside the P2P-RDO. After receiving a P2P-DRO with the Stop flag set to one, a router SHOULD NOT send or process any more DIOs for this temporary DAG and SHOULD also cancel any pending DIO transmission.

9.2. Trickle Operation For P2P Mode DIOs

An RPL router uses a Trickle timer [RFC6206] to control DIO transmissions. The Trickle control of DIO transmissions provides quick resolution of any "inconsistency" while avoiding redundant DIO transmissions. The Trickle algorithm also imparts protection against loss of DIOs due to inherent lack of reliability in LLNs. When controlling the transmissions of a P2P mode DIO, a Trickle timer SHOULD follow the following rules:

- o The receipt of a P2P mode DIO, that allows the router to advertise a better route (in terms of the routing metrics and the OF in use) than before, is considered "inconsistent" and hence resets the Trickle timer. Note that the first receipt of a P2P mode DIO advertising a particular temporary DAG is always considered an "inconsistent" event.
- o The receipt of a P2P mode DIO from a parent in the temporary DAG is considered neither "consistent" nor "inconsistent" if it does not allow the router to advertise a better route than before. Thus, the receipt of such DIOs has no impact on the Trickle operation. Note that this document does not impose any requirements on how a router might choose its parents in the temporary DAG.
- o The receipt of a P2P mode DIO is considered "consistent" if the source of the DIO is not a parent in the temporary DAG and either of the following conditions is true:
 - * The DIO advertises a better route than the router but does not allow the router to advertise a better route itself; or
 - * The DIO advertises a route as good as the route (to be) advertised by the router.

Note that the Trickle algorithm's DIO suppression rules are in effect at all times. Hence, a P2P-RPL router may suppress a DIO transmission even if it has not made any DIO transmission yet.

- o The receipt of a P2P mode DIO, that advertises a worse route than what the router advertises (or would advertise when it gets a chance to generate its DIO), is considered neither "consistent" nor "inconsistent", i.e., the receipt of such a DIO has no impact on the Trickle operation.
- o The Imin parameter SHOULD be set taking in account the connectivity within the network. For highly connected networks, a small Imin value (of the order of the typical transmission delay

for a DIO) may lead to congestion in the network as a large number of routers reset their Trickle timers in response to the first receipt of a DIO from the Origin. These routers would generate their DIOs within Imin interval and cause additional routers to reset their trickle timers and generate more DIOs. Thus, for highly connected networks, the Imin parameter SHOULD be set to a value at least one order of magnitude larger than the typical transmission delay for a DIO. For sparsely connected networks, the Imin parameter can be set to a value that is a small multiple of the typical transmission delay for a DIO. Note that the Imin value has a direct impact on the time required for a P2P-RPL route discovery to complete. In general, the time required for a P2P-RPL route discovery would increase approximately linearly with the value of the Imin parameter. Since the route discovery must complete while the Origin still belongs to the temporary DAG created for the purpose, the Origin should set the time duration a router maintains its membership in the temporary DAG (indicated by the L field inside the P2P-RDO) to a large enough value taking in account the Imin value as well as the expected distance (in terms of hops and/or ETX) to the Target(s).

- o The Imax parameter SHOULD be set to a large value (several orders of magnitude higher than the Imin value) and is unlikely to be critical for P2P-RPL operation. This is because the first receipt of a P2P mode DIO for a particular temporary DAG is considered an inconsistent event and would lead to resetting of Trickle timer duration to the Imin value. Given the temporary nature of the DAGs used in P2P-RPL, Trickle timer may not get a chance to increase much.
- o The recommended value of redundancy constant "k" is 1. With this value of "k", a DIO transmission will be suppressed if the router receives even a single "consistent" DIO during a timer interval. This setting for the redundancy constant is designed to reduce the number of messages generated during a route discovery process and is suitable for the environments with low or moderate packet loss rates. However, this setting may result in an increase in the time required for the route discovery process to complete. A higher value for the redundancy constant may be more suitable in
 - * Environments with high packet loss rates; or
 - * Deployments where the time required for the route discovery process to complete needs to be as small as possible; or
 - * Deployments where specific destinations are reachable only through specific intermediate routers (and hence these intermediate routers should not suppress their DIOs).

A particular deployment should take in account the above mentioned factors when deciding the value of the redundancy constant.

Individual P2P-RPL deployments are encouraged to share their experience with these rules to help guide the development of standards track version of the protocol. Applicability Statements that specify the use of P2P-RPL MUST provide guidance for setting Trickle parameters, particularly Imin and the redundancy constant.

9.3. Processing a P2P Mode DIO

The rules for DIO processing and transmission, described in Section 8 of RPL [RFC6550], apply to P2P mode DIOs as well except as modified in this document. In particular, in accordance with Section 8.2.3 of RPL [RFC6550], a received P2P mode DIO MUST be discarded if it is malformed according to the rules specified in this document and in [RFC6550].

The following rules for processing a received P2P mode DIO apply to both Intermediate Routers and the Target.

A router SHOULD discard a received P2P mode DIO with no further processing if it does not have bidirectional reachability with the neighbor that generated the received DIO. Note that bidirectional reachability does not mean that the link must have the same values for a routing metric in both directions. A router SHOULD calculate the values of the link-level routing metrics included in the received DIO taking in account the metric's value in both Forward and Reverse directions. Bidirectional reachability along a discovered route allows the Target to use this route to reach the Origin. In particular, the P2P-DRO messages travel from the Target to the Origin along a discovered route.

A router MUST discard a received P2P mode DIO with no further processing:

- o If the DIO advertises INFINITE_RANK as defined in Section 17 of [RFC6550].
- o If the integer part of the rank advertised in the DIO equals or exceeds the MaxRank limit listed in the P2P Route Discovery Option.
- o If the routing metric values do not satisfy one or more of the mandatory route constraints listed in the DIO or if the router cannot evaluate the mandatory route constraints, e.g., if the router does not support the metrics used in the constraints.

- o If the router previously received a P2P-DRO message with the same RPLInstanceID and DODAGID as the received DIO and with the Stop flag set to one.

The router MUST check the Target addresses listed in the P2P-RDO and any RPL Target Options included in the received DIO. If one of its IPv6 addresses is listed as a Target address or if it belongs to the multicast group specified as one of the Target addresses, the router considers itself a Target and processes the received DIO as specified in Section 9.5. Otherwise, the router considers itself an Intermediate Router and processes the received DIO as specified in Section 9.4.

9.4. Additional Processing of a P2P Mode DIO At An Intermediate Router

An Intermediate Router MUST discard a received P2P mode DIO with no further processing

- o if the DIO is received on an Ingress-only Interface; or
- o if the receiving interface does not have a global or unique local IPv6 address configured with the address prefix implied by the Compr field in the P2P-RDO inside the received DIO; or
- o if the router can not uniquely identify the address prefix implied by the Compr field in the P2P-RDO (this might happen if the receiving interface has multiple global/unique-local IPv6 addresses, each configured with a different address prefix); or
- o if adding its IPv6 address to the route in the Address vector inside the P2P-RDO would result in the route containing multiple addresses belonging to this router.

On receiving a P2P mode DIO, an Intermediate Router MUST do the following. The router MUST determine whether this DIO advertises a better route than the router itself and whether the receipt of the DIO would allow the router to advertise a better route than before. Accordingly, the router SHOULD consider this DIO as consistent/inconsistent from Trickle perspective as described in Section 9.2. Note that the route comparison in a P2P-RPL route discovery is performed using the parent selection rules of the OF in use as specified in Section 14 of RPL [RFC6550]. If the received DIO would allow the router to advertise a better route, the router MUST add a unicast IPv6 address of the receiving interface (after eliding Compr prefix octets) to the route in the Address vector inside the P2P-RDO and remember this route for inclusion in its future DIOs.

When an Intermediate Router adds an IPv6 address to a route, it MUST

ensure that

- o the IPv6 address is a unicast global or unique local IPv6 address assigned to the interface on which the DIO containing the route was received;
- o the IPv6 address was configured with the address prefix implied by the Compr field in the P2P-RDO inside the received DIO;

To improve the diversity of the routes being discovered, an Intermediate Router SHOULD keep track of multiple routes (as long as all these routes are the best seen so far), one of which SHOULD be selected in a uniform random manner for inclusion in the P2P-RDO inside the router's next DIO. Note that the route accumulation in a P2P mode DIO MUST take place even if the Origin does not want any P2P-RDO messages to be generated (i.e., the R flag inside the P2P-RDO is set to zero). This is because the Target may still be able to use the accumulated route as a source route to reach the Origin.

9.5. Additional Processing of a P2P Mode DIO At The Target

The Target MAY remember the discovered route contained in the P2P-RDO in the received DIO for use as a Source Route to reach the Origin. The lifetime of this Source Route is specified by the Default Lifetime and Lifetime Unit parameters inside the DODAG Configuration Option currently in effect. This lifetime can be extended (or shortened) appropriately following a hint from an upper-layer protocol.

If the Reply flag inside the P2P-RDO in the received DIO is set to one, the Target MUST select one or more discovered routes and send one or more P2P-DRO messages, carrying one discovered route each, back to the Origin. If the H flag inside the P2P-RDO is set to one, the Target needs to select one route and send a P2P-DRO message along this route back to the Origin. As this P2P-DRO message travels back to the Origin, the routers on the path establish hop-by-hop routing state, thereby establishing a Hop-by-hop Route in the Forward direction. If the H flag is set to zero, the number of Source Routes to be selected (and the number of P2P-DRO messages to be sent back) is given by one plus the value of the N field in the P2P-RDO. The Target may select the discovered route inside the received DIO as (one of) the route(s) that would be carried inside a P2P-DRO message back to the Origin. This document does not prescribe a particular method for the Target to select the routes. Example methods include selecting each route that meets the specified routing constraints until the desired number have been selected or selecting the best routes discovered over a certain time period. If multiple routes are to be selected, the Target SHOULD avoid selecting routes that have

large segments in common.

If the Target selects the route contained in the P2P-RDO in the received DIO, it sends a P2P-DRO message back to the Origin (identified by the DODAGID field in the DIO). The P2P-DRO message MUST include a P2P-RDO that contains the selected route inside the Address vector. Various fields inside the P2P-RDO MUST be set as specified in Section 8.2. The Target MAY set the A flag inside the P2P-DRO message to one if it desires the Origin to send back a P2P-DRO-ACK message on receiving the P2P-DRO. In this case, the Target waits for P2P_DRO_ACK_WAIT_TIME duration for the P2P-DRO-ACK message to arrive. Failure to receive the P2P-DRO-ACK message within this time duration causes the Target to retransmit the P2P-DRO message. The Target MAY retransmit the P2P-DRO message in this fashion up to MAX_P2P_DRO_RETRANSMISSIONS times. Both P2P_DRO_ACK_WAIT_TIME and MAX_P2P_DRO_RETRANSMISSIONS are configurable parameters to be decided based on the characteristics of individual deployments. Note that all P2P-DRO transmissions and retransmissions MUST take place while the Target is still a part of the temporary DAG created for the route discovery. A Target MUST NOT transmit a P2P-DRO if it no longer belongs to this DAG.

The Target MAY set the Stop flag inside the P2P-DRO message to one if

- o this router is the only Target specified in the corresponding DIO, i.e., the corresponding DIO specified a unicast address of the router as the TargetAddr inside the P2P-RDO with no additional Targets specified via RPL Target Options; and
- o the Target has already selected the desired number of routes.

The Target MAY include a Metric Container Option in the P2P-DRO message. This Metric Container contains the end-to-end routing metric values for the route specified in the P2P-RDO. The Target MUST transmit the P2P-DRO message via a link-local multicast.

A Target MUST NOT forward a P2P mode DIO any further if no other Targets are to be discovered, i.e., if a unicast IPv6 address (of this Target) is specified as the TargetAddr inside the P2P-RDO and no additional Targets are specified via RPL Target Options inside the DIOs for this route discovery. Otherwise, the Target MUST generate DIOs for this route discovery as an Intermediate Router would.

9.6. Processing a P2P-DRO At An Intermediate Router

If the DODAGID field in the received P2P-DRO does not list a router's own IPv6 address, the router considers itself an Intermediate Router and MUST process the received message in the following manner:

- o The router MUST discard the received P2P-DRO with no further processing if it does not belong to the temporary DAG identified by the RPLInstanceID and the DODAGID fields in the P2P-DRO.
- o If the Stop flag inside the received P2P-DRO is set to one, the router SHOULD NOT send or receive any more DIOs for this temporary DAG and SHOULD cancel any pending DIO transmission.
- o The router MUST ignore any Metric Container Options contained in the P2P-DRO message.
- o If Address[NH] element inside the P2P-RDO lists the router's own unicast IPv6 address, the router is a part of the route carried in the P2P-RDO. In this case, the router MUST do the following:
 - * To prevent loops, the router MUST discard the P2P-DRO message with no further processing if the Address vector in the P2P-RDO includes multiple IPv6 addresses assigned to the router's interfaces.
 - * If the H flag inside the P2P-RDO is one, the router MUST store the state for the Forward Hop-by-hop route carried inside the P2P-RDO. This state consists of:
 - + The RPLInstanceID and the DODAGID fields of the P2P-DRO.
 - + The route's destination, the Target (identified by TargetAddr field inside P2P-RDO).
 - + The IPv6 address of the next hop, Address[NH+1] (unless NH value equals the number of elements in the Address vector, in which case the Target itself is the next hop).

This Hop-by-hop routing state MUST expire at the end of the lifetime specified by the Default Lifetime and Lifetime Unit parameters inside the DODAG Configuration Option used in P2P mode DIOs for this route discovery.

- * If the router already maintains a Hop-by-hop state listing the Target as the destination and carrying same RPLInstanceID and DODAGID fields as the received P2P-DRO and the next hop information in the state does not match the next hop indicated in the received P2P-DRO, the router MUST discard the P2P-DRO message with no further processing. Note that this situation would occur in the following two cases:
 - + When the route listed in the Address vector inside the P2P-RDO contains a previously undetected loop. In this case,

the rule above causes the P2P-DRO messages to be discarded.

- + When a Hop-by-hop Route between the Origin and the Target, previously established using the same RPLInstanceID and DODAGID as the route currently being established, still exists and at least partially overlaps the route currently being established.
- * The router MUST decrement the NH field inside the P2P-RDO and send the P2P-DRO message further via link-local multicast.

9.7. Processing a P2P-DRO At The Origin

When a router receives a P2P-DRO message that lists its IPv6 address in the DODAGID field, the router recognizes itself as the Origin for the corresponding P2P-RPL route discovery, notes the Target that originated this message (from the TargetAddr field inside the P2P-RDO) and processes the message in the following manner:

- o The Origin MUST discard the received P2P-DRO with no further processing if it no longer belongs to the temporary DAG identified by the RPLInstanceID and the DODAGID fields in the P2P-DRO.
- o If the Stop flag inside the received P2P-DRO is set to one, the Origin SHOULD NOT generate any more DIOs for this temporary DAG and SHOULD cancel any pending DIO transmission.
- o If the P2P-RDO inside the P2P-DRO has the H flag set to 0, the Address vector inside the P2P-RDO contains a Source Route to this Target. The Origin MUST set the lifetime of this Source Route to the value specified by the Default Lifetime and Lifetime Unit parameters inside the DODAG Configuration Option in the P2P mode DIOs used for this route discovery. This lifetime could be extended (or shortened) appropriately following a hint from an upper-layer protocol.
- o If the P2P-RDO inside the P2P-DRO has the H flag set to 1, the P2P-DRO message is establishing a Hop-by-hop Route to this Target and the Origin MUST store in its memory the state for this Hop-by-hop Route in the manner described in Section 9.6. This Hop-by-hop routing state MUST expire at the end of the lifetime specified by the Default Lifetime and Lifetime Unit parameters inside the DODAG Configuration Option used in P2P mode DIOs for this route discovery. The standards track version of P2P-RPL may consider specifying a signaling mechanism that will allow the Origin to extend (or shorten) the lifetime of a P2P-RPL Hop-by-hop Route following a suitable hint from an upper-layer protocol.

- o If the received P2P-DRO message contains one or more Metric Container Options, the Origin MAY store the values of the routing metrics associated with the discovered route in its memory. This information may be useful in formulating the constraints for any future P2P-RPL route discovery to this Target.
- o If the A flag is set to one in the received P2P-DRO message, the Origin MUST generate a P2P-DRO-ACK message as described in Section 10 and unicast the message to the Target. The Origin MAY use the route just discovered to send the P2P-DRO-ACK message to the Target. Section 12 describes how a packet may be forwarded along a Source/Hop-by-hop Route discovered using P2P-RPL.

10. The P2P Discovery Reply Object Acknowledgement (P2P-DRO-ACK)

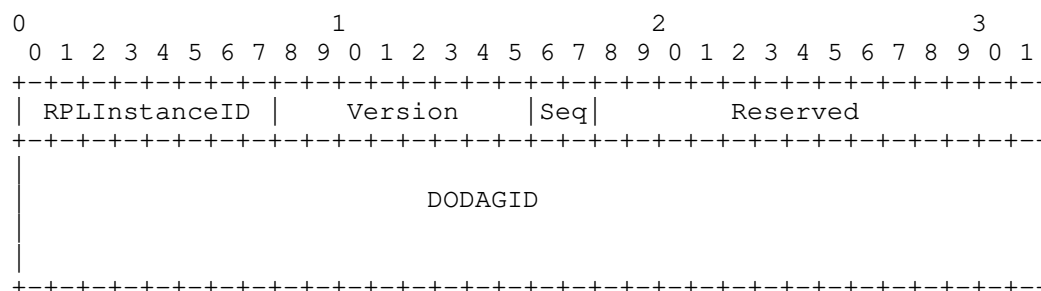


Figure 3: Format of the base P2P Discovery Reply Object Acknowledgement (P2P-DRO-ACK)

A P2P-DRO message may fail to reach the Origin due to a number of reasons. Unlike the DIO messages that benefit from Trickle-controlled retransmissions, the P2P-DRO messages are prone to loss due to unreliable packet transmission in LLNs. Since a P2P-DRO message travels via link-local multicast, it cannot use link-level acknowledgements to improve the reliability of its transmission. Also, an Intermediate Router may drop the P2P-DRO message (e.g., because of its inability to store the state for the Hop-by-hop Route the P2P-DRO is establishing). To protect against the potential failure of a P2P-DRO message to reach the Origin, the Target MAY request the Origin to send back a P2P-DRO Acknowledgement (P2P-DRO-ACK) message on receiving a P2P-DRO message. Failure to receive such an acknowledgement within the P2P_DRO_ACK_WAIT_TIME interval of sending the P2P-DRO message forces the Target to resend the message (as described in Section 9.5).

This section defines two new RPL Control Message types: P2P-DRO

Acknowledgement (P2P-DRO-ACK; with code TBD5) and Secure P2P-DRO-ACK (with code TBD6). A P2P-DRO-ACK message MUST travel as a unicast message from the Origin to the Target. The IPv6 source and destination addresses used in a P2P-DRO-ACK message MUST be global or unique local. The format of a base P2P-DRO-ACK message is shown in Figure 3. Various fields in a P2P-DRO-ACK message MUST have the same values as the corresponding fields in the P2P-DRO message. The field marked as "Reserved" MUST be set to zero on transmission and MUST be ignored on reception. A Secure P2P-DRO-ACK message follows the format in Figure 7 of [RFC6550], where the base format is same as the base P2P-DRO-ACK shown in Figure 3.

11. Secure P2P-RPL Operation

Each RPL control message type, including the ones defined in this document, has a secure version. A secure RPL control message is identified by the value 1 in the most significant bit of the Code field. Each secure RPL control message contains a security section (see Figure 7 of [RFC6550]), whose contents are described in Section 6.1 of [RFC6550]. Sections 6.1, 10 and 19 of [RFC6550] describe core RPL's security apparatus. These sections are applicable to P2P-RPL's secure operation as well except as constrained in this section.

Core RPL allows a router to decide locally on a per-packet basis whether to use security and if yes what Security Configuration (see definition in Section 3) to use (the only exception being the requirement to send a Secure DIO in response to a Secure DIS; see Section 10.2 of [RFC6550]). In contrast, this document requires routers participating in a P2P-RPL route discovery to follow the Origin's lead regarding security. The Origin decides whether to use security and the particular Security Configuration to be used for this purpose. All the routers participating in this route discovery MUST generate only secure control messages if the Origin decides so and MUST use for this purpose the Security Configuration that the Origin chose. The Origin MUST NOT set the "Key Identifier Mode" field inside the chosen Security Configuration to value 1 since this setting indicates the use of a per-pair key which is not suitable for securing messages that travel by (link local) multicast (e.g. DIOs) or that travel over multiple hops (e.g. P2P-DROs). The Origin MUST use the chosen Security Configuration to secure all the control messages (DIOs and P2P-DRO-ACKs) it generates.

A router MUST NOT join the temporary DAG being created for a P2P-RPL route discovery if:

- o it receives both secure and unsecure DIOs or Secure DIOs with different Security Configurations pertaining to this route

discovery (i.e., referring to the same RPLInstanceID and DODAGID combination) prior to joining; or

- o it can not use the Security Configuration found in the Secure DIOs pertaining to this route discovery.

When a router (an Intermediate Router or a Target) joins a temporary DAG being created using Secure DIOs, it MUST remember the common Security Configuration used in the received Secured DIOs and MUST use this configuration to secure all the control messages (DIOs and P2P-DROs) it generates.

If an Intermediate Router (or a Target) encounters a control message (a DIO or a P2P-DRO or a P2P-DRO-ACK) pertaining to this route discovery that is either not secure or does not follow the Security Configuration the router remembers for this route discovery, the router MUST enter the "lock down" mode for the remainder of its stay in this temporary DAG. An Intermediate Router (or a Target) in the "lock down" mode MUST NOT generate or process any control message (irrespective of the Security Configuration used) pertaining to this route discovery. If the Origin receives a control message (a P2P-DRO) that does not follow the Security Configuration the Origin has chosen for this route discovery, it MUST discard the received message with no further processing.

12. Packet Forwarding Along a Route Discovered Using P2P-RPL

An Origin uses the Source Routing Header (SRH) [RFC6554] to send a packet along a Source Route discovered using P2P-RPL.

Travel along a Hop-by-hop Route, established using P2P-RPL, requires specifying the RPLInstanceID and the DODAGID (of the temporary DAG used for the route discovery) to identify the route. This is because a P2P-RPL route discovery does not use globally unique RPLInstanceID values and hence both the RPLInstanceID (a local value assigned by the Origin) and the DODAGID (an IPv6 address of the Origin) are required to uniquely identify a P2P-RPL Hop-by-hop Route to a particular destination.

An Origin includes an RPL option [RFC6553] inside the IPv6 hop-by-hop options header of a packet to send it along a Hop-by-hop Route established using P2P-RPL. For this purpose, the Origin MUST set the DODAGID of the temporary DAG used for the route discovery as the source IPv6 address of the packet. Further, the Origin MUST specify inside the RPL option the RPLInstanceID of the temporary DAG used for the route discovery and set the O flag inside the RPL option to one. On receiving this packet, an Intermediate Router checks the O flag

and correctly infer the source IPv6 address of the packet as the DODAGID of the Hop-by-hop Route. The router then uses the DODAGID, the RPLInstanceID and the destination address to identify the routing state to be used to forward the packet further.

13. Interoperability with Core RPL

This section describes how RPL routers that implement P2P-RPL interact with RPL routers that do not. In general, P2P-RPL operation does not affect core RPL operation and vice versa. However, core RPL does allow a router to join a DAG as a leaf node even if it does not understand the Mode of Operation (MOP) used in the DAG. Thus, an RPL router that does not implement P2P-RPL may conceivably join a temporary DAG being created for a P2P-RPL route discovery as a leaf node and maintain its membership even though the DAG no longer exists. This may impose a drain on the router's memory. However, such RPL-only leaf nodes do not interfere with P2P-RPL route discovery since a leaf node may only generate a DIO advertising an INFINITE_RANK and all routers implementing P2P-RPL are required to discard such DIOs. Note that core RPL does not require a router to join a DAG whose MOP it does not understand. Moreover, RPL routers in a particular deployment may have strict restrictions on the DAGs they may join, thereby mitigating the problem.

The P2P-RPL mechanism described in this document works best when all the RPL routers in the LLN implement P2P-RPL. In general, the ability to discover routes as well as the quality of discovered routes would deteriorate with the fraction of RPL routers that implement P2P-RPL.

14. Security Considerations

In general, the security considerations for the operation of P2P-RPL are similar to the ones for the operation of RPL (as described in Section 19 of [RFC6550]). Sections 6.1 and 10 of RPL specification [RFC6550] describe RPL's security framework that provides data confidentiality, authentication, replay protection and delay protection services. This security framework can also be used in P2P-RPL after taking in account the constraints specified in Section 11. P2P-RPL requires all routers participating in a secure route discovery to use the Security Configuration decided by the Origin. The intention is to avoid compromising the overall security of a route discovery due to some routers using a weaker Security Configuration. With "lock down" mechanism, described in Section 11, in effect, it is unlikely that an Origin would accept a route discovered under a Security Configuration other than the one it

intended. Any attempt to use a different Security Configuration (than the one the Origin intended) is likely to result, in the worst case, in the failure of the route discovery process. In the best case scenario, any such attempt by a rogue router would result in its neighbors entering the "lock down" mode and acting as firewalls to allow the route discovery to proceed in the remaining network.

RPL specification describes three modes of security: unsecured, pre-installed and authenticated. In the unsecured mode, secure control messages are not used and the only available security is the one provided by the link layer protocols. In the pre-installed mode, all the nodes use a pre-installed group key to join a secure DAG as the "routers" or "hosts", where the term "router" means a node that is capable of forwarding packets received from its parents or children in the DAG and the term "host" refers to nodes that can not function as "routers". In the authenticated mode, the nodes can join a secure DAG as "hosts" using the pre-installed key but then need to authenticate themselves to a key server to obtain the key that will allow them to work as "routers". The temporary DAG created for a P2P-RPL discovery can not be used for routing packets. Hence, it is not meaningful to say that a node joins this DAG as a "router" or a "host" in the sense defined above. Hence, in P2P-RPL, there is no distinction between the pre-installed and authenticated modes. A router can join a temporary DAG created for a secure P2P-RPL route discovery only if it can support the Security Configuration in use, which also specifies the key in use. It does not matter whether the key is pre-installed or dynamically acquired. The router must have the key in use before it can join the DAG being created for a secure P2P-RPL route discovery.

If a rogue router can support the Security Configuration in use (in particular, it knows the key in use), it can join the secure P2P-RPL route discovery and cause a variety of damage. Such a rogue router could advertise false information in its DIOs in order to include itself in the discovered route(s). It could generate bogus P2P-DRO messages carrying bad routes or maliciously modify genuine P2P-DRO messages it receives. A rogue router acting as the Origin could launch denial of service attacks against the LLN deployment by initiating fake P2P-RPL route discoveries. Here, RPL's authenticated mode operation would be useful, where a node can obtain the key to use for a P2P-RPL route discovery only after proper authentication.

Since a P2P-DRO message travels along a Source Route specified inside the message, some of the security concerns that led to the deprecation of Type 0 routing header [RFC5095] may apply. To avoid the possibility of a P2P-DRO message traveling in a routing loop, this document requires each Intermediate Router to confirm that the Source Route listed inside the message does not contain any routing

loop involving itself before the router could forward the message further. As specified in Section 9.6, this check involves the router making sure that its IPv6 addresses do not appear multiple times inside the Source Route with one or more other IPv6 addresses in between.

15. IANA Considerations

15.1. Additions to Mode of Operation

This document defines a new Mode of Operation, entitled "P2P Route Discovery Mode" (see Section 6), assigned a value TBD1 from the "Mode of Operation" space [to be removed upon publication: <http://www.iana.org/assignments/rpl/rpl.xml#mop>] [RFC6550]. IANA is requested to allocate a suitable value to TBD1. The string TBD1 in this document should be replaced by the allocated value. The previous two sentences should be removed before publication.

Value	Description	Reference
TBD1	P2P Route Discovery Mode of Operation	This document

Mode of Operation

15.2. Additions to RPL Control Message Options

This document defines a new RPL option: "P2P Route Discovery Option" (see Section 7), assigned a value TBD2 from the "RPL Control Message Options" space [to be removed upon publication: <http://www.iana.org/assignments/rpl/rpl.xml#control-message-options>] [RFC6550]. IANA is requested to allocate a suitable value to TBD2. The string TBD2 in this document should be replaced by the allocated value. The previous two sentences should be removed before publication.

Value	Meaning	Reference
TBD2	P2P Route Discovery	This document

RPL Control Message Options

15.3. Additions to RPL Control Codes

This document defines the following new RPL messages:

- o "P2P Discovery Reply Object" (see Section 8), assigned a value TBD3 from the "RPL Control Codes" space [to be removed upon publication:
<http://www.iana.org/assignments/rpl/rpl.xml#control-codes>]
[RFC6550]. IANA is requested to allocate TBD3 from the range 0x00-0x7F to indicate a message without security enabled. The string TBD3 in this document should be replaced by the allocated value. The previous two sentences should be removed before publication.
- o "Secure P2P Discovery Reply Object" (see Section 8.1), assigned a value TBD4 from the "RPL Control Codes" space [to be removed upon publication:
<http://www.iana.org/assignments/rpl/rpl.xml#control-codes>]
[RFC6550]. IANA is requested to allocate TBD4 from the range 0x80-0xFF to indicate a message with security enabled. The string TBD4 in this document should be replaced by the allocated value. The previous two sentences should be removed before publication.
- o "P2P Discovery Reply Object Acknowledgement" (see Section 10), assigned a value TBD5 from the "RPL Control Codes" space [to be removed upon publication:
<http://www.iana.org/assignments/rpl/rpl.xml#control-codes>]
[RFC6550]. IANA is requested to allocate TBD5 from the range 0x00-0x7F to indicate a message without security enabled. The string TBD5 in this document should be replaced by the allocated value. The previous two sentences should be removed before publication.
- o "Secure P2P Discovery Reply Object Acknowledgement" (see Section 10), assigned a value TBD6 from the "RPL Control Codes" space [to be removed upon publication:
<http://www.iana.org/assignments/rpl/rpl.xml#control-codes>]
[RFC6550]. IANA is requested to allocate TBD6 from the range 0x80-0xFF to indicate a message with security enabled. The string TBD6 in this document should be replaced by the allocated value. The previous two sentences should be removed before publication.

Code	Description	Reference
TBD3	P2P Discovery Reply Object	This document
TBD4	Secure P2P Discovery Reply Object	This document
TBD5	P2P Discovery Reply Object Acknowledgement	This document
TBD6	Secure P2P Discovery Reply Object Acknowledgement	This document

RPL Control Codes

16. Acknowledgements

Authors gratefully acknowledge the contributions of the following individuals (in alphabetical order) in the development of this document: Dominique Barthel, Jakob Buron, Cedric Chauvenet, Thomas Clausen, Robert Cragie, Ralph Droms, Adrian Farrel, Stephen Farrell, Brian Haberman, Ted Humpal, Richard Kelsey, Phil Levis, Charles Perkins, Joseph Reddy, Michael Richardson, Zach Shelby, Martin Stiemerling, Pascal Thubert, Hristo Valev and JP Vasseur.

17. References

17.1. Normative References

- [I-D.ietf-roll-terminology]
Vasseur, J., "Terminology in Low power And Lossy Networks", draft-ietf-roll-terminology-10 (work in progress), January 2013.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC6206] Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko, "The Trickle Algorithm", RFC 6206, March 2011.
- [RFC6550] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R.

Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, March 2012.

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

17.2. Informative References

- [I-D.ietf-roll-p2p-measurement]
Goyal, M., Baccelli, E., Brandt, A., and J. Martocci, "A Mechanism to Measure the Routing Metrics along a Point-to-point Route in a Low Power and Lossy Network", draft-ietf-roll-p2p-measurement-09 (work in progress), February 2013.
- [RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, December 2007.
- [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.
- [RFC6552] Thubert, P., "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.

Authors' Addresses

Mukul Goyal (editor)
University of Wisconsin Milwaukee
3200 N Cramer St
Milwaukee, WI 53201
USA

Phone: +1 414 2295001
Email: mukul@uwm.edu

Emmanuel Baccelli
INRIA

Phone: +33-169-335-511
Email: Emmanuel.Baccelli@inria.fr
URI: <http://www.emmanuelbaccelli.org/>

Matthias Philipp
INRIA

Phone: +33-169-335-511
Email: Matthias.Philipp@inria.fr

Anders Brandt
Sigma Designs
Emdrupvej 26A, 1.
Copenhagen, Dk-2100
Denmark

Phone: +45-29609501
Email: abr@sdesigns.dk

Jerald Martocci
Johnson Controls
507 E Michigan St
Milwaukee, WI 53202
USA

Phone: +1 414-524-4010
Email: jerald.p.martocci@jci.com

ROLL
Internet-Draft
Intended status: Standards Track
Expires: December 26, 2011

D. Popa
J. Jetcheva
Itron
N. Dejean
Elster
R. Salazar
Landis+Gyr
J. Hui
Cisco
June 24, 2011

Applicability Statement for the Routing Protocol for Low Power and Lossy
Networks (RPL) in AMI Networks
draft-poparollapplicabilityami00

Abstract

This document discusses the applicability of RPL in Advanced Metering Infrastructure (AMI) networks.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 26, 2011.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Electric Metering	3
1.2. Gas and Water Metering	4
1.3. Routing Protocol for LLNs (RPL)	4
1.4. Requirements Language	5
2. Deployment Scenarios	5
2.1. Network Topology	5
2.2. Traffic Characteristics	6
2.2.1. Meter Data Management	6
2.2.2. Distribution Automation	7
2.2.3. Emerging Applications	7
3. Using RPL to Meet Functional Requirements	7
4. RPL Profile	8
4.1. RPL Features	8
4.1.1. Storing vs. Non-Storing Mode	8
4.1.2. DAO Policy	8
4.1.3. Path Metrics	8
4.1.4. Objective Function	9
4.1.5. DODAG Repair	9
4.1.6. Security	10
4.2. RPL Options	10
4.3. Recommended Configuration Defaults and Ranges	10
5. Other Related Protocols	10
6. IANA Considerations	11
7. Security Considerations	11
8. Acknowledgements	11
9. References	11
9.1. Informative References	11
9.2. Normative References	12
Authors' Addresses	12

1. Introduction

Advanced Metering Infrastructure (AMI) systems measure, collect, and analyze energy consumption information. An AMI system enables two-way communication with electricity, water, gas, and/or heat meters. The communication may be scheduled, on exception, or on-demand.

AMI networks are composed of millions of endpoints, including meters, distribution automation elements, and home area network devices, typically inter-connected using some combination of wireless technologies and power-line communications, along with a wired or wireless backhaul network providing connectivity to "command-and-control" management software applications at the utility company back office.

1.1. Electric Metering

In many deployments, in addition to measuring energy consumption, the electric meter network plays a central role in the Smart Grid since it enables the utility company to control and query the electric meters themselves and also since it can serve as a backhaul for all other devices in the Smart Grid, including water and gas meters, distribution automation and home area network devices. Electric meters may also be used as sensors to monitor electric grid quality and support applications such as Electric Vehicle charging.

Electric meter networks are composed of millions of smart meters (or nodes), each of which is resource constrained in terms of processing power, storage capabilities, and communication bandwidth, due to a combination of factors including Federal Communications Commission (FCC) or other continents' regulations on spectrum use, American National Standards Institute (ANSI) standards or other continents' regulation on meter behavior and performance, on heat emissions within the meter, form factor and cost considerations. This results in a compromise between range and throughput, with effective link throughput of tens to a few hundred kilobits per second per link, a potentially significant portion of which is taken up by protocol and encryption overhead when strong security measures are in place.

Electric meters are often interconnected into multi-hop mesh networks, each of which is connected to a backhaul network leading to the utility network through a network aggregation point (NAP) node. These kinds of networks increase coverage and reduce installation cost, time and complexity, as well as operational costs, as compared to single-hop wireless networks relying on a wired or cellular backhaul. Each electric meter mesh typically has in the order of several thousand wireless endpoints, with densities varying based on the area and the terrain, with apartment buildings in urban centers

having possibly hundreds of meters in close proximity, and rural areas having sparse node distributions, including nodes that only have one or two network neighbors. Mesh deployments can exhibit tens of hops between a network device and the nearest aggregation point.

1.2. Gas and Water Metering

While electric meters can typically consume electricity from the same electric feed that they are monitoring, gas and water meters typically run on a modest source of stored energy (i.e. batteries). In certain scenarios, gas and water meters are integrated with electric meters in the same AMI network. In this scenario, gas and water meters typically do not route messages or operate as hosts to prolong their lifetime.

In other scenarios, however, gas and water meters do not have the luxury of communicating with a powered routing infrastructure. Instead, they must communicate through other battery powered devices (i.e. through other gas and water meters) to reach a NAP. Alternative scenarios also include water and/or gas meters communicating directly to a sparsely deployed network infrastructure, requiring increased transmit power levels for increased range that significantly impacts energy consumption and battery lifetime. For such networks, the routing protocol must configure routes with energy consumption in mind. The NAPs, however, are typically mains powered as in AMI networks with electric meters.

RPL is designed to operate in energy-constrained environments and includes energy-saving mechanisms (e.g. Trickle timers) and energy-aware metrics. By supporting a number of different metrics and constraints, RPL is also designed to support networks composed of nodes that have vastly different characteristics [I-D.ietf-roll-routing-metrics].

1.3. Routing Protocol for LLNs (RPL)

RPL provides routing functionality for mesh networks composed of a large number of resource-constrained devices interconnected by low power and lossy links. Constrained devices within the same network typically communicate through a common aggregation point (e.g., a border router). RPL builds a Directed Acyclic Graph (DAG) routing structure rooted at the aggregation point. It ensures loop-free routing, support for alternate routes, and a wide range of routing metrics and policies.

This note describes the applicability of RPL defined in [I-D.ietf-roll-rpl] to AMI deployments. RPL was designed to meet the following application requirements:

- o Routing Requirements for Urban Low-Power and Lossy Networks [RFC5548].
- o Industrial Routing Requirements in Low-Power and Lossy Networks [RFC5673].
- o Home Automation Routing Requirements in Low-Power and Lossy Networks [RFC5826].
- o Building Automation Routing Requirements in Low-Power and Lossy Networks [RFC5867].

The Routing Requirements for Urban Low-Power and Lossy Networks is most applicable to AMI networks.

The terminology used in this document is defined in [I-D.ietf-roll-terminology].

1.4. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Deployment Scenarios

2.1. Network Topology

AMI networks are composed of millions of endpoints distributed across both urban and rural environments. Such endpoints include electric, gas, and water meters; distribution automation elements; and in-home devices. Devices in the network communicate directly with other devices in close proximity using a variety of low-power and/or lossy link technologies that are both wired and wireless (e.g. IEEE 802.15.4, IEEE P1901.2, and WiFi). Network elements may not only source and sink packets, but must also forward packets to reduce the need for dedicated routers and associated deployment costs.

In a typical AMI deployment, groups of meters within physical proximity form routing domains. The size of each group in a typical AMI deployment can be from 1000 to 10000 or 15000 meters

Powered from the main line electric meters have less energy constraints than battery powered devices and can afford the additional resources required for routing packets. In mixed environments, electric meters provide the routing topology while gas and water meters operate as leaves. However, in networks that cannot

afford a powered infrastructure, gas and water meters must either talk directly to a network infrastructure or form their own routing topology, albeit with energy consumption in mind.

Each meter routing domain is connected to a larger IP infrastructure through one or more LLN Border Routers (LBRs). The LBRs provide Wide Area Network (WAN) connectivity through more traditional links (e.g. Ethernet, Cellular, Private WAN) or other wireless technologies.

The meter networks may also serve as transit networks for other devices, including battery powered gas and water meters, distribution automation elements (i.e. distribution sensors and actuators), and in-home devices. These other devices may utilize a different link-layer technology than the one used in the metering network.

2.2. Traffic Characteristics

2.2.1. Meter Data Management

Meter Data Management (MDM) applications typically require every smart meter to communicate with a few head-end servers deployed in a utility data center. As a result, all smart metering traffic typically flows through the LBRs. In general, the vast majority of traffic flows from smart meter devices to the head-end servers with limited traffic flowing from head-end servers to smart meter devices. In RPL terminology, this traffic flow is also referred to as Multipoint-to-point Traffic (MP2P).

Smart meters may generate traffic according to a schedule (e.g. meter read reporting), in response to on-demand queries (e.g. on-demand meter read), or in response to events (e.g. power outages or leak detections). Such traffic is typically unicast since it is sent to a single head-end server.

Head-end servers may generate traffic to configure smart metering devices or initiate queries. Head-end servers generate both unicast and multicast traffic to efficiently communicate with a single device or groups of devices. In RPL terminology, this traffic flow is also referred to as Point-to-Multipoint Traffic (P2MP). The head-end server may send a single small packet at a time (e.g. a meter read request or small configuration change) or many large packets in sequence (e.g. a firmware upgrade across one or thousands of devices).

While smart metering applications typically do not have hard real-time constraints, they are often subject to stringent latency and reliability service level agreements. Some applications also have stringent latency requirements to function properly.

2.2.2. Distribution Automation

Distribution Automation (DA) applications typically involve a small number of devices that communicate with each other in a Point-to-Point (P2P) fashion. The DA devices may or may not be in close physical proximity.

DA applications typically have more stringent latency requirements than MDM applications.

2.2.3. Emerging Applications

There are a number of emerging applications (e.g. Electric Vehicle charging) that may involve P2P communication as well. These applications may eventually have more stringent latency requirements than MDM applications.

3. Using RPL to Meet Functional Requirements

The functional requirements for most AMI deployments are similar to those listed in [RFC5548].

- o The routing protocol MUST be capable of supporting the organization of a large number of nodes into regions containing on the order of 10^2 to 10^4 nodes each.
- o The routing protocol MUST provide mechanisms to support configuration of the routing protocol itself.
- o The routing protocol SHOULD support and utilize the large number of highly direct flows to a few head-end servers to handle scalability.
- o The routing protocol MUST dynamically compute and select effective routes composed of low-power and lossy links. Local network dynamics SHOULD NOT impact the entire network. The routing protocol MUST compute multiple paths when possible.
- o The routing protocol MUST support multicast and anycast addressing. The routing protocol SHOULD support formation and identification of groups of field devices in the network.

RPL efficiently supports scalability and highly directed traffic flows between every smart meter and the few head-end servers by building a Directed Acyclic Graph (DAG) rooted at each LBR.

RPL supports zero-touch configuration by providing in-band methods

for configuring RPL variables using DIO messages.

RPL supports time-varying link qualities by allowing the use of metrics that effectively characterize the quality of a path (e.g. Estimated Transmission Count (ETX)). RPL limits the impact of changing local conditions by discovering and maintaining multiple DAG parents and providing a local repair mechanism when all parents have been dropped.

4. RPL Profile

This section outlines a RPL profile for most representative AMI deployments.

4.1. RPL Features

4.1.1. Storing vs. Non-Storing Mode

In most scenarios, electric meters can utilize the power they are monitoring for their own processing and computation and are not as constrained in energy consumption. Instead, the capabilities of an electric meter are primarily constrained by cost. As a result, different AMI deployments can vary significantly in terms of the memory, computational, and communication trade-offs that were made for their devices. For this reason, the use of RPL storing or non-storing mode SHOULD be deployment specific.

When meters are memory constrained and cannot adequately store route tables to support downward routing, non-storing mode is preferred. However, when nodes are capable of adequately storing such routing tables, storing mode can lead to shorter paths and reduce channel utilization near the root.

4.1.2. DAO Policy

Two-way communication is required in AMI systems. As a result, electric meters SHOULD send DAO messages to establish downward paths back to themselves.

4.1.3. Path Metrics

Smart metering deployments utilize link technologies that can exhibit significant packet loss. To characterize a path over such link technologies, AMI deployments can use the Expected Transmission Count (ETX) metric as defined in [I-D.ietf-roll-routing-metrics].

For water- and gas-only networks that cannot rely on a powered

infrastructure, energy constraints may require simpler metrics that do not require as much energy to compute. In particular, Hop Count and Link Quality Level may be more suitable in such deployments. Other metrics may be vendor-specific or defined at a later time into companion RFCs.

4.1.4. Objective Function

RPL relies on an Objective Function for selecting parents and computing path costs and rank. This objective function is decoupled from the core RPL mechanisms but also from the metrics in use in the network. Two objective functions for RPL have been defined:

- o OF0 which does not deal with any metric,
- o MRHOF which deals with a single metric.

Both of them define the selection of a preferred parent and backup parents. Note that these Objective Functions do not support multiple metrics that might be required in heterogeneous networks (i.e. networks composed of devices with varying energy constraints). While RPL provides the flexibility to support additional metrics, a new Objective Function MAY be specified to properly handle additional metrics.

4.1.5. DODAG Repair

To effectively handle time-varying link characteristics, AMI deployments SHOULD utilize the local repair mechanisms in RPL.

The first mechanism for local repair when a node loses its parents is to detach from a DODAG then re-attach to the same or different DODAG at a later time. While detached, a node advertises an infinite rank value so that its children can select a different parent. This process is known as poisoning and described in Section 8.2.2.5 of [I-D.ietf-roll-rpl]. While RPL provides an option to form a local DODAG, doing so in AMI deployments is of little benefit since AMI applications typically communicate through a LBR. After the detached node has made sufficient effort to send notification to its children that it is detached, the node can rejoin the same DODAG with a higher rank value. Note that when joining a different DODAG, the node need not perform poisoning.

The second mechanism is a limit on how much a node can increase its rank within a given DODAG Version. Setting the DAGMaxRankIncrease to a non-zero value enables this local repair mechanism. Setting DAGMaxRankIncrease to a value less than infinity limits the cost of count-to-infinity scenarios when they occur.

The third mechanism is loop detection, enabled by including the rank value of a node in packets forwarded towards the root in RPL Packet Information [I-D.ietf-6man-rpl-option]. Note that loop detection is not needed when sending packets using strict source routing.

4.1.6. Security

AMI deployments operate in areas that do not provide any physical security. For this reason, the link technologies used within AMI deployments typically provide security mechanisms to ensure confidentiality, integrity, and freshness. As a result, AMI deployments may not need to implement RPL's security mechanisms and could rely on link layer security features.

4.2. RPL Options

4.3. Recommended Configuration Defaults and Ranges

- o AMI deployments can involve densities of hundreds of devices within communication range. As a result, such networks SHOULD set the DIOIntervalMin to 16 or more, giving a Trickle Imin of 1 minute or more. For low-energy consumption operations, such networks SHOULD set DIOIntervalMin to a higher value.
- o AMI deployments SHOULD set DIOIntervalDoublings to a value that gives a Trickle Imax of 2 hours or more. For low-energy consumption operations, such networks SHOULD set DIOIntervalDoublings to a value that gives a Trickle Imax of e.g. 2 days.
- o AMI deployments SHOULD set DIORedundancyConstant to a value of 10 or more.
- o AMI deployments SHOULD set MinHopRankIncrease to 256, giving 8 bits of resolution (e.g. for the ETX metric).
- o To enable local repair, AMI deployments SHOULD set MaxRankIncrease to a value that allows a device to move a small number of hops away from the root. With a MinHopRankIncrease of 256, a MaxRankIncrease of 1024 would allow a device to move up to 4 hops away.

5. Other Related Protocols

This document contains no other related protocols.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

This memo includes no security considerations.

8. Acknowledgements

The authors would like to acknowledge the review, feedback, and comments from Dominique Barthel.

9. References

9.1. Informative References

[I-D.ietf-6man-rpl-option]

Hui, J. and J. Vasseur, "RPL Option for Carrying RPL Information in Data-Plane Datagrams", draft-ietf-6man-rpl-option-03 (work in progress), March 2011.

[I-D.ietf-roll-routing-metrics]

Vasseur, J., Kim, M., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics used for Path Calculation in Low Power and Lossy Networks", draft-ietf-roll-routing-metrics-19 (work in progress), March 2011.

[I-D.ietf-roll-rpl]

Winter, T., Thubert, P., Brandt, A., Clausen, T., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., and J. Vasseur, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks", draft-ietf-roll-rpl-19 (work in progress), March 2011.

[I-D.ietf-roll-terminology]

Vasseur, J., "Terminology in Low power And Lossy Networks", draft-ietf-roll-terminology-05 (work in progress), March 2011.

[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.
- [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.

9.2. Normative References

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

Authors' Addresses

Daniel Popa
Itron

Email: daniel.popa@itron.com

Jorjeta Jetcheva
Itron
2111 N Molter Rd.
Liberty Lake, WA
USA

Phone: +408 688 1428
Email: jorjeta.jetcheva@itron.com

Nicolas Dejean
Elster

Email: nicolas.dejean@coronis.com

Ruben Salazar
Landis+Gyr

Email: ruben.salazar@landisgyr.com

Jonathan W. Hui
Cisco
170 West Tasman Drive
San Jose, California 95134
USA

Phone: +408 424 1547
Email: jonhui@cisco.com

