Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: August 11, 2011

M. Goyal, Ed.
University of Wisconsin Milwaukee
E. Baccelli
INRIA
A. Brandt
Sigma Designs
R. Cragie
Gridmerge Ltd
J. Martocci
Johnson Controls
C. Perkins
Tellabs Inc

February 7, 2011

Reactive Discovery of Point-to-Point Routes in Low Power and Lossy Networks
draft-ietf-roll-p2p-rpl-02

Abstract

Point to point (P2P) communication between arbitrary IPv6 routers and hosts in a Low power and Lossy Network (LLN) is a key requirement for many applications. RPL, the IPv6 Routing Protocol for LLNs, constrains the LLN topology to a Directed Acyclic Graph (DAG) and requires the P2P routing to take place along the DAG links. Such P2P routes may be suboptimal and may lead to traffic congestion near the DAG root. This document specifies a P2P route discovery mechanism, complementary to the RPL base functionality. This mechanism allows an RPL-aware IPv6 router or host to discover and establish, on demand, one or more routes to another RPL-aware IPv6 router or host in the LLN such that the discovered routes meet a specified cost criteria.

Table of Contents

1. Introduction 3
2. The Use Cases 4
3. Terminology 5
4. Applicability 6
5. Functional Overview 7
6. Propagation of Discovery Messages 8
6.1. The Route Discovery Option 9
6.2. Setting a DIO Carrying a Route Discovery Option 10
6.3. Joining a Temporary DAG 12
6.4. Processing a DIO Carrying a Route Discovery Option 12
6.5. Additional Processing of a DIO Carrying a Route Discovery Option At An Intermediate Router 13
6.6. Additional Processing of a DIO Carrying a Route Discovery Option At The Target Node 14
7. Propagation of Discovery Reply Messages 14
7.1. The Discovery Reply Object (DRO) 15
7.1.1. The Source Route Option 17
7.1.2. Processing a DRO At An Intermediate Router 19
7.2. DRO as Acknowledgement for Backward Source Routes 19
7.3. DRO as Carrier of Forward/Bidirectional Source Routes . . 19
7.4. Establishing Hop-by-hop Routes Via DRO 20
8. Security Considerations 20
9. IANA Considerations 20
10. Acknowledgements 20
11. References 21
11.1. Normative References 21
11.2. Informative References 22
Authors' Addresses 22

1. Introduction

RPL [I-D.ietf-roll-rpl] provides multipoint-to-point (MP2P) routes from nodes in a Low power and Lossy Network (LLN) to a sink node by organizing the nodes along a Directed Acyclic Graph (DAG) rooted at the sink. The nodes determine their position in the DAG so as to optimize their routing cost on the path towards the DAG root. A node advertises its position (the "rank") in the DAG by originating a DODAG Information Object (DIO) message. The DIO message is sent via link-local multicast and also includes information such as the DAG root's identity, routing metrics/constraints [I-D.ietf-roll-routing-metrics] and the objective function (OF) in use. When a node joins the DAG, it determines its own rank in the DAG based on that advertised by its neighbors and originates its own DIO message.

RPL enables point-to-multipoint (P2MP) routing from a node to its descendants in the DAG by allowing a node to send a Destination Advertisement Object (DAO) upwards along the DAG. The DAO carries potentially aggregated information regarding the descendants (and other local prefixes) reachable through the node originating this DAO.

RPL also provides mechanisms for point-to-point (P2P) routing between any two nodes in the DAG. If the destination is within the source's radio range, the source may directly send packets to the destination. Otherwise, a packet's path from the source to the destination depends on the storing/non-storing operation mode of the DAG. In non-storing mode operation, only the DAG root maintains downward routing information and hence a packet travels all the way to the DAG root, which then sends it towards its destination using a source route. In storing mode operation, if the destination is a DAG descendant and the source maintains "downwards" hop-by-hop routing state about this descendant, it can forward the packet to a descendant router closer to the destination. Otherwise, the source sends the packet to a DAG parent, which then applies the same set of rules to forward the packet further. Thus, a packet travels up the DAG until it reaches a node that knows of the downwards route to the destination and then it travels down the DAG towards its destination. A node may or may not maintain routing state about a descendant depending on whether its immediate children send it such information in their DAOs. Thus, in the best case with storing mode operation, the "upwards" segment of the P2P route between a source and a destination ends at the first common ancestor of the source and the destination. In the worst case, the "upwards" segment would extend all the way to the DAG root. In both storing and non-storing mode operations, if the destination did not originate a DAO, the packet will travel all the way to the DAG's root, where it will be dropped.

The P2P routing functionality available in RPL may be inadequate for applications in the home and commercial building domains for the following reasons [I-D.brandt-roll-rpl-applicability-home-building] [RFC5826][RFC5867]:

- The need to maintain routes "proactively", i.e., every possible destination in the DAG must originate a DAO.

- Depending on the network topology and OF/metrics in use, the constraint to route only along a DAG may cause significantly suboptimal P2P routes and severe traffic congestion near the DAG root.

Thus, there is a need for a mechanism that provides source-initiated discovery of P2P routes that are not along an existing DAG. This document describes such a mechanism, complementary to the basic RPL functionality.

The specified mechanism is based on a reactive on-demand approach, which enables a node to discover one or more routes in either direction between itself and another node in the LLN without any restrictions regarding the existing DAG-membership of the links that such routes may use. The discovered routes may be source routes or hop-by-hop routes. The 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, necessary to help decide whether to initiate a 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, described in [I-D.goyal-roll-p2p-measurement], can be used to determine the metric constraints for use in the route discovery mechanism described in this document.

2. The Use Cases

The mechanisms described in this document are intended to be employed as complementary to RPL in specific scenarios that need point-to-point (P2P) routes 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 P2P routes 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]. Specifically, the term node refers to an RPL router or an RPL host as defined in [I-D.ietf-roll-rpl]. This document introduces the following terms:

Origin : The RPL node initiating the route discovery. The origin acts as one end point of the routes to be discovered.

Target : The RPL node at the other end point of the routes to be discovered.

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

Forward Route: A route from the origin to the target.

Backward Route: A route from the target to the origin.

Bidirectional Route: A route that can carry traffic in both directions.

Source Route: A complete and ordered list of routers that can be used by a packet to travel from a source node to a destination node. Such source routes can be carried by a packet in a Type 4 Routing Header [I-D.ietf-6man-rpl-routing-header].

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.

Propagation Constraints: The constraints on the routing metrics [I-D.ietf-roll-routing-metrics] that MUST be satisfied before an intermediate router or the target will process the Route Discovery Option (defined in this document) contained inside a DODAG Information Object (DIO).

Route Constraints: Additional constraints on the routing metrics [I-D.ietf-roll-routing-metrics] that the target MUST enforce on the received DIOs.

4. Applicability

The route discovery mechanism, described in this document, may be invoked by an origin when no route exists between itself and the target or when the existing routes do not satisfy the desired performance requirements. The mechanism is designed to discover one or more "good enough" routes in either direction between an origin and a target. In some application contexts, the metric constraints that the discovered routes 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 propagation or route constraint. In other application contexts, the origin may need to measure the cost of an existing route to the target to determine the propagation/route 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 propagation/route constraint. The functionality required to measure the cost of an existing route between the origin and the target is described in [I-D.goyal-roll-p2p-measurement]. In case, 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 propagation/ route 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.

This document describes an on-demand discovery mechanism for P2P routes that is complementary to the proactive routes offered by RPL base functionality. The mechanism described in this document 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 in 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 this mechanism.

5. Functional Overview

This section contains a high level description of the route discovery mechanism proposed in this document.

The route discovery begins with the origin generating a "Discovery" message. The origin indicates in the message:

- The target;
- The relevant routing metrics;
- The constraints on how far the Discovery message may travel (henceforth called the propagation constraints);
- Additional constraints that the target must enforce (henceforth called the route constraints);
- The direction (forward: from the origin to the target; backward: from the target to the origin; or bidirectional) of the route being discovered;
- The desired number of routes (in case forward/bidirectional routes are being discovered);
- Whether the route is a source route or a hop-by-hop one.

The Discovery message propagates via IPv6 link-local multicast with a receiving router discarding the message if it does not satisfy the propagation constraints or if the hop-by-hop routes are desired and the router cannot store the state for such a route. As a copy of the Discovery message travels towards the target, it accumulates the relevant routing metric values as well as the route it takes. When the target receives a Discovery message, it applies both the propagation constraints and the route constraints on the routing metrics inside the Discovery message. Thus, the discovered routes satisfy both the propagation constraints as well as the route constraints, although the propagation of Discovery messages is guided by propagation constraints alone. Using only a subset of the constraints as propagation constraints simplifies the operation of intermediate routers, an important consideration in many LLN application domains [RFC5826][RFC5867].

The route discovery process may result in the discovery of several routes. This document does not specify how the target selects routes

among the ones discovered. Example selection methods include selecting routes as they are discovered or selecting the best routes discovered over a certain time period.

If the origin had requested the discovery of backward source-routes, the target caches one or more discovered source-routes. Additionally, the target sends one or more "Discovery Reply" messages to the origin to acknowledge the discovery of these routes.

If the origin had requested the discovery of "n" forward source-routes, the target sends "n" discovered source-routes it selects to the origin in one or more Discovery Reply messages.

If the origin had requested the discovery of "n" bidirectional source-routes, the target caches "n" discovered source-routes it selects and also sends these routes to the origin in one or more Discovery Reply messages.

If the origin had requested the discovery of "n" forward/backward/ bidirectional hop-by-hop routes, the target sends out a Discovery Reply message to the origin for each one of the "n" discovered routes it selects. The Discovery Reply message travels towards the origin along the discovered route. As this message travels towards the origin, it establishes appropriate forward/backward routing state in the routers on the path.

6. Propagation of Discovery Messages

RPL uses DIO message propagation to build a DAG. The DIO message travels via IPv6 link-local multicast. Each node joining the DAG determines a rank for itself 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. The DIO message generation at a node is further controlled by a trickle timer that allows a node to avoid generating unnecessary messages [I-D.ietf-roll-trickle]. The link-local multicast based propagation, trickle-controlled generation and the rank-based poisoning of messages traveling in the wrong direction (towards the DAG root) provide powerful incentives to use the DIO message as the Discovery message and propagate the DIO/ Discovery message by creating a "temporary" DAG. Such an approach also allows reuse of the routing metrics, objective function and packet forwarding framework developed for RPL. The routing metrics used for the creation of this temporary DAG SHOULD be same as (or be a subset of) the routing metrics being used for route discovery. Similarly, the objective function, used for rank calculation in the temporary DAG, SHOULD be same as the objective function that

determines the aggregated cost of a route when limited to the routing metrics being used for temporary DAG creation.

The propagation constraints limit the spread of the temporary DAG. The temporary DAG restricts the network topology within which the route discovery takes place. The routes accumulated by the DIOs lie within this restricted topology and implicitly satisfy the propagation constraints. As the target receives a DIO, it additionally applies the route constraints on the accumulated route. Thus, for successful route discovery, the propagation constraints and the route constraints MUST be compatible. The division of the overall constraints in the two categories is an implementation specific decision. If desired, an implementation MAY consider all the constraints as propagation constraints and keep the set of route constraints empty.

6.1. The Route Discovery Option

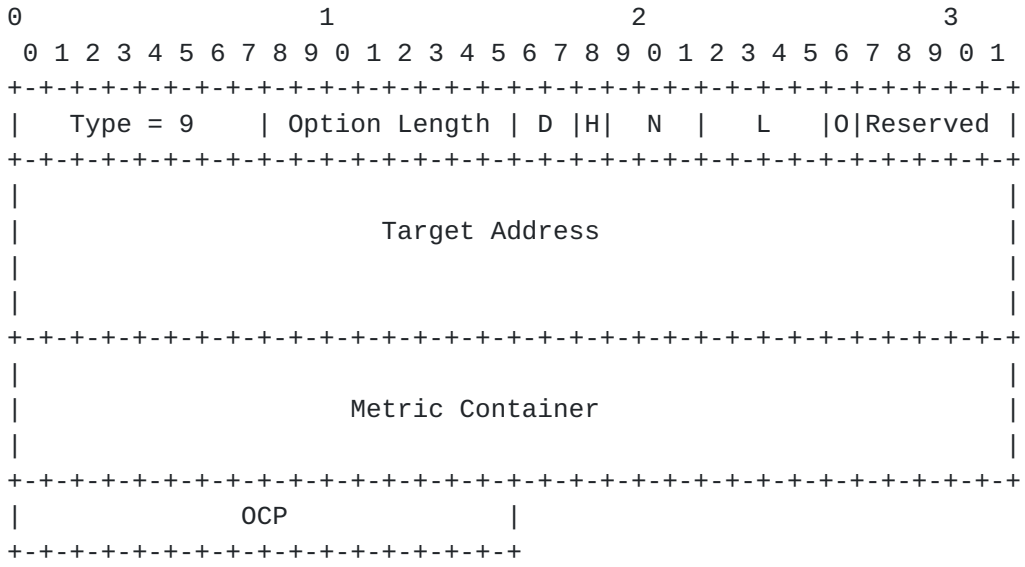

Figure 1: Format of the Route Discovery Option

In order to be used as a Discovery message, a DIO MUST carry a "Route Discovery" option illustrated in Figure 1. A DIO MUST NOT carry more than one Route Discovery options. A router MUST ignore the second and subsequent Route Discovery options carried by a DIO. A Route Discovery option consists of the following fields:

- Option Type = 0x09 (to be confirmed by IANA).

- Option Length = The length of Route Discovery option including any Metric Container and OCP fields.

- D: A 2-bit field that indicates the direction of the desired routes:

  * D = 0x00: Forward;

  * D = 0x01: Backward;

  * D = 0x02: Bidirectional.

  The D field also specifies the direction in which the link-level metrics being used for route discovery should be measured.

- H: This flag, when set, indicates if hop-by-hop routes are desired. The flag is cleared if source routes are desired.

- N: A 3-bit unsigned integer indicating the number of routes desired. Used when forward or bidirectional routes are being discovered.

- L: A 4-bit field containing an exponent of 2, such that 2 raised to the power L specifies, in units of seconds, the minimum "Life Time" of the temporary DAG, i.e., the minimum duration a router joining the temporary DAG must maintain its membership in the DAG.

- O: This flag, when set, indicates that an OCP field is present in the Route Discovery option.

- Target Address: The IPv6 address of the target.

- Metric Container: Contains the route constraints that the target MUST apply. Any metric objects contained in this metric container MUST be ignored.

- OCP: 16 bit unsigned integer. An optional field, present only if the O flag is set, This field indicates the objective function that MAY be used by the target to compare two discovered routes.

6.2. Setting a DIO Carrying a Route Discovery Option

A DIO message that carries a Route Discovery option MUST set the Base Object, described in [I-D.ietf-roll-rpl], in the following manner:

- RPLInstanceID: RPLInstanceID MUST be a local value as described in Section 5.1 of [I-D.ietf-roll-rpl]. The origin MUST ensure that different RPLInstanceID values are used in two or more concurrent route discoveries it initiates.

- Grounded (G) Flag: MUST be cleared since the objective of DAG formation is propagation of Route Discovery option. This DAG is temporary in nature and is not used for routing purpose.

- Destination Advertisement Supported (A) Flag: MUST be cleared for same reasons as described above.

- Destination Advertisement Trigger (T) Flag: MUST be cleared.

- Mode of Operation (MOP): This document suggests a new value (0x04) for this field (to be confirmed by IANA).

- DODAGPreference (Prf): TBD

- Destination Advertisement Trigger Sequence Number (DTSN): TBD

- DODAGID: IPv6 address of the origin.

The other fields in the Base Object are set as per the rules described in [I-D.ietf-roll-rpl].

The DODAG Configuration option, carried in the DIO message, specifies the parameters for the trickle timer operation that governs the generation of DIO messages by routers joining the temporary DAG. The future versions of this document will specify the default values to be used for these parameters. The other fields defined in the DODAG Configuration option are set as follows:

- The MaxRankIncrease field MUST be set to 0 to disable local repair of the temporary DAG.

- This document RECOMMENDS a value 1 for the MinHopRankInc field.

- Objective Code Point (OCP): The OCP to be used for temporary DAG formation. The objective function used for temporary DAG formation SHOULD be compatible with the objective function to determine the aggregated cost of a discovered route.

A DIO, that contains a Route Discovery option, MUST specify the propagation constraints in one or more Metric Container options placed outside the Route Discovery option. As mentioned before, the route constraints are listed in the Metric Container option placed inside the Route Discovery option. The routing metrics being used for temporary DAG formation SHOULD be same as or a subset of the routing metrics being used for route discovery. These routing metrics MUST be placed in the Metric Container options placed outside the Route Discovery option. Any link-level metrics being used for route discovery MUST be measured in the direction indicated by the D

field in Route Discovery option. Any metric object contained inside the Metric Container inside the Route Discovery option MUST be ignored.

A DIO, carrying a Route Discovery option, MUST NOT carry any Route Information or Prefix Information options described in [I-D.ietf-roll-rpl].

6.3. Joining a Temporary DAG

When a node joins a temporary DAG advertized by a DIO carrying the Route Discovery option, it MUST maintain its membership in the DAG for the Minimum Life Time duration listed in the Route Discovery option. Maintaining membership in the DAG implies remembering:

- The RPLInstanceID, the DODAGID and the DODAGVersionNumber for the temporary DAG;
- The node's rank in the temporary DAG as well as the address of at least one DAG parent;
- The propagation and the route constraints being used;
- In case of intermediate routers, the values for the routing metrics, along with the associated source route from the origin untill this node (carried in a Record Route IPv6 Extension Header proposed in [I-D.thubert-6man-reverse-routing-header]), contained in the best DIO (in terms of the routing metrics and potentially using the OCP specified in the DODAG Configuration option) received so far.

Although the main purpose of a temporary DAG's existence is to facilitate the propagation of the Route Discovery option, the temporary DAG MAY also be used for the Discovery Reply Object (defined in Section 7.1 to travel from the target to the origin. Hence, a node in a temporary DAG SHOULD also remember the address of at least one DAG parent that provides the best known path back to the origin. A node SHOULD delete information about a temporary DAG once the duration of its membership in the DAG has exceeded the DAG's minimum life time.

6.4. Processing a DIO Carrying a Route Discovery Option

The rules for DIO processing and transmission, described in Section 7 of RPL [I-D.ietf-roll-rpl], apply to DIOs carrying a Route Discovery option as well except as modified in this document.

The following rules for processing a DIO carrying a Route Discovery

Option apply to both intermediate routers and the target.

A node MUST discard a DIO with no further processing if:

- The DIO contains two or more Route Discovery options;
- The node can not evaluate one or more of the propagation constraints listed in a Metric Container inside the DIO.

A node MUST discard a DIO with no further processing if any of the following conditions are found to be true while processing a Route Discovery option contained in that DIO:

- The H field is set, i.e., hop-by-hop routes are desired, and the node chooses not to participate in a hop-by-hop route;
- The node cannot maintain its membership in the temporary DAG for the minimum life time specified in the Route Discovery option.

A node MUST update the values of link-level routing metrics included inside the DIO in accordance with the D field in the Route Discovery option. If the D field is 0x00, i.e., the forward routes are being discovered, any link-level routing metric MUST be measured in the direction towards the node receiving the DIO. If the D field is 0x01, i.e., the backward routes are being discovered, any link-level routing metric MUST be measured in the direction towards the node originating the DIO. If the D field is 0x02, i.e., the bidirectional routes are being discovered, any link-level routing metric MUST be calculated so as to take in account the metric's value in both directions. The rules for calculating bidirectional metric values will be specified in a separate document.

6.5. Additional Processing of a DIO Carrying a Route Discovery Option At An Intermediate Router

An intermediate router MUST process a received DIO, carrying a Route Discovery option, in accordance with the following rules.

An intermediate router MUST discard the DIO with no further processing if the routing metric values do not satisfy one or more propagation constraints listed in the DIO. The router MAY check the route constraints listed inside the Route Discovery option and discard the DIO with no further processing if these constraints are not met.

An intermediate router MUST determine if this DIO is the best it has received so far for this temporary DAG in terms of the routing metrics (potentially using the OCP in the DODAG Configuration

object). If yes, the intermediate router MUST remember the routing metric values contained in this DIO along with the route travelled by the DIO so far and reset the trickle timer associated with the temporary DAG.

When the trickle timer associated with the temporary DAG fires, an intermediate router MUST generate a new DIO for this temporary DAG carrying the Route Discovery option, the best metric values it knows and the source route associated with these values (in a Record Route IPv6 extension header [I-D.thubert-6man-reverse-routing-header]).

6.6. Additional Processing of a DIO Carrying a Route Discovery Option At The Target Node

A node MUST process a received DIO, carrying a Route Discovery option that lists this node as the target, in accordance with the following rules.

A target MUST discard the DIO with no further processing if it can not evaluate the route constraints listed inside the Route Discovery option or if the routing metric values do not satisfy one or more of the propagation and route constraints.

Otherwise, the target considers the source route accumulated by the received DIO as one of the discovered routes. This document does not prescribe a particular method for selecting routes among the discovered ones. Suppose the Route Discovery option requires the discovery of "n" routes. The target may select these "n" routes in any manner it desires. Example selection methods include selecting the first "n" routes it discovers or selecting the "n" best routes discovered over a certain time period, potentially using the OCP specified in the Route Discovery option for route comparison.

After selecting one or more discovered routes, the target MUST send one or more RPL Control Messages carrying a Discovery Reply Object (defined in the next section) back to the origin (identified by the DODAGID field in the DIO Base Object) as specified in Section 7.

A target MUST NOT forward a DIO carrying a Route Discovery option any further.

7. Propagation of Discovery Reply Messages

7.1. The Discovery Reply Object (DRO)

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

This document defines a new RPL Control Message type, the Discovery Reply Object (DRO) with code 0x04 (to be confirmed by IANA), that serves one of the following functions:

- An acknowledgement from the target to the origin regarding the successful discovery of backward source routes;
- Carries one or more forward/bidirectional source routes from the target to the origin;
- Establishes one hop-by-hop forward/backward/bidirectional route as it travels from the target to the origin.

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

- RPLInstanceID: The RPLInstanceID of the temporary DAG used for route discovery.
- Version: The Version of the temporary DAG used for route discovery.
- D: A 2-bit field that indicates the direction of the discovered routes:

* D = 0x00: Forward;

* D = 0x01: Backward;

* D = 0x02: Bidirectional.

This field has the same value as the corresponding field in the Route Discovery option.

- H: A flag that is set if the DRO is establishing an hop-by-hop route. If this flag is set, the DRO MUST travel from the target to the origin along the hop-by-hop route being established and MUST include one Source Route option (defined in Section 7.1.1) that contains the remaining routers on this route (as described in Section 7.4). Since the state that a node needs to maintain regarding a hop-by-hop route includes the RPLInstanceID, the DODAGID and the IPv6 address of the route's destination, a DRO with H flag set MUST also include:

  * The DODAGID of the temporary DAG used for route discovery; and

  * The Target Address if the hop-by-hop route is forward or bidirectional.

  The H flag MUST be clear if the DRO carries (or is an acknowledgement for the discovery of) one or more source routes contained in the Source Route options. The target can unicast such a DRO to the origin or send it along the temporary DAG used for route discovery. If the DRO is unicast to the origin, it MUST NOT include the DODAGID and Target Address fields. If the DRO is sent along the temporary DAG, it MUST include the DODAGID field and MUST NOT include the Target Address field.

- N: A 3-bit field that indicates the number of source routes carried or acknowledged in the DRO. This field MUST have value 1 if the DRO is establishing a hop-by-hop route.

- Reserved: These bits are reserved for future use. These bits MUST be set to zero on transmission and MUST be ignored on reception.

- DODAGID: The DODAGID of the temporary DAG used for route discovery. The DODAGID also identifies the origin. This field MUST be present in the DRO if the H flag is set or if the H flag is clear but the DRO needs to travel along the temporary DAG. Otherwise, this field need not be present in the DRO. The RPLInstanceID, the Version and the DODAGID together uniquely identify the temporary DAG used for route discovery and can be copied from the Base Object of the DIO advertizing the temporary

DAG.

- Target Address: The IPv6 address of the target generating the Discovery Reply Object. This field MUST be present in the DRO if the H flag is set and the hop-by-hop route being established is forward or bidirectional.

- Options: The Discovery Reply Object MAY carry up to N Source Route options (defined in the next section) with each such option carrying a source route and optionally followed by a Metric Container option that lists the aggregated values for the routing metrics for the source route.

7.1.1. The Source Route Option

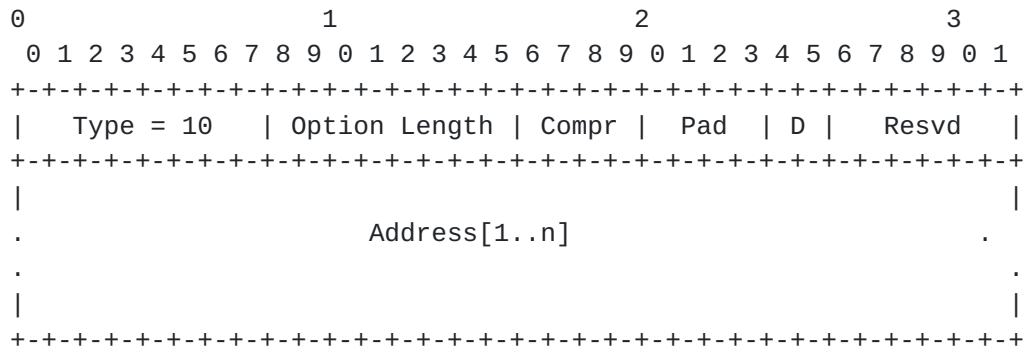

Figure 3: Format of the Source Route Option

The Source Route option, illustrated in Figure 3, carries a source route. When a Source Route option carries a complete source route between the origin and the target, it MAY be immediately followed by a Metric Container option that contains the aggregated values of the routing metrics for this source route.

A Source Route option consists of the following fields:

- Option Type = 0x0A (to be confirmed by IANA).

- Option Length = Variable, depending on the size of the Addresses vector.

- Compr: 4-bit unsigned integer indicating the number of prefix octets that are elided from each address. For example, Compr value will be 0 if full IPv6 addresses are carried in the Addresses vector.

- Pad: 4-bit unsigned integer. Number of octets that are used for padding between Address[n] and the end of the Source Route option.

- D: A 2-bit field that indicates the direction of the source route:

  * D = 0x00: Forward, i.e., from the origin to the target;

  * D = 0x01: Backward i. e., from the target to the origin;

  * D = 0x02: Bidirectional.

  Note that the D field in a Source Route option is independent from the D field in the DRO containing the Source Route option.

- Resvd: These bits are reserved for future use. These bits MUST be set to zero on transmission and MUST be ignored on reception.

- Address[1..n]: Vector of addresses, numbered 1 to n. Each vector element has size (16 - Compr) octets.

Note that the format of the Source Route option is very similar to that of proposed Type 4 Routing Header [I-D.ietf-6man-rpl-routing-header].

A common network configuration for an RPL domain is that all routers within an LLN share a common prefix. The Source Route option uses the Compr field to allow compaction of the Address[1..n] vector when all entries share the same prefix as the DODAGID or the Target Address of the encapsulating Discovery Reply Object. The shared prefix octets are not carried within the Source Route option and each entry in Address[1..n] has size (16 - Compr) octets. When Compr is non-zero, there may exist unused octets between the last entry, Address[n], and the end of the Source Route option. The Pad field indicates the number of unused octets that are used for padding. Note that when Compr is 0, Pad MUST be null and carry a value 0.

The Source Route option MUST NOT specify a path that visits a router more than once. When generating a Source Route option, the target may not know the mapping between IPv6 addresses and routers. Minimally, the target MUST ensure that:

- The IPv6 Addresses do not appear more than once;

- The IPv6 addresses of the origin and the target do not appear in the Address vector.

Multicast addresses MUST NOT appear in a Source Route option.

7.1.2. Processing a DRO At An Intermediate Router

When an intermediate router receives a DRO with a clear H flag, it MUST forward the DRO to a parent node in the temporary DAG.

When an intermediate router receives a DRO that has H flag set and contains multiple Source Route options, the router MUST drop the DRO with no further processing.

When an intermediate router receives a DRO that has H flag set and contains a single Source Route option, the router processes the DRO as described in Section 7.4.

7.2. DRO as Acknowledgement for Backward Source Routes

After selecting one or more backward source routes, a target MAY send a DRO message to the origin as an acknowledgement for the discovered routes. A DRO, serving as an acknowledgement for backward source route discovery, has its D field set to 0x01 (indicating backward) while the H flag is cleared (indicating source route). The N field is set to indicate the number of discovered backward source routes being acknowledged. Such a DRO message MUST NOT contain any option.

The target MAY unicast this DRO message to the origin or it MAY forward the DRO message to a parent in the temporary DAG. The target should take into consideration the minimum life time of the temporary DAG when deciding to use it to send the DRO to the origin.

7.3. DRO as Carrier of Forward/Bidirectional Source Routes

The target MUST convey the discovered forward/bidirectional source routes to the origin via the Source Route options inside one or more DRO messages. Such a DRO message MUST have its D field set to 0x00 (if it carries forward routes) or 0x02 (if its carries bidirectional routes). Also, the H flag MUST be cleared and the N field MUST indicate the number of Source Route options in the DRO. Each Source Route option inside the DRO MAY immediately be followed by a Metric Container option that carries the aggregated values of the relevant routing metrics for this source route.

The target MAY unicast this DRO message to the origin or it MAY forward the DRO message to a parent in the temporary DAG. The target should take into consideration the minimum life time of the temporary DAG when deciding to use it to send the DRO to the origin.

7.4. Establishing Hop-by-hop Routes Via DRO

In order to establish a hop-by-hop route, the target MUST send a DRO message along the discovered route, which is specified in a Source Route option. The D field in the DRO MUST reflect the direction of the discovered route. The H bit in the DRO MUST be set and the DRO MUST include the DODAGID field. If a forward or bidirectional hop-by-hop route is being established, the DRO MUST include the Target Address field as well. The N field in the DRO MUST be set to 1 and the DRO MUST include exactly one Source Route option. The target forwards the DRO to the next hop along the discovered route and includes the discovered route, excluding itself and the origin, inside the Source Route option in backward direction. Thus, the D field in the Source Route option MUST be 0x01.

If the hop-by-hop route is in the backward direction, the target MUST establish the hop-by-hop state for the route before sending the DRO message. Such hop-by-hop state includes the RPLInstanceID, the DODAGID and the route's destination (in this case, the origin's address or the DODAGID).

A router receiving a DRO message MUST drop the DRO if the router cannot establish the hop-by-hop state for the route or if its own address does not appear as the first element in the Address vector in the Source Route option. Otherwise, the router MUST establish the hop-by-hop state in the direction specified in the D field in the DRO. The hop-by-hop state in the forward direction includes the RPLInstanceID, the DODAGID and the target's address. The hop-by-hop state in the backward direction includes the RPLInstanceID, the DODAGID and the origin's address. After establishing the hop-by-hop state, the router MUST remove its own address from the route contained in the Source Route option and forward the DRO to the next hop (Address[0] in the Source Route option).

8. Security Considerations

TBA

9. IANA Considerations

TBA

10. Acknowledgements

Authors gratefully acknowledge the contributions of the following

individuals (in alphabetical order) in the development of this document: Dominique Barthel, Thomas Clausen, Richard Kelsey, Zach Shelby, Pascal Thubert and JP Vasseur.

11. References

11.1. Normative References

[I-D.goyal-roll-p2p-measurement]
Goyal, M. and E. Baccelli, "A Mechanism to Measure the Quality of a Point-to-point Route in a Low Power and Lossy Network", draft-goyal-roll-p2p-measurement-01 (work in progress), October 2010.

[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-01 (work in progress), October 2010.

[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-01 (work in progress), October 2010.

[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-17 (work in progress), December 2010.

[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.

[I-D.thubert-6man-reverse-routing-header]
Thubert, P., "Reverse Routing Header",

draft-thubert-6man-reverse-routing-header-00 (work in progress), June 2010.

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

11.2. Informative References

[I-D.brandt-roll-rpl-applicability-home-building]
Brandt, A., Baccelli, E., and R. Cragie, "Applicability Statement: The use of RPL in Building and Home Environments", draft-brandt-roll-rpl-applicability-home-building-01 (work in progress), November 2010.

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

[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.

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/

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

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

Robert Cragie
Gridmerge Ltd
89 Greenfield Crescent
Wakefield WF4 4WA
UK

Phone: +44-1924910888
Email: robert.cragie@gridmerge.com

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

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

Charles Perkins
Tellabs Inc.

charliep@computer.org