

Internet Engineering Task Force  
Internet-Draft  
Intended status: Experimental  
Expires: July 22, 2012

M. Goyal, Ed.  
University of Wisconsin  
Milwaukee  
E. Baccelli  
M. Philipp  
INRIA  
A. Brandt  
Sigma Designs  
J. Martocci  
Johnson Controls  
January 19, 2012

Reactive Discovery of Point-to-Point Routes in Low Power and Lossy  
Networks  
draft-ietf-roll-p2p-rpl-06

## 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 and establish, on demand, a route to another IPv6 router in the LLN such that the discovered route meets specified metrics constraints, without necessarily going along the DAG links established by core RPL.

## 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 July 22, 2012.

## Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

Internet-Draft

[draft-ietf-roll-p2p-rpl-06](#)

January 2012

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

<a href="#">1.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">2.</a>	The Use Cases . . . . .	<a href="#">3</a>
<a href="#">3.</a>	Terminology . . . . .	<a href="#">4</a>
<a href="#">4.</a>	Applicability . . . . .	<a href="#">5</a>
<a href="#">5.</a>	Functional Overview . . . . .	<a href="#">5</a>
<a href="#">6.</a>	P2P Route Discovery Mode Of Operation . . . . .	<a href="#">7</a>
<a href="#">6.1.</a>	Setting a P2P Mode DIO . . . . .	<a href="#">8</a>
<a href="#">7.</a>	P2P Route Discovery Option (P2P-RDO) . . . . .	<a href="#">10</a>
<a href="#">8.</a>	The Discovery Reply Object (DRO) . . . . .	<a href="#">13</a>
<a href="#">8.1.</a>	Secure DRO . . . . .	<a href="#">14</a>
<a href="#">8.2.</a>	Setting a P2P-RDO Carried in a Discovery Reply Object . . . . .	<a href="#">14</a>
<a href="#">9.</a>	P2P-RPL Route Discovery By Creating a Temporary DAG . . . . .	<a href="#">15</a>
<a href="#">9.1.</a>	Joining a Temporary DAG . . . . .	<a href="#">15</a>
<a href="#">9.2.</a>	Trickle Operation For P2P Mode DIOs . . . . .	<a href="#">15</a>
<a href="#">9.3.</a>	Processing a P2P Mode DIO . . . . .	<a href="#">17</a>
9.4.	Additional Processing of a P2P Mode DIO At An Intermediate Router . . . . .	<a href="#">18</a>
<a href="#">9.5.</a>	Additional Processing of a P2P Mode DIO At The Target . . . . .	<a href="#">18</a>
<a href="#">9.6.</a>	Processing a DRO At An Intermediate Router . . . . .	<a href="#">19</a>
<a href="#">9.7.</a>	Processing a DRO At The Origin . . . . .	<a href="#">20</a>
<a href="#">10.</a>	The Discovery Reply Object Acknowledgement (DRO-ACK) . . . . .	<a href="#">21</a>
<a href="#">11.</a>	Packet Forwarding Along a P2P-RPL Route . . . . .	<a href="#">22</a>
<a href="#">12.</a>	Constants . . . . .	<a href="#">23</a>
<a href="#">13.</a>	Interoperability With Core RPL . . . . .	<a href="#">23</a>
<a href="#">14.</a>	Security Considerations . . . . .	<a href="#">23</a>
<a href="#">15.</a>	IANA Considerations . . . . .	<a href="#">24</a>
<a href="#">15.1.</a>	Additions to DIO Mode of Operation . . . . .	<a href="#">24</a>
<a href="#">15.2.</a>	Additions to RPL Control Message Options . . . . .	<a href="#">24</a>
<a href="#">15.3.</a>	Additions to RPL Control Codes . . . . .	<a href="#">25</a>
<a href="#">16.</a>	Acknowledgements . . . . .	<a href="#">25</a>

<a href="#">17. References</a>	<a href="#">25</a>
<a href="#">17.1. Normative References</a>	<a href="#">25</a>
<a href="#">17.2. Informative References</a>	<a href="#">26</a>
Authors' Addresses	<a href="#">26</a>

## [1. Introduction](#)

Targeting Low power and Lossy Networks (LLNs), the RPL routing protocol [[I-D.ietf-roll-rpl](#)] provides paths along a Directed Acyclic Graph (DAG) rooted at a single router in the network. Establishment and maintenance of the DAG is performed by each router in the LLN using specific link-local multicast signaling (DIO messages).

When two arbitrary routers (neither of which is the DAG's root) need to communicate, core RPL provides dog-legged paths along DAG links, which may not be efficient enough for several Home and Building Automation applications [[RFC5826](#)][RFC5867], for the following reasons:

- o The need to preprovision routes: each potential destination in the network must declare itself as such, via specific additional signaling (DAO messages).
- o The need to route along DAG links: depending on the network topology and metrics in use, the constraint to route along a DAG may cause significantly suboptimal P2P routes and severe traffic congestion near the DAG root.

This document thus describes a mechanism, complementary to the core RPL functionality, that enables a router to discover on-demand a route to another arbitrary router in the LLN, such that the discovered route meets specified metrics constraints, without necessarily going along an existing DAG. This reactive P2P route discovery mechanism is henceforth referred to as P2P-RPL. P2P-RPL allows for the discovery of source routes as well as hop-by-hop routes. Discovered routes may not be the best available but are guaranteed to satisfy the desired constraints in terms of the routing metrics and are thus considered "good enough" from the application's perspective.

A complementary functionality that helps decide whether or not to

initiate a P2P route discovery, is a mechanism to measure the end-to-end cost of an existing route. [Section 4](#) provides further details on how such functionality, specified in [[I-D.ietf-roll-p2p-measurement](#)], is used to determine the value of metric constraints for the route discovery using P2P-RPL.

## [2.](#) The Use Cases

P2P-RPL is intended to be employed as complementary to RPL in specific scenarios that need P2P paths between arbitrary routers.

One use case, common in a home environment, involves a remote control (or a motion sensor) that suddenly needs to communicate with a lamp module, whose network address is a-priori known. In this case, the source of data (the remote control or the motion sensor) must be able to discover a route to the destination (the lamp module) "on demand".

Another use case, common in a large 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, and thus routes across this DAG are desirable.

The use cases also include scenarios where energy or latency constraints are not satisfied by the routes provided by core RPL along a 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 [[I-D.ietf-roll-terminology](#)] and [[I-D.ietf-roll-rpl](#)]. This document introduces the following terms:

Origin : The RPL router initiating the P2P route discovery.

Target : The RPL router at the other end point of the P2P route(s) to be discovered.

Intermediate Router: An RPL router that is neither the origin nor the target.

Forward Route: A route in the forward direction, i.e., from the origin to the target.

Backward Route: A route in the backward direction, i.e., from the target to the origin.

Bidirectional Route: A route that can be used in both forward and backward directions.

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.

#### 4. Applicability

A route discovery using P2P-RPL may be performed by an origin when no route exists between itself and the target or when the existing routes do not satisfy the application requirements. P2P-RPL is designed to discover and establish one hop-by-hop route or discover one or more source routes such that the discovered route(s) meet the specified constraints. In some application contexts, the constraints that the discovered route(s) must satisfy are intrinsically known or can be specified by the application. For example, an origin that expects a target 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 an existing route to the target to determine the constraints. For example, an origin that measures the total ETX of its along-DAG route to the 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 the origin and the target is specified in

[\[I-D.ietf-roll-p2p-measurement\]](#). If there is no existing route between the origin and target or the cost measurement for the existing route fails, the origin will have to guess the constraints 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 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 routing metrics in use and the prevalent conditions in the network. A network designer may take into consideration both the benefits (potentially better routes; no need to maintain routes proactively) and costs (control messages generated during the route discovery process) when using P2P-RPL.

## 5. Functional Overview

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

As is the case with core RPL, P2P-RPL uses IPv6 link-local multicast DIO messages to establish a DAG (unlike core RPL, this DAG is temporary). Each router joining the DAG determines a rank for itself in the DAG and ignores the subsequent DIO messages received from lower (higher in numerical value) ranked neighbors. Thus, the DIO

messages propagate outward from the DAG root rather than return inward towards the DAG root. As in core RPL, DIO message generation at a router is further controlled by a Trickle timer that allows a router to avoid generating unnecessary messages [\[RFC6206\]](#). P2P-RPL also uses the routing metrics [\[I-D.ietf-roll-routing-metrics\]](#), objective functions and packet forwarding framework specified for core RPL.

In P2P-RPL, a route discovery takes place by forming a temporary DAG rooted at the origin. The DIOs, used to create the temporary DAG, are identified by a new Mode of Operation (P2P Route Discovery mode defined in [Section 6](#)) and carry the following information (in a P2P Route Discovery Option defined in [Section 7](#)):

- o An IPv6 address of the target.

- 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 in the forward direction.
- o The desired number of routes (if source routes are being discovered).
- o Whether the route(s) need to be bidirectional. If bidirectional route(s) are being discovered, the target may store the route in backward direction for use as a source route. This specification does not provide for the establishment of backward hop-by-hop routes.

The DIOs, listing the P2P Route Discovery mode as the Mode of Operation, are henceforth referred to as the P2P mode DIOs. The P2P mode DIOs MAY also carry the following information (in one or more Metric Container Options):

- o The relevant routing metrics
- o The constraints that the discovered route must satisfy. These constraints also limit how far the DIOs message may travel.

As the routers join the temporary DAG, they keep track of the best (partial) route(s) they have seen and advertise these routes, along with the corresponding routing metrics, in their P2P mode DIOs. The routing metrics are measured in forward direction unless bidirectional routes are being discovered, in which case the measurement of routing metrics need to take into account both forward and backward directions. A router, including the target, 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 the target receives a P2P mode DIO, it checks whether the route advertised therein satisfies the routing constraints. If yes, the target may select the route for further processing as described next.

This document does not specify a particular method for the target to select a route among the ones that satisfy the route constraints. Examples 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 routes are being discovered, the target sends the discovered source routes to the origin via Discovery Reply Object (DRO) messages (defined in [Section 8](#)) with one DRO message carrying one discovered route. On receiving a DRO message, the origin stores the route contained therein in its memory.

If a hop-by-hop route is being discovered, the target sends a DRO message to the origin after selecting a suitable route among the ones that satisfy the route constraints. The DRO message travels towards the origin along the discovered route, establishing state for this route in the routers on the path.

The target may store a discovered route in its memory if it is bidirectional and use it as a backward source-route to send packets to the origin.

The target may request the origin to acknowledge the receipt of a DRO message by sending back a DRO Acknowledgement (DRO-ACK) message (defined in [Section 10](#)). The origin unicasts a DRO-ACK message to the target. When the target does not receive the requested DRO-ACK within a certain time interval of sending a DRO, it resends the 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" bit in the DRO message to let the nodes in the LLN know about the completion of the route discovery process.

## [6.](#) P2P Route Discovery Mode Of Operation

This section specifies a new RPL Mode of Operation (MOP), P2P Route

confirmed by IANA). A DIO message, listing P2P mode as the MOP, is identified as performing reactive P2P route discovery by creating a temporary DAG. A P2P mode DIO MUST carry one P2P Route Discovery Option (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 [[I-D.ietf-roll-rpl](#)]. The origin MUST NOT use the same RPLInstanceID in two or more concurrent route discoveries. The origin MAY use the same RPLInstanceID value to establish hop-by-hop P2P-RPL routes to different target routers as long as these route discoveries are not concurrent.
- 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: MUST be cleared since this DAG is temporary in nature, is created solely for the purpose of P2P-RPL route discovery and MUST NOT be used for packet routing.
- o Mode of Operation (MOP): MUST be set to 4, corresponding to P2P Route Discovery mode.
- o DTSN: MUST be set to value zero on transmission and ignored on reception.
- o DODAGPreference (Prf): This field MUST be set to value 0 (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 [[I-D.ietf-roll-rpl](#)].

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

- o MaxRankIncrease: This field MUST be set to 0 to disable local repair of the temporary DAG.
- o Trickle parameters (DIOIntervalDoublings, DIOIntervalMin, DIORedundancyConstant) SHOULD be set as described in [Section 9.2](#).

- o The Default Lifetime and Lifetime Unit parameters in DODAG Configuration option indicate the life time of the state the routers maintain for a hop-by-hop route established using P2P-RPL and may be set as desired.
- o The other fields in the DODAG Configuration Option, including the OCP identifying the Objective function, can be set in the desired fashion as per the rules described in [[I-D.ietf-roll-rpl](#)].

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 DIORedundancyConstant: 1
- o MaxRankIncrease: 0
- o Default Lifetime: 0xFF
- o Lifetime Unit: 0xFFFF
- o Objective Code Point: 0, i.e., 0F0 [[I-D.ietf-roll-of0](#)] is the default objective function.
- o The remaining parameters have default values as specified in [[I-D.ietf-roll-rpl](#)].

The routing metrics and constraints [[I-D.ietf-roll-routing-metrics](#)] used in P2P-RPL route discovery are included in one or more Metric Container options [[I-D.ietf-roll-rpl](#)] inside the P2P mode DIO. Note that a DIO need not include a Metric Container if 0F0 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 Route Discovery Option (described in [Section 7](#)).

A P2P mode DIO:

- o MUST NOT carry any Route Information or Prefix Information Options (described in [[I-D.ietf-roll-rpl](#)]).
- o MUST carry one (and only one) P2P Route Discovery Option (described in [Section 7](#)).

A router MUST discard a received P2P mode DIO if it violates any of the rules listed above.

7. P2P Route Discovery Option (P2P-RD0)

This section specifies a new RPL option, P2P Route Discovery Option (P2P-RD0), one instance of which MUST be carried inside a P2P mode DIO message.

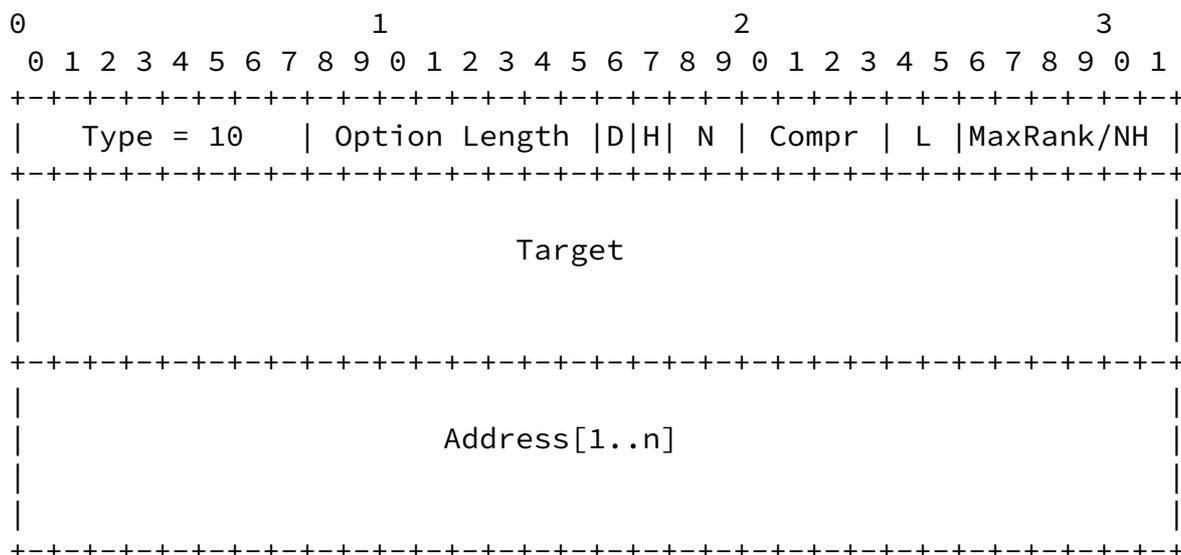


Figure 1: Format of P2P Route Discovery Option (P2P-RD0)

The format of a P2P-RD0 is illustrated in Figure 1. A P2P mode DIO and a DR0 (defined in [Section 8](#) message MUST carry one P2P-RD0. A P2P-RD0 consists of the following fields:

- o Option Type: 0x0A (to be confirmed by IANA).
- 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 Direction (D): This flag indicates the direction in which the desired routes should be optimized. The flag is set to 1 if the routes are to be optimized for use in both forward and backward directions. If the discovered routes need to be optimized in the

forward direction only, the flag is reset to 0. Note that the discovered routes should have bidirectional reachability irrespective of the value of the D flag. This is because DR0 messages travel from the target back to the origin along one of the discovered routes. The link-level metric objects contained in the P2P mode DIO SHOULD be measured in the direction indicated by the D flag.

- o Hop-by-hop (H): This flag is set to 1 if a hop-by-hop route is desired. The flag is reset to zero if source routes are desired. This specification allows for the establishment of one hop-by-hop route or up to four source routes in the forward direction. This specification does not allow for the establishment of hop-by-hop routes in the backward direction. If a bidirectional route is discovered, the target MAY use the route in backward direction as a source route to reach the origin, irrespective of the value of the H flag.
- o Number of Routes (N): When source routes are being discovered, the value in this field plus one indicates the desired number of routes. When a hop-by-hop route is being discovered this 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 0 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 suggested life time of the temporary DAG, i.e., the suggested duration a router joining the temporary DAG SHOULD maintain its membership in the DAG. The mapping between the values in this field and the life time of 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 DIOs to reach the target. Note that a router MAY detach from the temporary DAG sooner if it receives a DRO message concerning this DAG with "stop" bit set.

- o MaxRank/NH:
  - \* When a P2P Route Discovery Option 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 [[I-D.ietf-roll-rpl](#)]) 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. The 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 DRO message, this field indicates the index of the next hop address inside the Address vector.
- o Target: The IPv6 address of the target after eliding Compr number of prefix octets.
- o Address[1..n]: A vector of IPv6 addresses representing a (partial) route in the forward direction:
  - \* Each element in the Address vector has size (16 - Compr) octets and MUST contain a valid 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 Address vector is used to accumulate a route optimized in

the direction specified by the D flag.

- \* The IPv6 addresses in the Address vector MUST be accessible in both forward and backward directions. Accessibility in the backward direction is required because the DRO message uses the route accumulated in the Address vector to travel from the target 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 its address does not already exist in the vector. A router 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 Discovery Reply Object (DRO)

This section defines two new RPL Control Message types, the Discovery Reply Object (DRO), with code 0x04 (to be confirmed by IANA), and the Secure DRO, with code 0x84 (to be confirmed by IANA). A DRO serves one of the following functions:

- o Carry a discovered source route from the target to the origin;
- o Establish a hop-by-hop route as it travels from the target to the origin.

A DRO message MAY 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 DRO message MUST carry one P2P-RDO and travel from the target to the origin via link-local multicast along the route specified inside the Address vector in the P2P-RDO.

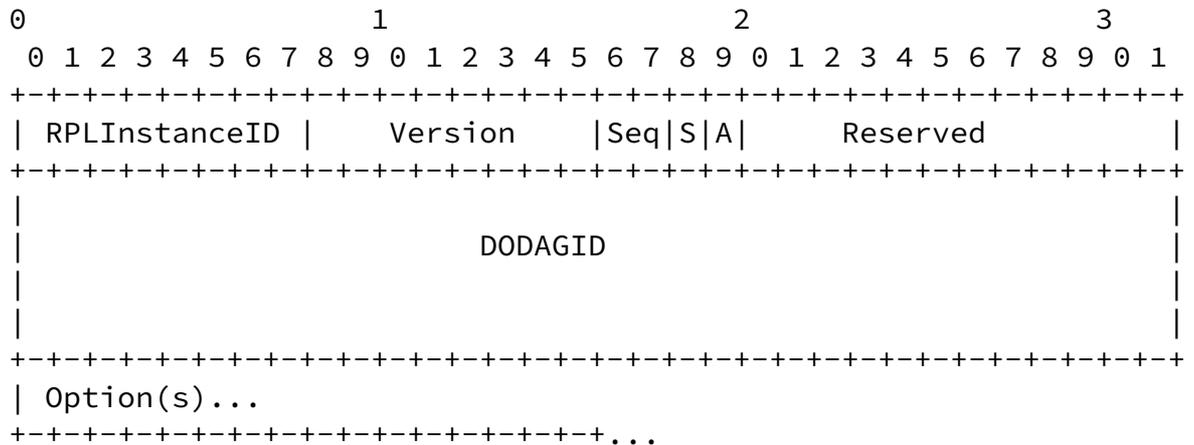


Figure 2: Format of the base Discovery Reply Object (DRO)

The format of the base Discovery Reply Object (DRO) is shown in Figure 2. A base 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 Sequence Number (Seq): This 2-bit field indicates the sequence number for the DRO. This field is relevant when the A flag (specified below) is set, i.e., the target requests an

acknowledgement from the origin for a received DRO. The origin includes the RPLInstanceID, the DODAGID and the Sequence Number of the received DRO inside the DRO-ACK message it sends back to the target.

- o Stop (S): This flag, when set by the target, indicates that the P2P-RPL route discovery is over. All the routers receiving such a DRO, including the ones not listed in the route carried inside P2P-RDO, SHOULD cancel any pending DIO transmissions for the temporary DAG created for the route discovery and MAY detach from this DAG immediately. Note that the stop flag serves to stop

further DIO transmissions for a P2P-RPL route discovery but it does not affect the processing of 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 DRO messages even if an earlier DRO message (with same RPLInstanceID, DODAGID and Version Number fields) had the stop flag set.

- o Ack Required (A): This flag, when set by the target, indicates that the origin SHOULD unicast a DRO-ACK message (defined in [Section 10](#)) to the target when it receives the DRO.
- 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 DRO message MUST carry one P2P-RDO that MUST specify a complete route between the target and the origin. The DRO message MAY carry a Metric Container Option that contains the aggregated routing metrics values for the route specified in P2P-RDO.

### [8.1.](#) Secure DRO

A Secure DRO message follows the format in Figure 7 of [\[I-D.ietf-roll-rpl\]](#), where the base format is the base DRO shown in Figure 2.

### [8.2.](#) Setting a P2P-RDO Carried in a Discovery Reply Object

A Discovery Reply Object MUST carry one P2P-RDO, which MUST be set as defined in [Section 7](#) with the following exceptions:

- o Direction (D): This flag MUST be set to zero on transmission and ignored on reception.
- 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 DRO message, the NH field is set to  $n = (\text{Option Length} - 2 - (16 - \text{Compr})) / (16 - \text{Compr})$ .
- 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 functioning of P2P-RPL route discovery by creating a temporary DAG, using the P2P mode DIO, DRO and DRO-ACK messages.

### [9.1.](#) Joining a Temporary DAG

All the routers participating in a P2P-RPL route discovery, including the origin and the target, MUST join the temporary DAG being created for the purpose. When a router joins a temporary DAG advertised by a P2P mode DIO, it SHOULD maintain its membership in the temporary DAG for the suggested Life Time duration listed in 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 packets. A router SHOULD detach from the temporary DAG once the duration of its membership in the DAG has exceeded the DAG's suggested life time. A router MAY detach from a temporary DAG sooner when it receives a DRO about the temporary DAG with the stop flag set.

### [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 wireless communication. 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 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.

- 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 environments with low or moderate packet loss rates. In environments with high packet loss rates, a higher value for the redundancy constant may be more suitable.

### [9.3.](#) Processing a P2P Mode DIO

The rules for DIO processing and transmission, described in [Section 8](#) of RPL [[I-D.ietf-roll-rpl](#)], apply to P2P mode DIOs as well except as modified in this document.

The following rules for processing a 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 originated the received DIO. This is to ensure that a discovered route can be used to send a DR0 message from the target to

the origin. 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 update the values of the link-level routing metrics included inside the DIO in the direction indicated by the D flag in the P2P-RD0. If the D flag is 0, i.e., the discovered routes need not be bidirectional, the link-level routing metrics SHOULD be measured in the forward direction, i.e., towards the node receiving the DIO. If the D flag is 1, i.e., bidirectional routes

are desired, the link-level routing metrics SHOULD be calculated so as to take into account the metric's value in both forward and backward directions.

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

- o If the DIO advertises INFINITE\_RANK as defined in [[I-D.ietf-roll-rpl](#)].
- 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 router cannot evaluate the mandatory route constraints listed in the DIO or if the routing metric values do not satisfy one or more of the mandatory constraints.

#### [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 if the router cannot elide Compr (as specified in the P2P-RD0) prefix octets from its IPv6 address.

On receiving a P2P mode DIO, an intermediate 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 [[I-D.ietf-roll-rpl](#)]. If the received DIO would allow the router to improve the route it

advertises, the router MUST store the route advertised in the DIO in memory (after adding its own IPv6 address to the route) for inclusion in its future DIOs. When an intermediate router adds itself to a route, it MUST ensure that the IPv6 address added to the route is accessible in both forward and backward directions. To improve the diversity of the routes being discovered, an intermediate router SHOULD keep track of multiple partial routes to be advertised in the P2P-RD0 inside its DIO. When the router generates its DIO, it SHOULD randomly select the partial route to be included in the P2P-RD0.

#### [9.5.](#) Additional Processing of a P2P Mode DIO At The Target

The target discards a received P2P mode DIO with no further processing if the routing metrics inside the DIO do not satisfy the mandatory constraints. Otherwise, the target MAY select the route

contained in the P2P-RD0 for further processing. This document does not prescribe a particular method for the target to select such routes. Examples include selecting the desired number of routes as they are identified or selecting the best routes discovered over a certain time period. If multiple routes are desired, the target SHOULD avoid selecting routes that have large segments in common. If a discovered route is bidirectional (D=1), the target MAY store the route in backward direction, obtained by reversing the discovered forward route, for use as a source route to reach the origin. After selecting a route, the target sends a Discovery Reply Object (DRO) message back to the origin (identified by the DODAGID field in the DIO). In this DRO, the target includes a P2P-RD0 that contains the selected route inside the Address vector. The P2P-RD0 included in the DRO message MUST copy the H flag from the P2P-RD0 inside the received DIO message. The other fields inside the P2P-RD0 MUST be set as specified in [Section 7](#). The mechanism for the propagation of DRO messages is described in [Section 8](#).

The target MAY set the A flag inside the DRO message if it desires the origin to send back a DRO-ACK message on receiving the DRO. In this case, the target waits for DRO\_ACK\_WAIT\_TIME duration for the DRO-ACK message to arrive. Failure to receive the DRO-ACK message within this time duration causes the target to retransmit the DRO message. The target MAY retransmit the DRO message in this fashion up to MAX\_DRO\_RETRANSMISSIONS times. The values of DRO\_ACK\_WAIT\_TIME and MAX\_DRO\_RETRANSMISSIONS are defined in [Section 12](#).

The target MAY include a Metric Container Option in the DRO message. This Metric Container contains the end-to-end routing metric values for the route specified in the P2P-RD0. The target MAY set the stop flag inside the DRO message (and detach from the temporary DAG) if it has already selected the desired number of routes. A target MUST NOT forward a P2P mode DIO any further.

#### [9.6.](#) Processing a DRO At An Intermediate Router

When a router receives a DRO message that does not list its IPv6 address in the DODAGID field, the router MUST process the received message in the following manner:

- o If the stop flag inside the received DRO is set and the router currently belongs to the temporary DAG identified by the (RPLInstanceID, DODAGID and Version fields of the) DRO, the router SHOULD cancel any pending DIO transmissions for this temporary DAG. Additionally, the router MAY detach from the temporary DAG immediately.

- o An intermediate router MUST ignore any Metric Container Option contained in the DRO message.
- o If Address[NH] element inside the Route Discovery Option lists the router's own IPv6 address, the router is a part of the route carried in the P2P-RD0. In this case, the router MUST do the following:
  - \* If the H flag inside the P2P-RD0 inside the DRO message is set, the router SHOULD store the state for the forward hop-by-hop route carried inside the P2P-RD0. This state consists of:
    - + The RPLInstanceID and the DODAGID fields of the DRO.
    - + The route's destination, the target (identified by Target field in P2P-RD0).
    - + 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).

The router MUST drop the DR0 message with no further processing if the H flag inside the P2P-RD0 is set but the router chooses not to store the state for the hop-by-hop route.

- \* 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 DR0 and the next hop information in the state does not match the next hop indicated in the received DR0, the router MUST drop the DR0 message with no further processing.
- \* The router MUST decrement the NH field inside the P2P-RD0 and send the DR0 further via link-local multicast.

### [9.7.](#) Processing a DR0 At The Origin

When a router receives a DR0 message that lists its IPv6 address in the DODAGID field, the router recognizes itself as the origin for the corresponding P2P-RPL route discovery and processes the P2P-RD0 contained in the DR0 in the following manner.

If the stop flag inside the received DR0 is set and the origin still belongs to the temporary DAG it initiated, it SHOULD cancel any pending DIO transmissions for this temporary DAG. Additionally, the origin MAY detach from the temporary DAG immediately.

If the P2P-RD0 inside the DR0 identifies the discovered route as a

source route (H=0), the origin SHOULD store in its memory the discovered route contained in the Address vector.

If the P2P-RD0 inside the DR0 identifies the discovered route as a hop-by-hop route (H=1), the origin SHOULD store in its memory the state for the discovered route in the manner described in [Section 9.6](#).

If the received DR0 message contains a Metric Container Option as well, 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 the target.

If the A flag is set to one in the received DRO message, the origin SHOULD generate a DRO-ACK message as described in [Section 10](#) and unicast the message to the target. The origin MAY source route the DRO-ACK message to the target using the route contained in the received DRO. If the received DRO established a hop-by-hop route to the target, the origin MAY send the DRO-ACK message along this route. [Section 11](#) describes how a packet may be forwarded along a route discovered using P2P-RPL.

[10.](#) The Discovery Reply Object Acknowledgement (DRO-ACK)

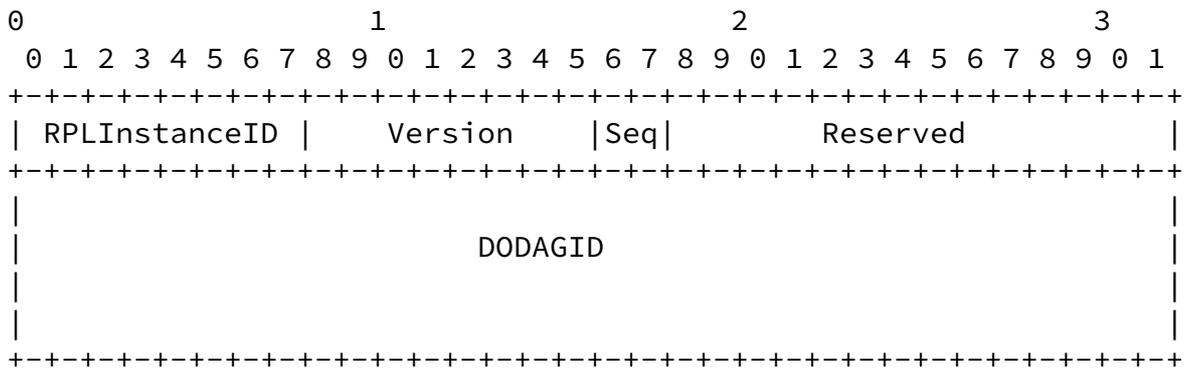


Figure 3: Format of the base Discovery Reply Object Acknowledgement (DRO-ACK)

A 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 DRO messages are prone to loss due to reasons associated with wireless communication. Since a 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 DRO message (e.g., because

of its inability to store the state for the hop-by-hop route the DRO is establishing). To protect against the potential failure of a DRO message to reach the origin, the target MAY request the origin to send back a DRO Acknowledgement (DRO-ACK) message on receiving a DRO message. Failure to receive such an acknowledgement within the

DRO\_ACK\_WAIT\_TIME interval of sending the DRO message forces the target to resend the message.

This section defines two new RPL Control Message types: DRO Acknowledgement (DRO-ACK; with code 0x05; to be confirmed by IANA) and Secure DRO-ACK (with code 0x85; to be confirmed by IANA). A DRO-ACK message MUST travel as a unicast message from the origin to the target. The format of a base DRO-ACK message is shown in Figure 3. Various fields in a DRO-ACK message MUST have the same values as the corresponding fields in the DRO message. The field marked as "Reserved" MUST be set to zero on transmission and MUST be ignored on reception. A Secure DRO-ACK message follows the format in Figure 7 of [[I-D.ietf-roll-rpl](#)], where the base format is same as the base DRO-ACK shown in Figure 3.

## 11. Packet Forwarding Along a P2P-RPL Route

This document specifies a mechanism to discover P2P routes, which can be either source routes or hop-by-hop ones. A packet MAY use an SRH header [[I-D.ietf-6man-rpl-routing-header](#)] to travel 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 to identify the route. This is because P2P-RPL route discovery does not use globally unique RPLInstanceID values and hence both the RPLInstanceID, which is a local value assigned by the origin, and the DODAGID, which is an IPv6 address belonging to the origin, are required to uniquely identify a P2P-RPL hop-by-hop route to a particular destination. A packet MAY include an RPL option [[I-D.ietf-6man-rpl-option](#)] inside the IPv6 hop-by-hop options header to travel along a hop-by-hop route established using P2P-RPL. In this case, the origin MUST set the DODAGID of the P2P-RPL route as the source IPv6 address of the packet. Further, the origin MUST specify the RPLInstanceID, associated with the P2P-RPL route, inside the RPL option and set the O flag inside the RPL option to 1. A router receiving this packet will check the O flag inside the RPL option and correctly infer the source IPv6 address of the packet as the DODAGID of the hop-by-hop route to be used for forwarding the packet further.

## 12. Constants

This document defines the following constants:

- o `DRO_ACK_WAIT_TIME`: The time duration a target waits for the DRO-ACK before retransmitting a DRO message. `DRO_ACK_WAIT_TIME` has a value of 1 second.
- o `MAX_DRO_RETRANSMISSIONS`: The maximum number of times a DRO message may be retransmitted if the target does not receive a DRO-ACK in response. `MAX_DRO_RETRANSMISSIONS` has a value 2.

## 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 would, in practice, have strict restrictions on the DAGs that may join. Thus, the problem described in the preceding paragraph may not occur in practice.

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

The security considerations for the operation of the reactive P2P route discovery mechanism described in this document are similar to the ones for the operation of RPL (as described in Section 19 of [\[I-D.ietf-roll-rpl\]](#)). [Section 10](#) of RPL specification

Internet-Draft

[draft-ietf-roll-p2p-rpl-06](#)

January 2012

[I-D.ietf-roll-rpl] describes a variety of security mechanisms to provide data confidentiality, authentication, replay protection and delay protection services. Each RPL control message has a secure version that allows the specification of the level of security and the algorithms used to secure the message. The mechanism defined in this document is based on the use of DIOs to form temporary DAGs and discover P2P routes. These DIOs can be used in their secure versions if desired. New RPL control messages defined in this document (DRO and DRO-ACK) have secure versions as well. Thus, a particular deployment of the reactive P2P route discovery mechanism described in this document can analyze its security requirements and use the appropriate set of RPL security mechanisms that meet those requirements.

## [15.](#) IANA Considerations

### [15.1.](#) Additions to DIO Mode of Operation

IANA is requested to allocate a new value in the "DIO Mode of Operation" registry for the "P2P Route Discovery Mode" described in this document.

MOP Value	Description	Reference
4	Reactive P2P route discovery mode of operation	This document

#### DIO Mode of Operation

### [15.2.](#) Additions to RPL Control Message Options

IANA is requested to allocate a new value in the "RPL Control Message Options" registry for the "Route Discovery Option" described in this document.

Value	Meaning	Reference
-------	---------	-----------

## RPL Control Message Options

Goyal, et al.

Expires July 22, 2012

[Page 24]

Internet-Draft

[draft-ietf-roll-p2p-rpl-06](#)

January 2012

### [15.3.](#) Additions to RPL Control Codes

IANA is requested to allocate new code points in the "RPL Control Codes" registry for the "Discovery Reply Object" and "Discovery Reply Object Acknowledgement" (and their secure versions) described in this document.

Code	Description	Reference
0x04	Discovery Reply Object	This document
0x05	Discovery Reply Object Acknowledgement	This document
0x84	Secure Discovery Reply Object	This document
0x85	Secure 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, Thomas Clausen, Richard Kelsey, Phil Levis, Zach Shelby, Pascal Thubert, Hristo Valev and JP Vasseur.

## [17.](#) References

### [17.1.](#) 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

Goyal, et al.

Expires July 22, 2012

[Page 25]

---

Internet-Draft

[draft-ietf-roll-p2p-rpl-06](#)

January 2012

Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

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

## [17.2](#). 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-06](#) (work in progress),  
December 2011.

[I-D.ietf-6man-rpl-routing-header]

Hui, J., Vasseur, J., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with RPL",  
[draft-ietf-6man-rpl-routing-header-07](#) (work in progress),  
December 2011.

[I-D.ietf-roll-of0]

Thubert, P., "RPL Objective Function Zero",  
[draft-ietf-roll-of0-20](#) (work in progress), September 2011.

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

Goyal, M., Baccelli, E., Brandt, A., and J. Martocci, "A Mechanism to Measure the Quality of a Point-to-point Route in a Low Power and Lossy Network",  
[draft-ietf-roll-p2p-measurement-02](#) (work in progress),

October 2011.

[I-D.ietf-roll-terminology]

Vasseur, J., "Terminology in Low power And Lossy Networks", [draft-ietf-roll-terminology-06](#) (work in progress), September 2011.

[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. Vermeulen, "Building Automation Routing Requirements in Low-Power and Lossy Networks", [RFC 5867](#), June 2010.

Goyal, et al.

Expires July 22, 2012

[Page 26]

---

Internet-Draft

[draft-ietf-roll-p2p-rpl-06](#)

January 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](mailto:mukul@uwm.edu)

Emmanuel Baccelli  
INRIA

Phone: +33-169-335-511  
Email: [Emmanuel.Baccelli@inria.fr](mailto: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