

ROLL  
Internet-Draft  
Intended status: Standards Track  
Expires: January 3, 2019

S. Anamalamudi  
SRM University-AP  
M. Zhang  
Huawei Technologies  
AR. Sangi  
Huaiyin Institute of Technology  
C. Perkins  
Futurewei  
S.V.R.Anand  
Indian Institute of Science  
B. Liu  
Huawei Technologies  
July 2, 2018

Asymmetric AODV-P2P-RPL in Low-Power and Lossy Networks (LLNs)  
draft-ietf-roll-aodv-rpl-04

## Abstract

Route discovery for symmetric and asymmetric Point-to-Point (P2P) traffic flows is a desirable feature in Low power and Lossy Networks (LLNs). For that purpose, this document specifies a reactive P2P route discovery mechanism for both hop-by-hop routing and source routing: Ad Hoc On-demand Distance Vector Routing (AODV) based RPL protocol. Paired Instances are used to construct directional paths, in case some of the links between source and target node are asymmetric.

## Status of This Memo

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

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

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

This Internet-Draft will expire on January 3, 2019.

## Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	4
3. Overview of AODV-RPL . . . . .	6
4. AODV-RPL DIO Options . . . . .	7
4.1. AODV-RPL DIO RREQ Option . . . . .	7
4.2. AODV-RPL DIO RREP Option . . . . .	9
4.3. AODV-RPL DIO Target Option . . . . .	10
5. Symmetric and Asymmetric Routes . . . . .	11
6. AODV-RPL Operation . . . . .	13
6.1. Route Request Generation . . . . .	13
6.2. Receiving and Forwarding RREQ messages . . . . .	14
6.2.1. General Processing . . . . .	14
6.2.2. Additional Processing for Multiple Targets . . . . .	15
6.3. Generating Route Reply (RREP) at TargNode . . . . .	16
6.3.1. RREP-DIO for Symmetric route . . . . .	16
6.3.2. RREP-DIO for Asymmetric Route . . . . .	16
6.3.3. RPLInstanceID Pairing . . . . .	16
6.4. Receiving and Forwarding Route Reply . . . . .	17
7. Gratuitous RREP . . . . .	18
8. Operation of Trickle Timer . . . . .	19
9. IANA Considerations . . . . .	19
9.1. New Mode of Operation: AODV-RPL . . . . .	19
9.2. AODV-RPL Options: RREQ, RREP, and Target . . . . .	19
10. Security Considerations . . . . .	20
11. Future Work . . . . .	20
12. References . . . . .	20
12.1. Normative References . . . . .	20
12.2. Informative References . . . . .	21
Appendix A. Example: ETX/RSSI Values to select S bit . . . . .	21
Appendix B. Changelog . . . . .	22
B.1. Changes to version 02 . . . . .	22

B.2. Changes to version 03 . . . . .	23
Authors' Addresses . . . . .	23

## 1. Introduction

RPL[RFC6550] is a IPv6 distance vector routing protocol for Low-power and Lossy Networks (LLNs), and is designed to support multiple traffic flows through a root-based Destination-Oriented Directed Acyclic Graph (DODAG). Typically, a router does not have routing information for most other routers. Consequently, for traffic between routers within the DODAG (i.e., Point-to-Point (P2P) traffic) data packets either have to traverse the root in non-storing mode, or traverse a common ancestor in storing mode. Such P2P traffic is thereby likely to traverse longer routes and may suffer severe congestion near the DAG root [RFC6997], [RFC6998].

To discover better paths for P2P traffic flows in RPL, P2P-RPL [RFC6997] specifies a temporary DODAG where the source acts as a temporary root. The source initiates DIOs encapsulating the P2P Route Discovery option (P2P-RDO) with an address vector for both hop-by-hop mode (H=1) and source routing mode (H=0). Subsequently, each intermediate router adds its IP address and multicasts the P2P mode DIOs, until the message reaches the target node (TargNode), which then sends the "Discovery Reply" object. P2P-RPL is efficient for source routing, but much less efficient for hop-by-hop routing due to the extra address vector overhead. However, for symmetric links, when the P2P mode DIO message is being multicast from the source hop-by-hop, receiving nodes can infer a next hop towards the source. When TargNode subsequently replies to the source along the established forward route, receiving nodes determine the next hop towards TargNode. For hop-by-hop routes (H=1) over symmetric links, this would allow efficient use of routing tables for P2P-RDO messages instead of the "Address Vector".

RPL and P2P-RPL both specify the use of a single DODAG in networks of symmetric links, where the two directions of a link MUST both satisfy the constraints of the objective function. This disallows the use of asymmetric links which are qualified in one direction. But, application-specific routing requirements as defined in IETF ROLL Working Group [RFC5548], [RFC5673], [RFC5826] and [RFC5867] may be satisfied by routing paths using bidirectional asymmetric links. For this purpose, [I-D.thubert-roll-asymlink] described bidirectional asymmetric links for RPL [RFC6550] with Paired DODAGs, for which the DAG root (DODAGID) is common for two Instances. This can satisfy application-specific routing requirements for bidirectional asymmetric links in core RPL [RFC6550]. Using P2P-RPL twice with Paired DODAGs, on the other hand, requires two roots: one for the source and another for the target node due to temporary DODAG

formation. For networks composed of bidirectional asymmetric links (see Section 5), AODV-RPL specifies P2P route discovery, utilizing RPL with a new MoP. AODV-RPL makes use of two multicast messages to discover possibly asymmetric routes, which can achieve higher route diversity. AODV-RPL eliminates the need for address vector overhead in hop-by-hop mode. This significantly reduces the control packet size, which is important for Constrained LLN networks. Both discovered routes (upward and downward) meet the application specific metrics and constraints that are defined in the Objective Function for each Instance [RFC6552].

The route discovery process in AODV-RPL is modeled on the analogous procedure specified in AODV [RFC3561]. The on-demand nature of AODV route discovery is natural for the needs of peer-to-peer routing in RPL-based LLNs. AODV terminology has been adapted for use with AODV-RPL messages, namely RREQ for Route Request, and RREP for Route Reply. AODV-RPL currently omits some features compared to AODV -- in particular, flagging Route Errors, blacklisting unidirectional links, multihoming, and handling unnumbered interfaces.

## 2. Terminology

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

### AODV

Ad Hoc On-demand Distance Vector Routing[RFC3561].

### AODV-RPL Instance

Either the RREQ-Instance or RREP-Instance

### Asymmetric Route

The route from the OrigNode to the TargNode can traverse different nodes than the route from the TargNode to the OrigNode. An asymmetric route may result from the asymmetry of links, such that only one direction of the series of links fulfills the constraints in route discovery.

### Bi-directional Asymmetric Link

A link that can be used in both directions but with different link characteristics.

### DIO

DODAG Information Object

### DODAG RREQ-Instance (or simply RREQ-Instance)

RPL Instance built using the DIO with RREQ option; used for control message transmission from OrigNode to TargNode, thus enabling data transmission from TargNode to OrigNode.

DODAG RREP-Instance (or simply RREP-Instance)

RPL Instance built using the DIO with RREP option; used for control message transmission from TargNode to OrigNode thus enabling data transmission from OrigNode to TargNode.

Downward Direction

The direction from the OrigNode to the TargNode.

Downward Route

A route in the downward direction.

hop-by-hop routing

Routing when each node stores routing information about the next hop.

on-demand routing

Routing in which a route is established only when needed.

OrigNode

The IPv6 router (Originating Node) initiating the AODV-RPL route discovery to obtain a route to TargNode.

Paired DODAGs

Two DODAGs for a single route discovery process between OrigNode and TargNode.

P2P

Point-to-Point -- in other words, not constrained a priori to traverse a common ancestor.

reactive routing

Same as "on-demand" routing.

RREQ-DIO message

An AODV-RPL MoP DIO message containing the RREQ option. The RPLInstanceID in RREQ-DIO is assigned locally by the OrigNode.

RREP-DIO message

An AODV-RPL MoP DIO message containing the RREP option. The RPLInstanceID in RREP-DIO is typically paired to the one in the associated RREQ-DIO message.

Source routing

A mechanism by which the source supplies the complete route towards the target node along with each data packet [RFC6550].

Symmetric route

The upstream and downstream routes traverse the same routers.

TargNode

The IPv6 router (Target Node) for which OrigNode requires a route and initiates Route Discovery within the LLN network.

Upward Direction

The direction from the TargNode to the OrigNode.

Upward Route

A route in the upward direction.

### 3. Overview of AODV-RPL

With AODV-RPL, routes from OrigNode to TargNode within the LLN network established are "on-demand". In other words, the route discovery mechanism in AODV-RPL is invoked reactively when OrigNode has data for delivery to the TargNode but existing routes do not satisfy the application's requirements. The routes discovered by AODV-RPL are not constrained to traverse a common ancestor. Unlike RPL [RFC6550] and P2P-RPL [RFC6997], AODV-RPL can enable asymmetric communication paths in networks with bidirectional asymmetric links. For this purpose, AODV-RPL enables discovery of two routes: namely, one from OrigNode to TargNode, and another from TargNode to OrigNode. When possible, AODV-RPL also enables symmetric route discovery along Paired DODAGs (see Section 5).

In AODV-RPL, routes are discovered by first forming a temporary DAG rooted at the OrigNode. Paired DODAGs (Instances) are constructed according to the AODV-RPL Mode of Operation (MoP) during route formation between the OrigNode and TargNode. The RREQ-Instance is formed by route control messages from OrigNode to TargNode whereas the RREP-Instance is formed by route control messages from TargNode to OrigNode. Intermediate routers join the Paired DODAGs based on the rank as calculated from the DIO message. Henceforth in this document, the RREQ-DIO message means the AODV-RPL mode DIO message from OrigNode to TargNode, containing the RREQ option (see Section 4.1). Similarly, the RREP-DIO message means the AODV-RPL mode DIO message from TargNode to OrigNode, containing the RREP option (see Section 4.2). The route discovered in the RREQ-Instance is used for transmitting data from TargNode to OrigNode, and the route discovered in RREP-Instance is used for transmitting data from OrigNode to TargNode.

#### 4. AODV-RPL DIO Options

#### 4.1. AODV-RPL DIO RREQ Option

OrigNode sets its IPv6 address in the DODAGID field of the RREQ-DIO message. A RREQ-DIO message MUST carry exactly one RREQ option.

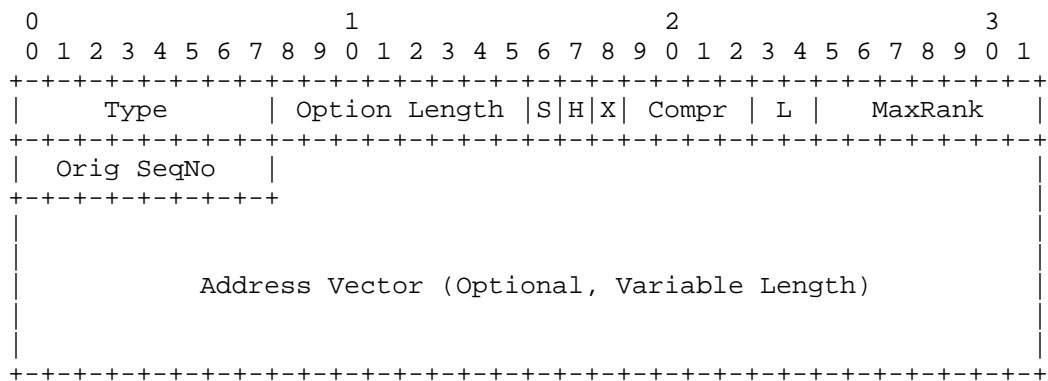


Figure 1: DIO RREQ option format for AODV-RPL MoP

OrigNode supplies the following information in the RREQ option:

Type

The type assigned to the RREQ option (see Section 9.2).

Option Length

The length of the option in octets, excluding the Type and Length fields. Variable due to the presence of the address vector and the number of octets elided according to the Compr value.

S

Symmetric bit indicating a symmetric route from the OrigNode to the router transmitting this RREQ-DIO.

H

Set to one for a hop-by-hop route. Set to zero for a source route. This flag controls both the downstream route and upstream route.

X

Reserved.

Compr

4-bit unsigned integer. Number of prefix octets that are elided from the Address Vector. The octets elided are shared with the

IPv6 address in the DODAGID. This field is only used in source routing mode (H=0). In hop-by-hop mode (H=1), this field MUST be set to zero and ignored upon reception.

#### L

2-bit unsigned integer determining the duration that a node is able to belong to the temporary DAG in RREQ-Instance, including the OrigNode and the TargNode. Once the time is reached, a node MUST leave the DAG and stop sending or receiving any more DIOs for the temporary DODAG. The definition for the "L" bit is similar to that found in [RFC6997], except that the values are adjusted to enable arbitrarily long route lifetime.

- \* 0x00: No time limit imposed.
- \* 0x01: 2 seconds
- \* 0x02: 16 seconds
- \* 0x03: 64 seconds

L is independent from the route lifetime, which is defined in the DODAG configuration option. The route entries in hop-by-hop routing and states of source routing can still be maintained even after the DAG expires.

#### MaxRank

This field indicates the upper limit on the integer portion of the rank (calculated using the DAGRank() macro defined in [RFC6550]). A value of 0 in this field indicates the limit is infinity.

#### Orig SeqNo

Sequence Number of OrigNode, defined similarly as in AODV [RFC3561].

#### Address Vector

A vector of IPv6 addresses representing the route that the RREQ-DIO has passed. It is only present when the 'H' bit is set to 0. The prefix of each address is elided according to the Compr field.

A node MUST NOT join a RREQ instance if its own rank would equal to or higher than MaxRank. Targnode can join the RREQ instance at a rank whose integer portion is equal to the MaxRank. A router MUST discard a received RREQ if the integer part of the advertised rank equals or exceeds the MaxRank limit. This definition of MaxRank is the same as that found in [RFC6997].



## 4.2. AODV-RPL DIO RREP Option

TargNode sets its IPv6 address in the DODAGID field of the RREP-DIO message. A RREP-DIO message MUST carry exactly one RREP option. TargNode supplies the following information in the RREP option:

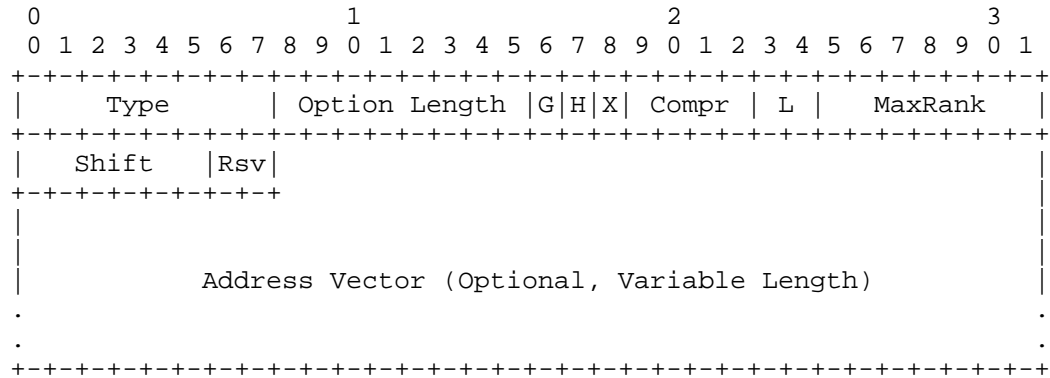


Figure 2: DIO RREP option format for AODV-RPL MoP

**Type**

The type assigned to the RREP option (see Section 9.2)

**Option Length**

The length of the option in octets, excluding the Type and Length fields. Variable due to the presence of the address vector and the number of octets elided according to the Compr value.

**G**

Gratuitous route (see Section 7).

**H**

Requests either source routing (H=0) or hop-by-hop (H=1) for the downstream route. It MUST be set to be the same as the 'H' bit in RREQ option.

**X**

Reserved.

**Compr**

4-bit unsigned integer. Same definition as in RREQ option.

**L**

2-bit unsigned integer defined as in RREQ option.

**MaxRank**

Similarly to MaxRank in the RREQ message, this field indicates the upper limit on the integer portion of the rank. A value of 0 in this field indicates the limit is infinity.

Shift

6-bit unsigned integer. This field is used to recover the original InstanceID (see Section 6.3.3); 0 indicates that the original InstanceID is used.

Rsv

MUST be initialized to zero and ignored upon reception.

Address Vector

Only present when the 'H' bit is set to 0. For an asymmetric route, the Address Vector represents the IPv6 addresses of the route that the RREP-DIO has passed. For a symmetric route, it is the Address Vector when the RREQ-DIO arrives at the TargNode, unchanged during the transmission to the OrigNode.

#### 4.3. AODV-RPL DIO Target Option

The AODV-RPL Target Option is defined based on the Target Option in core RPL [RFC6550]: the Destination Sequence Number of the TargNode is added.

A RREQ-DIO message MUST carry at least one AODV-RPL Target Options.  
A RREP-DIO message MUST carry exactly one AODV-RPL Target Option.

OrigNode can include multiple TargNode addresses via multiple AODV-RPL Target Options in the RREQ-DIO, for routes that share the same constraints. This reduces the cost to building only one DODAG. Furthermore, a single Target Option can be used for different TargNode addresses if they share the same prefix; in that case the use of the destination sequence number is not defined in this document.

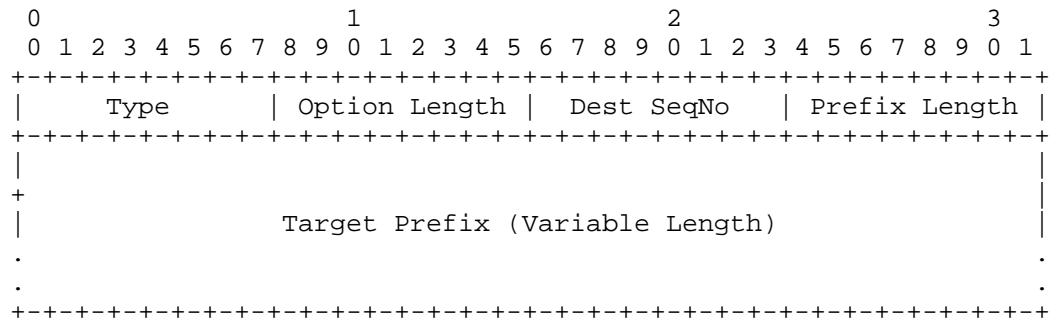


Figure 3: Target option format for AODV-RPL MoP

**Type**

The type assigned to the AODV-RPL Target Option

**Dest SeqNo**

In RREQ-DIO, if nonzero, it is the last known Sequence Number for TargNode for which a route is desired. In RREP-DIO, it is the destination sequence number associated to the route.

**5. Symmetric and Asymmetric Routes**

In Figure 4 and Figure 5, BR is the Border Router, O is the OrigNode, R is an intermediate router, and T is the TargNode. If the RREQ-DIO arrives over an interface that is known to be symmetric, and the 'S' bit is set to 1, then it remains as 1, as illustrated in Figure 4. If an intermediate router sends out RREQ-DIO with the 'S' bit set to 1, then all the one-hop links on the route from the OrigNode O to this router meet the requirements of route discovery, and the route can be used symmetrically.

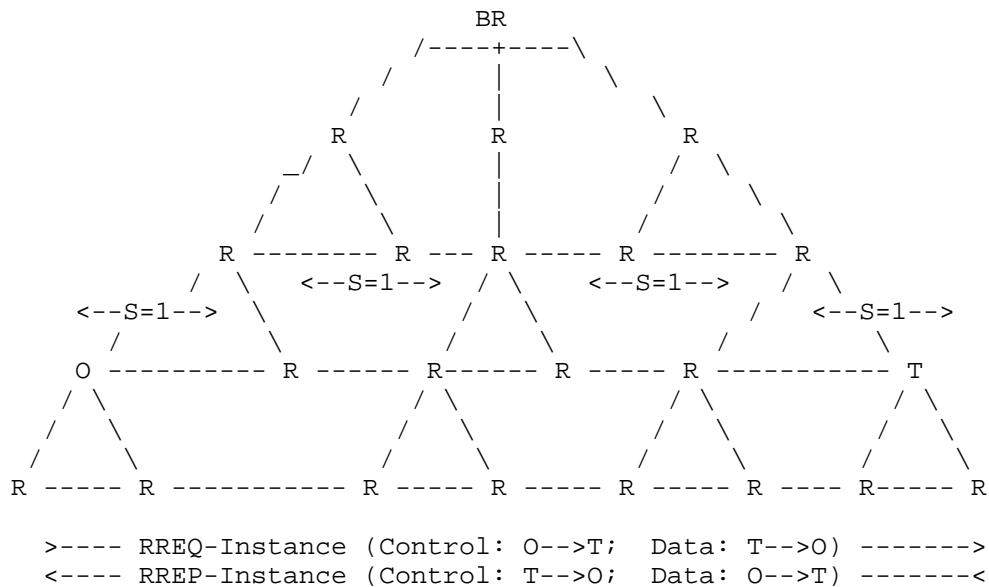


Figure 4: AODV-RPL with Symmetric Paired Instances

Upon receiving a RREQ-DIO with the 'S' bit set to 1, a node determines whether this one-hop link can be used symmetrically, i.e., both the two directions meet the requirements of data transmission. If the RREQ-DIO arrives over an interface that is not known to be symmetric, or is known to be asymmetric, the 'S' bit is set to 0. If the 'S' bit arrives already set to be '0', it is set to be '0' on retransmission (Figure 5). Therefore, for asymmetric route, there is at least one hop which doesn't fulfill the constraints in the two directions. Based on the 'S' bit received in RREQ-DIO, the TargNode T determines whether or not the route is symmetric before transmitting the RREP-DIO message upstream towards the OrigNode O.

The criteria used to determine whether or not each link is symmetric is beyond the scope of the document, and may be implementation-specific. For instance, intermediate routers MAY use local information (e.g., bit rate, bandwidth, number of cells used in 6tisch), a priori knowledge (e.g. link quality according to previous communication) or use averaging techniques as appropriate to the application.

Appendix A describes an example method using the ETX and RSSI to estimate whether the link is symmetric in terms of link quality is given in using an averaging technique.

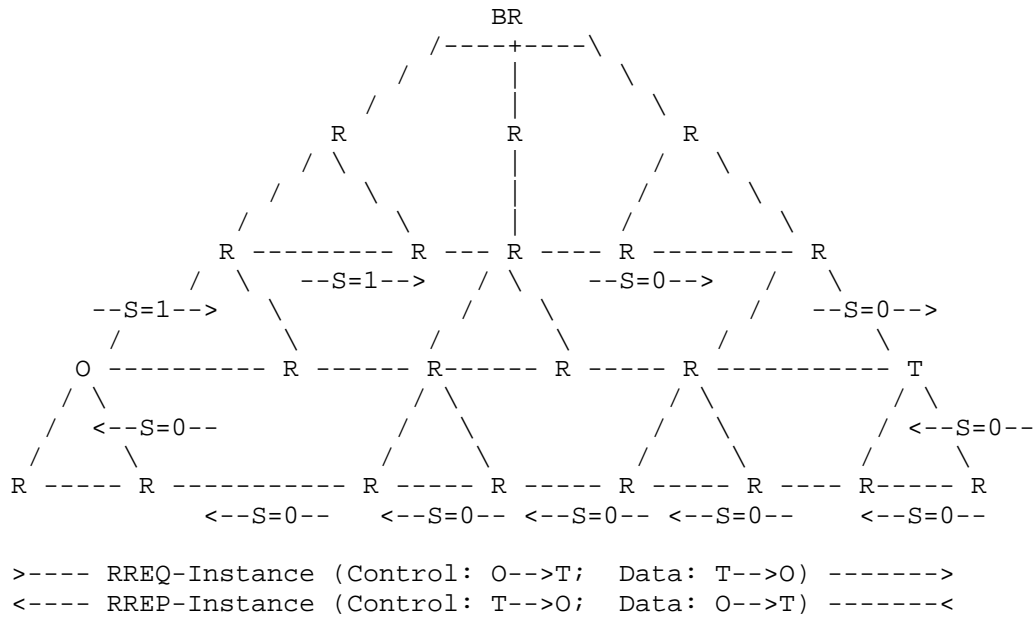


Figure 5: AODV-RPL with Asymmetric Paired Instances

## 6. AODV-RPL Operation

### 6.1. Route Request Generation

The route discovery process is initiated when an application at the OrigNode has data to be transmitted to the TargNode, but does not have a route for the target that fulfills the requirements of the data transmission. In this case, the OrigNode builds a local RPLInstance and a DODAG rooted at itself. Then it transmits a DIO message containing exactly one RREQ option (see Section 4.1) via link-local multicast. The DIO MUST contain at least one AODV-RPL Target Option (see Section 4.3). The 'S' bit in RREQ-DIO sent out by the OrigNode is set to 1.

The OrigNode maintains its Sequence Number as defined in AODV [RFC3561]. Namely, the OrigNode increments its Sequence number each time it initiate a new route discovery operation by transmitting a new RREQ message. Similarly, TargNode increments its Sequence number each time it transmits a RREP message in response to a new RREQ message (one with an incremented Sequence Number for OrigNode).

The address in the AODV-RPL Target Option can be a unicast IPv6 address, or a prefix. The OrigNode can initiate the route discovery process for multiple targets simultaneously by including multiple

AODV-RPL Target Options, and within a RREQ-DIO the requirements for the routes to different TargNodes MUST be the same.

OrigNode can maintain different RPLInstances to discover routes with different requirements to the same targets. Using the InstanceID pairing mechanism (see Section 6.3.3), route replies (RREP-DIOs) for different RPLInstances can be distinguished.

The transmission of RREQ-DIO obeys the Trickle timer. If the duration specified by the "L" bit has elapsed, the OrigNode MUST leave the DODAG and stop sending RREQ-DIOs in the related RPLInstance.

## 6.2. Receiving and Forwarding RREQ messages

### 6.2.1. General Processing

Upon receiving a RREQ-DIO, a router which does not belong to the RREQ-instance goes through the following steps:

#### Step 1:

If the 'S' bit in the received RREQ-DIO is set to 1, the router MUST check the two directions of the link by which the RREQ-DIO is received. In case that the downward (i.e. towards the TargNode) direction of the link can't fulfill the requirements, the link can't be used symmetrically, thus the 'S' bit of the RREQ-DIO to be sent out MUST be set as 0. If the 'S' bit in the received RREQ-DIO is set to 0, the router only checks into the upward direction (towards the OrigNode) of the link.

If the upward direction of the link can fulfill the requirements indicated in the constraint option, and the router's rank would not exceed the MaxRank limit, the router joins the DODAG of the RREQ-Instance. The router that transmitted the received RREQ-DIO is selected as the preferred parent. Later, other RREQ-DIO messages might be received. How to maintain the parent set, select the preferred parent, and update the router's rank obeys the core RPL and the OFs defined in ROLL WG. In case that the constraint or the MaxRank limit is not fulfilled, the router MUST discard the received RREQ-DIO and MUST NOT join the DODAG.

#### Step 2:

Then the router checks if one of its addresses is included in one of the AODV-RPL Target Options. If so, this router is one of the TargNodes. Otherwise, it is an intermediate router.

## Step 3:

If the 'H' bit is set to 1, then the router (TargNode or intermediate) MUST build the upward route entry accordingly. The route entry MUST include at least the following items: Source Address, InstanceID, Destination Address, Next Hop and Lifetime. The Destination Address and the InstanceID can be respectively learned from the DODAGID and the RPLInstanceID of the RREQ-DIO, and the Source Address is copied from the AODV-RPL Target Option. The next hop is the preferred parent. And the lifetime is set according to DODAG configuration and can be extended when the route is actually used.

If the 'H' bit is set to 0, an intermediate router MUST include the address of the interface receiving the RREQ-DIO into the address vector.

## Step 4:

An intermediate router transmits a RREQ-DIO via link-local multicast. TargNode prepares a RREP-DIO.

## 6.2.2. Additional Processing for Multiple Targets

If the OrigNode tries to reach multiple TargNodes in a single RREQ-instance, one of the TargNodes can be an intermediate router to the others, therefore it SHOULD continue sending RREQ-DIO to reach other targets. In this case, before rebroadcasting the RREQ-DIO, a TargNode MUST delete the Target Option encapsulating its own address, so that downstream routers with higher ranks do not try to create a route to this TargetNode.

An intermediate router could receive several RREQ-DIOs from routers with lower ranks in the same RREQ-instance but have different lists of Target Options. When rebroadcasting the RREQ-DIO, the intersection of these lists SHOULD be included. For example, suppose two RREQ-DIOs are received with the same RPLInstance and OrigNode. Suppose further that the first RREQ has (T1, T2) as the targets, and the second one has (T2, T4) as targets. Then only T2 needs to be included in the generated RREQ-DIO. If the intersection is empty, it means that all the targets have been reached, and the router SHOULD NOT send out any RREQ-DIO. Any RREQ-DIO message with different AODV-RPL Target Options coming from a router with higher rank is ignored.

### 6.3. Generating Route Reply (RREP) at TargNode

#### 6.3.1. RREP-DIO for Symmetric route

If a RREQ-DIO arrives at TargNode with the 'S' bit set to 1, there is a symmetric route along which both directions can fulfill the requirements. Other RREQ-DIOs might later provide asymmetric upward routes (i.e. S=0). Selection between a qualified symmetric route and an asymmetric route that might have better performance is implementation-specific and out of scope. If the implementation uses the symmetric route, the TargNode MAY delay transmitting the RREP-DIO for duration RREP\_WAIT\_TIME to await a better symmetric route.

For a symmetric route, the RREP-DIO message is unicast to the next hop according to the accumulated address vector (H=0) or the route entry (H=1). Thus the DODAG in RREP-Instance does not need to be built. The RPLInstanceID in the RREP-Instance is paired as defined in Section 6.3.3. In case the 'H' bit is set to 0, the address vector received in the RREQ-DIO MUST be included in the RREP-DIO. The address of the OrigNode MUST be encapsulated in an AODV-RPL Target Option and included in this RREP-DIO message, and the Dest SeqNo is incremented, as is done in AODV [RFC3561].

#### 6.3.2. RREP-DIO for Asymmetric Route

When a RREQ-DIO arrives at a TargNode with the 'S' bit set to 0, the TargNode MUST build a DODAG in the RREP-Instance rooted at itself in order to discover the downstream route from the OrigNode to the TargNode. The RREP-DIO message MUST be re-transmitted via link-local multicast until the OrigNode is reached or MaxRank is exceeded.

The settings of the fields in RREP option are the same as in symmetric route except for the 'S' bit.

#### 6.3.3. RPLInstanceID Pairing

Since the RPLInstanceID is assigned locally (i.e., there is no coordination between routers in the assignment of RPLInstanceID), the tuple (OrigNode, TargNode, RPLInstanceID) is needed to uniquely identify a discovered route. The upper layer applications may have different requirements and they can initiate the route discoveries simultaneously. Thus between the same pair of OrigNode and TargNode, there can be multiple AODV-RPL instances. To avoid any mismatch, the RREQ-Instance and the RREP-Instance in the same route discovery MUST be paired somehow, e.g. using the RPLInstanceID.

When preparing the RREP-DIO, a TargNode could find the RPLInstanceID to be used for the RREP-Instance is already occupied by another RPL



Instance from an earlier route discovery operation which is still active. In other words, it might happen that two distinct OrigNodes need routes to the same TargNode, and they happen to use the same RPLInstanceID for RREQ-Instance. In this case, the occupied RPLInstanceID MUST NOT be used again. Then the second RPLInstanceID MUST be shifted into another integer so that the two RREP-instances can be distinguished. In RREP option, the Shift field indicates the shift to be applied to original RPLInstanceID. When the new InstanceID after shifting exceeds 63, it rolls over starting at 0. For example, the original InstanceID is 60, and shifted by 6, the new InstanceID will be 2. Related operations can be found in Section 6.4.

#### 6.4. Receiving and Forwarding Route Reply

Upon receiving a RREP-DIO, a router which does not belong to the RREQ-instance goes through the following steps:

##### Step 1:

If the 'S' bit is set to 1, the router proceeds to step 2.

If the 'S' bit of the RREP-DIO is set to 0, the router MUST check the downward direction of the link (towards the TargNode) over which the RREP-DIO is received. If the downward direction of the link can fulfill the requirements indicated in the constraint option, and the router's rank would not exceed the MaxRank limit, the router joins the DODAG of the RREP-Instance. The router that transmitted the received RREP-DIO is selected as the preferred parent. Afterwards, other RREP-DIO messages can be received. How to maintain the parent set, select the preferred parent, and update the router's rank obeys the core RPL and the OFs defined in ROLL WG.

If the constraints are not fulfilled, the router MUST NOT join the DODAG; the router MUST discard the RREQ-DIO, and does not execute the remaining steps in this section.

##### Step 2:

The router next checks if one of its addresses is included in the AODV-RPL Target Option. If so, this router is the OrigNode of the route discovery. Otherwise, it is an intermediate router.

##### Step 3:

If the 'H' bit is set to 1, then the router (OrigNode or intermediate) MUST build a downward route entry. The route entry

SHOULD include at least the following items: OrigNode Address, InstanceID, TargNode Address as destination, Next Hop and Lifetime. For a symmetric route, the next hop in the route entry is the router from which the RREP-DIO is received. For an asymmetric route, the next hop is the preferred parent in the DODAG of RREQ-Instance. The InstanceID in the route entry MUST be the original RPLInstanceID (after subtracting the Shift field value). The source address is learned from the AODV-RPL Target Option, and the destination address is learned from the DODAGID. The lifetime is set according to DODAG configuration and can be extended when the route is actually used.

If the 'H' bit is set to 0, for an asymmetric route, an intermediate router MUST include the address of the interface receiving the RREP-DIO into the address vector; for a symmetric route, there is nothing to do in this step.

#### Step 4:

If the receiver is the OrigNode, it can start transmitting the application data to TargNode along the path as provided in RREP-Instance, and processing for the RREP-DIO is complete. Otherwise, in case of an asymmetric route, the intermediate router transmits the RREP-DIO via link-local multicast. In case of a symmetric route, the RREP-DIO message is unicast to the next hop according to the address vector in the RREP-DIO (H=0) or the local route entry (H=1). The RPLInstanceID in the transmitted RREP-DIO is the same as the value in the received RREP-DIO.

### 7. Gratuitous RREP

In some cases, an Intermediate router that receives a RREQ-DIO message MAY transmit a "Gratuitous" RREP-DIO message back to OrigNode instead of continuing to multicast the RREQ-DIO towards TargNode. The intermediate router effectively builds the RREP-Instance on behalf of the actual TargNode. The 'G' bit of the RREP option is provided to distinguish the Gratuitous RREP-DIO (G=1) sent by the Intermediate node from the RREP-DIO sent by TargNode (G=0).

The gratuitous RREP-DIO can be sent out when an intermediate router R receives a RREQ-DIO for a TargNode T, and R happens to have both downward and upward routes to T which also fulfill the requirements.

In case of source routing, the intermediate router R MUST unicast the received RREQ-DIO to TargNode T including the address vector between the OrigNode O and the router R. Thus T can have a complete upward route address vector from itself to O. Then R MUST send out the gratuitous RREP-DIO including the address vector from R to T.

In case of hop-by-hop routing, R MUST unicast the received RREQ-DIO to T. The routers along the route SHOULD build new route entries with the related RPLInstanceID and DODAGID in the downward direction. Then T MUST unicast the RREP-DIO to R, and the routers along the route SHOULD build new route entries in the upward direction. Upon received the unicast RREP-DIO, R sends the gratuitous RREP-DIO to the OrigNode as the same way defined in Section 6.3.

## 8. Operation of Trickle Timer

The trickle timer operation to control RREQ-Instance/RREP-Instance multicast is similar to that in P2P-RPL [RFC6997].

## 9. IANA Considerations

### 9.1. New Mode of Operation: AODV-RPL

IANA is required to assign a new Mode of Operation, named "AODV-RPL" for Point-to-Point(P2P) hop-by-hop routing under the RPL registry. The value of TBD1 is assigned from the "Mode of Operation" space [RFC6550].

Value	Description	Reference
TBD1 (5)	AODV-RPL	This document

Figure 6: Mode of Operation

### 9.2. AODV-RPL Options: RREQ, RREP, and Target

Three entries are required for new AODV-RPL options "RREQ", "RREP" and "AODV-RPL Target" with values of TBD2 (0x0A), TBD3 (0x0B) and TBD4 (0x0C) from the "RPL Control Message Options" space [RFC6550].

Value	Meaning	Reference
TBD2 (0x0A)	RREQ Option	This document
TBD3 (0x0B)	RREP Option	This document
TBD3 (0x0C)	AODV-RPL Target Option	This document

Figure 7: AODV-RPL Options

## 10. Security Considerations

This document does not introduce additional security issues compared to base RPL. For general RPL security considerations, see [RFC6550].

## 11. Future Work

There has been some discussion about how to determine the initial state of a link after an AODV-RPL-based network has begun operation. The current draft operates as if the links are symmetric until additional metric information is collected. The means for making link metric information is considered out of scope for AODV-RPL. In the future, RREQ and RREP messages could be equipped with new fields for use in verifying link metrics. In particular, it is possible to identify unidirectional links; an RREQ received across a unidirectional link has to be dropped, since the destination node cannot make use of the received DODAG to route packets back to the source node that originated the route discovery operation. This is roughly the same as considering a unidirectional link to present an infinite cost metric that automatically disqualifies it for use in the reverse direction.

## 12. References

### 12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3561] Perkins, C., Belding-Royer, E., and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", RFC 3561, DOI 10.17487/RFC3561, July 2003, <<https://www.rfc-editor.org/info/rfc3561>>.
- [RFC5548] Dohler, M., Ed., Watteyne, T., Ed., Winter, T., Ed., and D. Barthel, Ed., "Routing Requirements for Urban Low-Power and Lossy Networks", RFC 5548, DOI 10.17487/RFC5548, May 2009, <<https://www.rfc-editor.org/info/rfc5548>>.
- [RFC5673] Pister, K., Ed., Thubert, P., Ed., Dwars, S., and T. Phinney, "Industrial Routing Requirements in Low-Power and Lossy Networks", RFC 5673, DOI 10.17487/RFC5673, October 2009, <<https://www.rfc-editor.org/info/rfc5673>>.

- [RFC5826] Brandt, A., Buron, J., and G. Porcu, "Home Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5826, DOI 10.17487/RFC5826, April 2010, <<https://www.rfc-editor.org/info/rfc5826>>.
- [RFC5867] Martocci, J., Ed., De Mil, P., Riou, N., and W. Vermeulen, "Building Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5867, DOI 10.17487/RFC5867, June 2010, <<https://www.rfc-editor.org/info/rfc5867>>.
- [RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <<https://www.rfc-editor.org/info/rfc6550>>.
- [RFC6552] Thubert, P., Ed., "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6552, DOI 10.17487/RFC6552, March 2012, <<https://www.rfc-editor.org/info/rfc6552>>.
- [RFC6998] Goyal, M., Ed., Baccelli, E., Brandt, A., and J. Martocci, "A Mechanism to Measure the Routing Metrics along a Point-to-Point Route in a Low-Power and Lossy Network", RFC 6998, DOI 10.17487/RFC6998, August 2013, <<https://www.rfc-editor.org/info/rfc6998>>.

## 12.2. Informative References

- [I-D.thubert-roll-asymlink]  
Thubert, P., "RPL adaptation for asymmetrical links", draft-thubert-roll-asymlink-02 (work in progress), December 2011.
- [RFC6997] Goyal, M., Ed., Baccelli, E., Philipp, M., Brandt, A., and J. Martocci, "Reactive Discovery of Point-to-Point Routes in Low-Power and Lossy Networks", RFC 6997, DOI 10.17487/RFC6997, August 2013, <<https://www.rfc-editor.org/info/rfc6997>>.

## Appendix A. Example: ETX/RSSI Values to select S bit

We have tested the combination of "RSSI(downstream)" and "ETX (upstream)" to determine whether the link is symmetric or asymmetric at the intermediate nodes. The example of how the ETX and RSSI values are used in conjunction is explained below:

Source----->NodeA----->NodeB----->Destination

Figure 8: Communication link from Source to Destination

RSSI at NodeA for NodeB	Expected ETX at NodeA for NodeB->NodeA
> -15	150
-25 to -15	192
-35 to -25	226
-45 to -35	662
-55 to -45	993

Table 1: Selection of 'S' bit based on Expected ETX value

We tested the operations in this specification by making the following experiment, using the above parameters. In our experiment, a communication link is considered as symmetric if the ETX value of NodeA->NodeB and NodeB->NodeA (See Figure.8) are, say, within 1:3 ratio. This ratio should be taken as a notional metric for deciding link symmetric/asymmetric nature, and precise definition of the ratio is beyond the scope of the draft. In general, NodeA can only know the ETX value in the direction of NodeA -> NodeB but it has no direct way of knowing the value of ETX from NodeB->NodeA. Using physical testbed experiments and realistic wireless channel propagation models, one can determine a relationship between RSSI and ETX representable as an expression or a mapping table. Such a relationship in turn can be used to estimate ETX value at nodeA for link NodeB--->NodeA from the received RSSI from NodeB. Whenever nodeA determines that the link towards the nodeB is bi-directional asymmetric then the "S" bit is set to "S=0". Later on, the link from NodeA to Destination is asymmetric with "S" bit remains to "0".

## Appendix B. Changelog

### B.1. Changes to version 02

- o Include the support for source routing.
- o Import some features from [RFC6997], e.g., choice between hop-by-hop and source routing, the "L" bit which determines the duration of residence in the DAG, MaxRank, etc.
- o Define new target option for AODV-RPL, including the Destination Sequence Number in it. Move the TargNode address in RREQ option and the OrigNode address in RREP option into ADOV-RPL Target Option.

- o Support route discovery for multiple targets in one RREQ-DIO.
- o New InstanceID pairing mechanism.

#### B.2. Changes to version 03

- o Updated RREP option format. Remove the 'T' bit in RREP option.
- o Using the same RPLInstanceID for RREQ and RREP, no need to update [RFC6550].
- o Explanation of Shift field in RREP.
- o Multiple target options handling during transmission.

#### Authors' Addresses

Satish Anamalamudi  
SRM University-AP  
Amaravati Campus  
Amaravati, Andhra Pradesh 522 502  
India

Email: satishnaidu80@gmail.com

Mingui Zhang  
Huawei Technologies  
No. 156 Beiqing Rd. Haidian District  
Beijing 100095  
China

Email: zhangmingui@huawei.com

Abdur Rashid Sangi  
Huaiyin Institute of Technology  
No.89 North Beijing Road, Qinghe District  
Huaian 223001  
P.R. China

Email: sangi\_bahrian@yahoo.com

Charles E. Perkins  
Futurewei  
2330 Central Expressway  
Santa Clara 95050  
Unites States

Email: charliep@computer.org

S.V.R Anand  
Indian Institute of Science  
Bangalore 560012  
India

Email: anand@ece.iisc.ernet.in

Bing Liu  
Huawei Technologies  
No. 156 Beiqing Rd. Haidian District  
Beijing 100095  
China

Email: remy.liubing@huawei.com



ROLL  
Internet-Draft  
Intended status: Standards Track  
Expires: 8 September 2022

C.E. Perkins  
Lupin Lodge  
S.V.R.Anand  
Indian Institute of Science  
S. Anamalamudi  
SRM University-AP  
B. Liu  
Huawei Technologies  
7 March 2022

Supporting Asymmetric Links in Low Power Networks: AODV-RPL  
draft-ietf-roll-aodv-rpl-13

## Abstract

Route discovery for symmetric and asymmetric Peer-to-Peer (P2P) traffic flows is a desirable feature in Low power and Lossy Networks (LLNs). For that purpose, this document specifies a reactive P2P route discovery mechanism for both hop-by-hop routing and source routing: Ad Hoc On-demand Distance Vector Routing (AODV) based RPL protocol (AODV-RPL). Paired Instances are used to construct directional paths, for cases where there are asymmetric links between source and target nodes.

## Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://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 8 September 2022.

## Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	4
3. Overview of AODV-RPL . . . . .	6
4. AODV-RPL DIO Options . . . . .	7
4.1. AODV-RPL RREQ Option . . . . .	7
4.2. AODV-RPL RREP Option . . . . .	9
4.3. AODV-RPL Target Option . . . . .	10
5. Symmetric and Asymmetric Routes . . . . .	12
6. AODV-RPL Operation . . . . .	14
6.1. Route Request Generation . . . . .	14
6.2. Receiving and Forwarding RREQ messages . . . . .	14
6.2.1. Step 1: RREQ reception and evaluation . . . . .	15
6.2.2. Step 2: TargNode and Intermediate Router determination . . . . .	15
6.2.3. Step 3: Intermediate Router RREQ processing . . . . .	16
6.2.4. Step 4: Symmetric Route Processing at an Intermediate Router . . . . .	16
6.2.5. Step 5: RREQ propagation at an Intermediate Router . . . . .	17
6.2.6. Step 6: RREQ reception at TargNode . . . . .	17
6.3. Generating Route Reply (RREP) at TargNode . . . . .	17
6.3.1. RREP-DIO for Symmetric route . . . . .	18
6.3.2. RREP-DIO for Asymmetric Route . . . . .	18
6.3.3. RPLInstanceID Pairing . . . . .	18
6.4. Receiving and Forwarding Route Reply . . . . .	19
6.4.1. Step 1: Receiving and Evaluation . . . . .	19
6.4.2. Step 2: OrigNode or Intermediate Router . . . . .	19
6.4.3. Step 3: Build Route to TargNode . . . . .	20
6.4.4. Step 4: RREP Propagation . . . . .	20
7. Gratuitous RREP . . . . .	20
8. Operation of Trickle Timer . . . . .	21
9. IANA Considerations . . . . .	21
10. Security Considerations . . . . .	22
11. Acknowledgements . . . . .	23
12. References . . . . .	23
12.1. Normative References . . . . .	23
12.2. Informative References . . . . .	24

Appendix A. Example: Using ETX/RSSI Values to determine value of S	
bit . . . . .	25
Appendix B. Changelog . . . . .	27
B.1. Changes from version 12 to version 13 . . . . .	27
B.2. Changes from version 11 to version 12 . . . . .	28
B.3. Changes from version 10 to version 11 . . . . .	28
B.4. Changes from version 09 to version 10 . . . . .	29
B.5. Changes from version 08 to version 09 . . . . .	30
B.6. Changes from version 07 to version 08 . . . . .	30
B.7. Changes from version 06 to version 07 . . . . .	31
B.8. Changes from version 05 to version 06 . . . . .	31
B.9. Changes from version 04 to version 05 . . . . .	31
B.10. Changes from version 03 to version 04 . . . . .	32
B.11. Changes from version 02 to version 03 . . . . .	32
Appendix C. Contributors . . . . .	32
Authors' Addresses . . . . .	33

## 1. Introduction

Routing Protocol for Low-Power and Lossy Networks (RPL) [RFC6550] is an IPv6 distance vector routing protocol designed to support multiple traffic flows through a root-based Destination-Oriented Directed Acyclic Graph (DODAG). Typically, a router does not have routing information for most other routers. Consequently, for traffic between routers within the DODAG (i.e., Peer-to-Peer (P2P) traffic) data packets either have to traverse the root in non-storing mode, or traverse a common ancestor in storing mode. Such P2P traffic is thereby likely to traverse longer routes and may suffer severe congestion near the root (for more information see [RFC6997], [RFC6998]). The network environment that is considered in this document is assumed to be the same as described in Section 1 of [RFC6550].

The route discovery process in AODV-RPL is modeled on the analogous procedure specified in AODV [RFC3561]. The on-demand nature of AODV route discovery is natural for the needs of peer-to-peer routing in RPL-based LLNs. AODV terminology has been adapted for use with AODV-RPL messages, namely RREQ for Route Request, and RREP for Route Reply. AODV-RPL currently omits some features compared to AODV -- in particular, flagging Route Errors, "blacklisting" unidirectional links ([RFC3561]), multihoming, and handling unnumbered interfaces.

AODV-RPL reuses and extends the core RPL functionality to support routes with bidirectional asymmetric links. It retains RPL's DODAG formation, RPL Instance and the associated Objective Function (defined in [RFC6551]), trickle timers, and support for storing and non-storing modes. AODV-RPL adds basic messages RREQ and RREP as part of RPL DODAG Information Object (DIO) control message, which go

in separate (paired) RPL instances. AODV-RPL does not utilize the Destination Advertisement Object (DAO) control message of RPL. AODV-RPL uses the "P2P Route Discovery Mode of Operation" (MOP == 4) with three new Options for the DIO message, dedicated to discover P2P routes. These P2P routes may differ from routes discoverable by native RPL. Since AODV-RPL uses newly defined Options, there is no conflict with P2P-RPL [RFC6997], a previous document using the same MOP. AODV-RPL can be operated whether or not P2P-RPL or native RPL is running otherwise. For many networks AODV-RPL could be a replacement for RPL, since it can find better routes at very moderate extra cost. Consequently, it is unlikely that RPL would be needed in a network that is running AODV-RPL, even though it would be possible to run both protocols at the same time.

## 2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

AODV-RPL reuses names for messages and data structures, including Rank, DODAG and DODAGID, as defined in RPL [RFC6550].

### AODV

Ad Hoc On-demand Distance Vector Routing [RFC3561].

### Asymmetric Route

The route from the OrigNode to the TargNode can traverse different nodes than the route from the TargNode to the OrigNode. An asymmetric route may result from the asymmetry of links, such that only one direction of the series of links satisfies the Objective Function during route discovery.

### Bi-directional Asymmetric Link

A link that can be used in both directions but with different link characteristics.

### DIO

DODAG Information Object

### DODAG RREQ-Instance (or simply RREQ-Instance)

RPL Instance built using the DIO with RREQ option; used for transmission of control messages from OrigNode to TargNode, thus enabling data transmission from TargNode to OrigNode.

DODAG RREP-Instance (or simply RREP-Instance)

RPL Instance built using the DIO with RREP option; used for transmission of control messages from TargNode to OrigNode thus enabling data transmission from OrigNode to TargNode.

Downward Direction

The direction from the OrigNode to the TargNode.

Downward Route

A route in the downward direction.

hop-by-hop routing

Routing when each router stores routing information about the next hop.

on-demand routing

Routing in which a route is established only when needed.

OrigNode

The IPv6 router (Originating Node) initiating the AODV-RPL route discovery to obtain a route to TargNode.

Paired DODAGs

Two DODAGs for a single route discovery process between OrigNode and TargNode.

P2P

Peer-to-Peer -- in other words, not constrained a priori to traverse a common ancestor.

reactive routing

Same as "on-demand" routing.

RREQ-DIO message

A DIO message containing the RREQ option. The RPLInstanceID in RREQ-DIO is assigned locally by the OrigNode. The RREQ-DIO message has a secure variant as noted in [RFC6550].

RREQ-InstanceID

The RPLInstanceID for the RREQ-Instance. This term is used to indicate the value of the RPLInstanceID as provided by OrigNode in the RREQ message. The RPLInstanceID in the RREP message along with the Delta value determines the associated RREQ-InstanceID.

**RREP-DIO message**

A DIO message containing the RREP option. OrigNode pairs the RPLInstanceID in RREP-DIO to the one in the associated RREQ-DIO message (i.e., the RREQ-InstanceID) as described in Section 6.3.2. The RREP-DIO message has a secure variant as noted in [RFC6550].

**Source routing**

A mechanism by which the source supplies the complete route towards the target node along with each data packet [RFC6550].

**Symmetric route**

The upstream and downstream routes traverse the same routers and over the same links.

**TargNode**

The IPv6 router (Target Node) for which OrigNode requires a route and initiates Route Discovery within the LLN network.

**Upward Direction**

The direction from the TargNode to the OrigNode.

**Upward Route**

A route in the upward direction.

**ART option**

AODV-RPL Target option: a target option defined in this document.

### 3. Overview of AODV-RPL

With AODV-RPL, routes from OrigNode to TargNode within the LLN network are established "on-demand". In other words, the route discovery mechanism in AODV-RPL is invoked reactively when OrigNode has data for delivery to the TargNode but existing routes do not satisfy the application's requirements. AODV-RPL works without requiring the use of RPL or any other routing protocol.

The routes discovered by AODV-RPL are not constrained to traverse a common ancestor. AODV-RPL can enable asymmetric communication paths in networks with bidirectional asymmetric links. For this purpose, AODV-RPL enables discovery of two routes: namely, one from OrigNode to TargNode, and another from TargNode to OrigNode. AODV-RPL also enables discovery of symmetric routes along Paired DODAGs, when symmetric routes are possible (see Section 5).

In AODV-RPL, routes are discovered by first forming a temporary DAG rooted at the OrigNode. Paired DODAGs (Instances) are constructed during route formation between the OrigNode and TargNode. The RREQ-Instance is formed by route control messages from OrigNode to

TargNode whereas the RREP-Instance is formed by route control messages from TargNode to OrigNode. Intermediate routers join the DODAGs based on the Rank [RFC6550] as calculated from the DIO message. Henceforth in this document, "RREQ-DIO message" means the DIO message from OrigNode toward TargNode, containing the RREQ option as specified in Section 4.1. Similarly, "RREP-DIO message" means the DIO message from TargNode toward OrigNode, containing the RREP option as specified in Section 4.2. The route discovered in the RREQ-Instance is used for transmitting data from TargNode to OrigNode, and the route discovered in RREP-Instance is used for transmitting data from OrigNode to TargNode.

#### 4. AODV-RPL DIO Options

#### 4.1. AODV-RPL RREQ Option

OrigNode selects one of its IPv6 addresses and sets it in the DODAGID field of the RREQ-DIO message. Exactly one RREQ option **MUST** be present in a RREQ-DIO message, otherwise the message **MUST** be dropped.

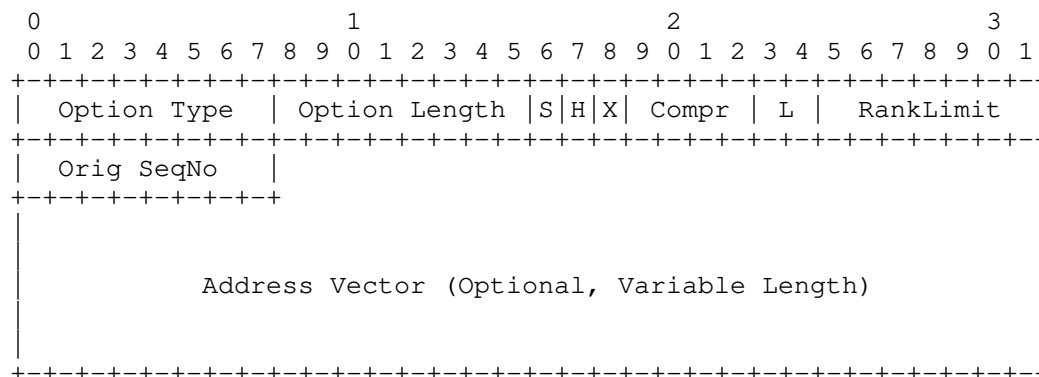


Figure 1: Format for AODV-RPL RREQ Option

OrigNode supplies the following information in the RREQ option:

Option Type  
TBD2

Option	Length
1	10
2	10
3	10
4	10
5	10
6	10
7	10
8	10
9	10
10	10
11	10
12	10
13	10
14	10
15	10
16	10
17	10
18	10
19	10
20	10
21	10
22	10
23	10
24	10
25	10
26	10
27	10
28	10
29	10
30	10
31	10
32	10
33	10
34	10
35	10
36	10
37	10
38	10
39	10
40	10
41	10
42	10
43	10
44	10
45	10
46	10
47	10
48	10
49	10
50	10
51	10
52	10
53	10
54	10
55	10
56	10
57	10
58	10
59	10
60	10
61	10
62	10
63	10
64	10
65	10
66	10
67	10
68	10
69	10
70	10
71	10
72	10
73	10
74	10
75	10
76	10
77	10
78	10
79	10
80	10
81	10
82	10
83	10
84	10
85	10
86	10
87	10
88	10
89	10
90	10
91	10
92	10
93	10
94	10
95	10
96	10
97	10
98	10
99	10
100	10

The length of the option in octets, excluding the Type and Length fields. Variable due to the presence of the address vector and the number of octets elided according to the Compr value.

- S**  
Symmetric bit indicating a symmetric route from the OrigNode to the router transmitting this RREQ-DIO. See Section 5.
- H**  
Set to one for a hop-by-hop route. Set to zero for a source route. This flag controls both the downstream route and upstream route.
- X**  
Reserved; MUST be initialized to zero and ignored upon reception.
- Compr**  
4-bit unsigned integer. When Compr is nonzero, exactly that number of prefix octets MUST be elided from each address before storing it in the Address Vector. The octets elided are shared with the IPv6 address in the DODAGID. This field is only used in source routing mode (H=0). In hop-by-hop mode (H=1), this field MUST be set to zero and ignored upon reception.
- L**  
2-bit unsigned integer determining the length of time that a node is able to belong to the RREQ-Instance (a temporary DAG including the OrigNode and the TargNode). Once the time is reached, a node MUST leave the RREQ-Instance and stop sending or receiving any more DIOs for the RREQ-Instance. This naturally depends on the node's ability to keep track of time. Once a node leaves an RREQ-Instance, it MUST NOT rejoin the same RREQ-Instance. L is independent from the route lifetime, which is defined in the DODAG configuration option.
- \* 0x00: No time limit imposed.
  - \* 0x01: 16 seconds
  - \* 0x02: 64 seconds
  - \* 0x03: 256 seconds
- RankLimit**  
This field indicates the upper limit on the integer portion of the Rank (calculated using the DAGRank() macro defined in [RFC6550]). A value of 0 in this field indicates the limit is infinity.
- Orig SeqNo**  
Sequence Number of OrigNode. See Section 6.1.



#### Address Vector

A vector of IPv6 addresses representing the route that the RREQ-DIO has passed. It is only present when the H bit is set to 0. The prefix of each address is elided according to the Compr field.

TargNode can join the RREQ instance at a Rank whose integer portion is less than or equal to the RankLimit. Any other node **MUST NOT** join a RREQ instance if its own Rank would be equal to or higher than RankLimit. A router **MUST** discard a received RREQ if the integer part of the advertised Rank equals or exceeds the RankLimit.

#### 4.2. AODV-RPL RREP Option

TargNode sets one of its IPv6 addresses in the DODAGID field of the RREP-DIO message. Exactly one RREP option **MUST** be present in a RREP-DIO message, otherwise the message **MUST** be dropped. TargNode supplies the following information in the RREP option:

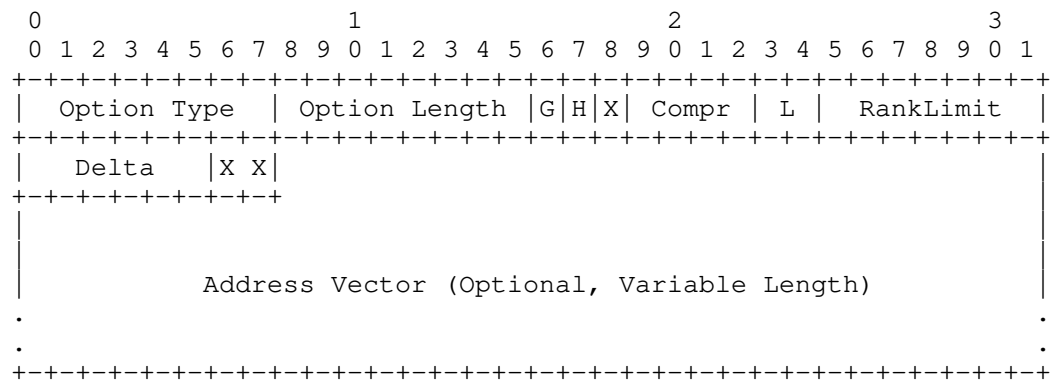


Figure 2: Format for AODV-RPL RREP option

#### Option Type

TBD3

#### Option Length

The length of the option in octets, excluding the Type and Length fields. Variable due to the presence of the address vector and the number of octets elided according to the Compr value.

#### G

Gratuitous route (see Section 7).

- H  
The H bit in the RREP option MUST be set to be the same as the H bit in RREQ option. It requests either source routing (H=0) or hop-by-hop (H=1) for the downstream route.
- X  
Reserved; MUST be initialized to zero and ignored upon reception.
- Compr  
4-bit unsigned integer. Same definition as in RREQ option.
- L  
2-bit unsigned integer defined as in RREQ option. The lifetime of the RREP-Instance MUST be shorter than the lifetime of the RREQ-Instance to which it is paired.
- RankLimit  
Similarly to RankLimit in the RREQ message, this field indicates the upper limit on the integer portion of the Rank. A value of 0 in this field indicates the limit is infinity.
- Delta  
6-bit unsigned integer. This field is used to recover the RREQ-InstanceID (see Section 6.3.3); 0 indicates that the RREQ-InstanceID has the same value as the RPLInstanceID of the RREP message.
- X X  
Reserved; MUST be initialized to zero and ignored upon reception.
- Address Vector  
Only present when the H bit is set to 0. For an asymmetric route, the Address Vector represents the IPv6 addresses of the path through the network the RREP-DIO has passed. For a symmetric route, it is the Address Vector when the RREQ-DIO arrives at the TargNode, unchanged during the transmission to the OrigNode.

#### 4.3. AODV-RPL Target Option

The AODV-RPL Target (ART) Option is based on the Target Option in core RPL [RFC6550]. The Flags field is replaced by the Destination Sequence Number of the TargNode and the Prefix Length field is reduced to 7 bits so that the value is limited to be no greater than 127.

A RREQ-DIO message MUST carry at least one ART Option. A RREP-DIO message MUST carry exactly one ART Option. Otherwise, the message MUST be dropped.

OrigNode can include multiple TargNode addresses via multiple AODV-RPL Target Options in the RREQ-DIO, for routes that share the same requirement on metrics. This reduces the cost to building only one DODAG.

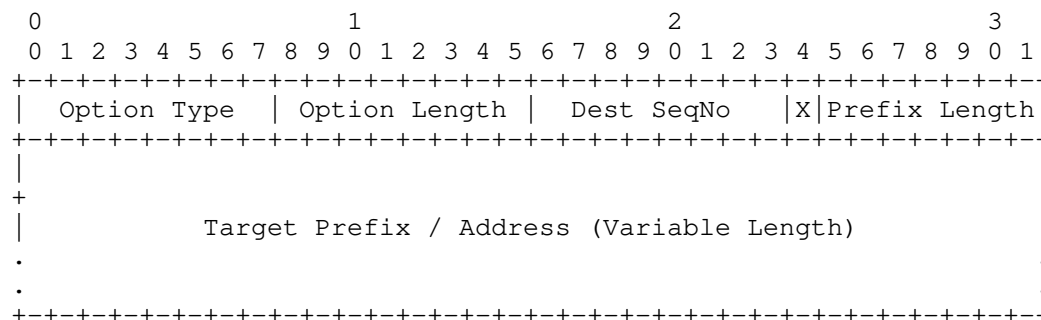


Figure 3: ART Option format for AODV-RPL

Option Type

TBD4

Option Length

Length of the option in octets excluding the Type and Length fields.

Dest SeqNo

In RREQ-DIO, if nonzero, it is the Sequence Number for the last route that OrigNode stored to the TargNode for which a route is desired. In RREP-DIO, it is the destination sequence number associated to the route. Zero is used if there is no known information about the sequence number of TargNode, and not used otherwise.

X

A one-bit reserved field. This field MUST be initialized to zero by the sender and MUST be ignored by the receiver.

Prefix Length

7-bit unsigned integer. Number of valid leading bits in the IPv6 Prefix. If Prefix Length is 0, then the value in the Target Prefix / Address field represents an IPv6 address, not a prefix.

Target Prefix / Address

(variable-length field) An IPv6 destination address or prefix. The Prefix Length field contains the number of valid leading bits

in the prefix. The Target Prefix / Address field contains the least number of octets that can represent all of the bits of the Prefix, in other words  $\text{Ceil}(\text{Prefix Length}/8)$  octets. The initial bits in the Target Prefix / Address field preceding the prefix length (if any) MUST be set to zero on transmission and MUST be ignored on receipt. If Prefix Length is zero, the Address field is 128 bits for IPv6 addresses.

## 5. Symmetric and Asymmetric Routes

Links are considered symmetric until indication to the contrary is received. In Figure 4 and Figure 5, BR is the Border Router, O is the OrigNode, each R is an intermediate router, and T is the TargNode. If the RREQ-DIO arrives over an interface that is known to be symmetric, and the S bit is set to 1, then it remains as 1, as illustrated in Figure 4. If an intermediate router sends out RREQ-DIO with the S bit set to 1, then each link en route from the OrigNode O to this router has met the requirements of route discovery, and the route can be used symmetrically.

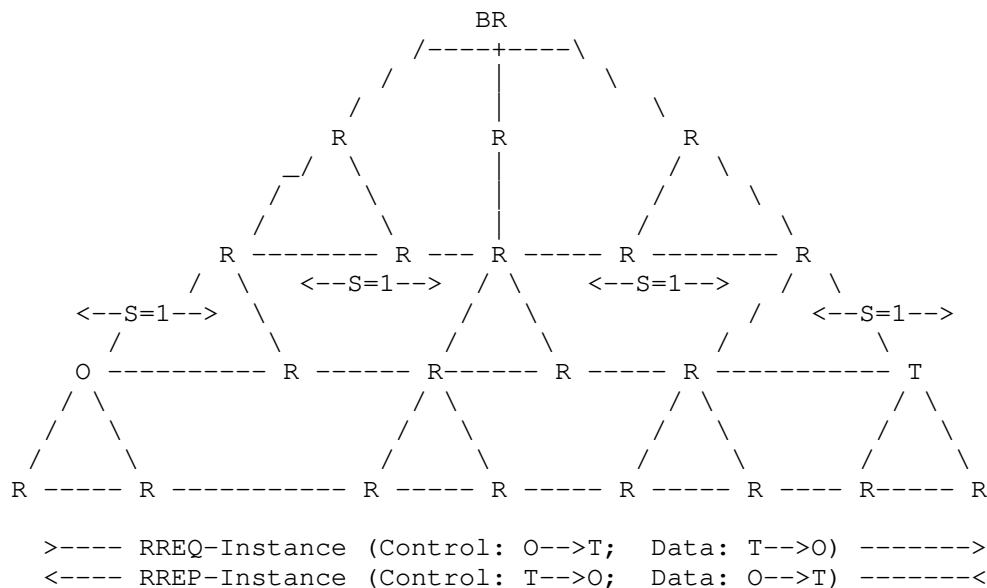


Figure 4: AODV-RPL with Symmetric Instances

Upon receiving a RREQ-DIO with the S bit set to 1, a node determines whether this link can be used symmetrically, i.e., both directions meet the requirements of data transmission. If the RREQ-DIO arrives over an interface that is not known to be symmetric, or is known to be asymmetric, the S bit is set to 0. If the S bit arrives already

set to be '0', it is set to be '0' when the RREQ-DIO is propagated (Figure 5). For an asymmetric route, there is at least one hop which doesn't satisfy the Objective Function. Based on the S bit received in RREQ-DIO, TargNode T determines whether or not the route is symmetric before transmitting the RREP-DIO message upstream towards the OrigNode O.

It is beyond the scope of this document to specify the criteria used when determining whether or not each link is symmetric. As an example, intermediate routers can use local information (e.g., bit rate, bandwidth, number of cells used in 6tisch [RFC9030]), a priori knowledge (e.g., link quality according to previous communication) or use averaging techniques as appropriate to the application. Other link metric information can be acquired before AODV-RPL operation, by executing evaluation procedures; for instance test traffic can be generated between nodes of the deployed network. During AODV-RPL operation, OAM techniques for evaluating link state (see [RFC7548], [RFC7276], [co-ioam]) MAY be used (at regular intervals appropriate for the LLN). The evaluation procedures are out of scope for AODV-RPL.

Appendix A describes an example method using the upstream Expected Number of Transmissions (ETX) and downstream Received Signal Strength Indicator (RSSI) to estimate whether the link is symmetric in terms of link quality using an averaging technique.

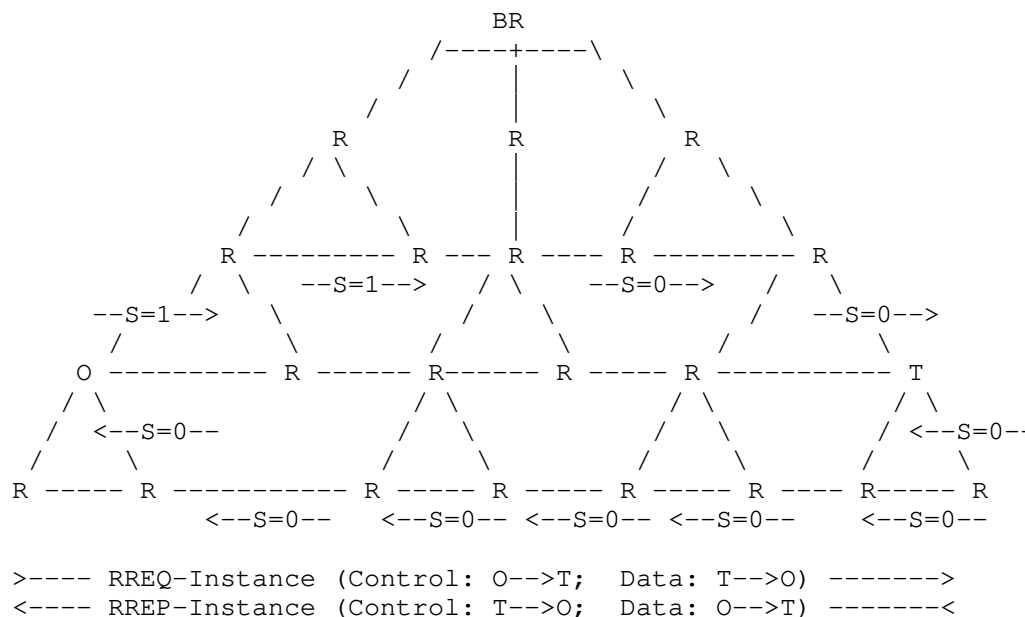


Figure 5: AODV-RPL with Asymmetric Paired Instances

As illustrated in Figure 5, an intermediate router determines the S bit value that the RREQ-DIO should carry using link asymmetry detection methods as discussed earlier in this section. In many cases the intermediate router has already made the link asymmetry decision by the time RREQ-DIO arrives.

## 6. AODV-RPL Operation

### 6.1. Route Request Generation

The route discovery process is initiated when an application at the OrigNode has data to be transmitted to the TargNode, but does not have a route that satisfies the Objective Function for the target of the application's data. In this case, the OrigNode builds a local RPLInstance and a DODAG rooted at itself. Then it transmits a DIO message containing exactly one RREQ option (see Section 4.1) to multicast group all-RPL-nodes. The DIO MUST contain at least one ART Option (see Section 4.3), which indicates the TargNode. The S bit in RREQ-DIO sent out by the OrigNode is set to 1.

Each node maintains a sequence number; the operation is specified in section 7.2 of [RFC6550]. When the OrigNode initiates a route discovery process, it MUST increase its own sequence number to avoid conflicts with previously established routes. The sequence number is carried in the Orig SeqNo field of the RREQ option.

The address in the ART Option can be a unicast IPv6 address or a prefix. The OrigNode can initiate the route discovery process for multiple targets simultaneously by including multiple ART Options. Within a RREQ-DIO the Objective Function for the routes to different TargNodes MUST be the same.

OrigNode can maintain different RPLInstances to discover routes with different requirements to the same targets. Using the RPLInstanceID pairing mechanism (see Section 6.3.3), route replies (RREP-DIOs) for different RPLInstances can be generated.

The transmission of RREQ-DIO obeys the Trickle timer [RFC6206]. If the length of time specified by the L field has elapsed, the OrigNode MUST leave the DODAG and stop sending RREQ-DIOs in the related RPLInstance.

### 6.2. Receiving and Forwarding RREQ messages

### 6.2.1. Step 1: RREQ reception and evaluation

When a router X receives a RREQ message over a link from a neighbor Y, X first determines whether or not the RREQ is valid. If so, X then determines whether or not it has sufficient resources available to maintain the state needed to process an eventual RREP if the RREP were to be received. If not, then X MUST drop the packet and discontinue processing of the RREQ. Otherwise, X next determines whether the RREQ advertises a usable route to OrigNode, by checking whether the link to Y can be used to transmit packets to OrigNode.

When H=0 in the incoming RREQ, the router MUST drop the RREQ-DIO if one of its addresses is present in the Address Vector. When H=1 in the incoming RREQ, the router MUST drop the RREQ message if OrigSeqNo field of the RREQ is older than the SeqNo value that X has stored for a route to OrigNode. Otherwise, the router determines whether to propagate the RREQ-DIO. It does this by determining whether or not a route to OrigNode using the upstream direction of the incoming link satisfies the Objective Function (OF). In order to evaluate the OF, the router first determines the maximum useful rank (MaxUsefulRank). If the router has previously joined the RREQ-Instance associated with the RREQ-DIO, then MaxUsefulRank is set to be the Rank value that was stored when the router processed the best previous RREQ for the DODAG with the given RREQ-Instance. Otherwise, MaxUsefulRank is set to be RankLimit. If OF cannot be satisfied (i.e., the Rank evaluates to a value greater than MaxUsefulRank) the RREQ-DIO MUST be dropped, and the following steps are not processed. Otherwise, the router MUST join the RREQ-Instance and prepare to propagate the RREQ-DIO, as follows. The upstream neighbor router that transmitted the received RREQ-DIO is selected as the preferred parent.

### 6.2.2. Step 2: TargNode and Intermediate Router determination

After determining that a received RREQ provides a usable route to OrigNode, a router determines whether it is a TargNode, or a possible intermediate router between OrigNode and a TargNode, or both. The router is a TargNode if it finds one of its own addresses in a Target Option in the RREQ. After possibly propagating the RREQ according to the procedures in Steps 3, 4, and 5, the TargNode generates a RREP as specified in Section 6.3.

If the OrigNode tries to reach multiple TargNodes in a single RREQ-Instance, one of the TargNodes can be an intermediate router to other TargNodes. In this case, before transmitting the RREQ-DIO to multicast group all-RPL-nodes, a TargNode MUST delete the Target Option encapsulating its own address, so that downstream routers with higher Rank values do not try to create a route to this TargNode.

An intermediate router could receive several RREQ-DIOs from routers with lower Rank values in the same RREQ-Instance with different lists of Target Options. For the purposes of determining the intersection with previous incoming RREQ-DIOs, the intermediate router maintains a record of the targets that have been requested for a given RREQ-Instance. An incoming RREQ-DIO message having multiple ART Options coming from a router with higher Rank than the Rank of the stored targets is ignored. When transmitting the RREQ-DIO, the intersection of all received lists MUST be included if it is nonempty after TargNode has deleted the Target Option encapsulating its own address. If the intersection is empty, it means that all the targets have been reached, and the router MUST NOT transmit any RREQ-DIO. Otherwise it proceeds to Section 6.2.3.

For example, suppose two RREQ-DIOs are received with the same RPLInstance and OrigNode. Suppose further that the first RREQ has (T1, T2) as the targets, and the second one has (T2, T4) as targets. Then only T2 needs to be included in the generated RREQ-DIO.

#### 6.2.3. Step 3: Intermediate Router RREQ processing

The intermediate router establishes itself as a viable node for a route to OrigNode as follows. If the H bit is set to 1, for hop-by-hop routing, then the router MUST build or update its upward route entry towards OrigNode, which includes at least the following items: Source Address, RPLInstanceID, Destination Address, Next Hop, Lifetime, and Sequence Number. The Destination Address and the RPLInstanceID respectively can be learned from the DODAGID and the RPLInstanceID of the RREQ-DIO. The Source Address is the address used by the router to send data to the Next Hop, i.e., the preferred parent. The lifetime is set according to DODAG configuration (not the L field) and can be extended when the route is actually used. The sequence number represents the freshness of the route entry; it is copied from the Orig SeqNo field of the RREQ option. A route entry with the same source and destination address, same RPLInstanceID, but stale sequence number, MUST be deleted.

#### 6.2.4. Step 4: Symmetric Route Processing at an Intermediate Router

If the S bit of the incoming RREQ-DIO is 0, then the route cannot be symmetric, and the S bit of the RREQ-DIO to be transmitted is set to 0. Otherwise, the router MUST determine whether the downward (i.e., towards the TargNode) direction of the incoming link satisfies the OF. If so, the S bit of the RREQ-DIO to be transmitted is set to 1. Otherwise the S bit of the RREQ-DIO to be transmitted is set to 0.



When a router joins the RREQ-Instance, it also associates within its data structure for the RREQ-Instance the information about whether or not the RREQ-DIO to be transmitted has the S-bit set to 1. This information associated to RREQ-Instance is known as the S-bit of the RREQ-Instance. It will be used later during the RREP-DIO message processing Section 6.3.2.

Suppose a router has joined the RREQ-Instance, and  $H=0$ , and the S-bit of the RREQ-Instance is set to 1. In this case, the router MAY optionally associate to the RREQ-Instance, the Address Vector of the symmetric route back to OrigNode. This is useful if the router later receives an RREP-DIO that is paired with the RREQ.

#### 6.2.5. Step 5: RREQ propagation at an Intermediate Router

If the router is an intermediate router, then it transmits the RREQ-DIO to the multicast group all-RPL-nodes; if the H bit is set to 0, the intermediate router MUST append the address of its interface receiving the RREQ-DIO into the address vector.

#### 6.2.6. Step 6: RREQ reception at TargNode

If the router is a TargNode and was already associated with the RREQ-Instance, it takes no further action and does not send an RREP-DIO. If TargNode is not already associated with the RREQ-Instance, it prepares and transmits a RREP-DIO, possibly after waiting for RREP\_WAIT\_TIME, as detailed in (Section 6.3).

#### 6.3. Generating Route Reply (RREP) at TargNode

When a TargNode receives a RREQ message over a link from a neighbor Y, TargNode first follows the procedures in Section 6.2. If the link to Y can be used to transmit packets to OrigNode, TargNode generates a RREP according to the steps below. Otherwise TargNode drops the RREQ and does not generate a RREP.

If the L field is not 0, the TargNode MAY delay transmitting the RREP-DIO for duration RREP\_WAIT\_TIME to await a route with a lower Rank. The value of RREP\_WAIT\_TIME is set by default to 1/4 of the duration determined by the L field. For  $L == 0$ , RREP\_WAIT\_TIME is set by default to 0. Depending upon the application, RREP\_WAIT\_TIME may be set to other values. Smaller values enable quicker formation for the P2P route. Larger values enable formation of P2P routes with better Rank values.

The address of the OrigNode MUST be encapsulated in the ART Option and included in this RREP-DIO message along with the SeqNo of TargNode.

### 6.3.1. RREP-DIO for Symmetric route

If the RREQ-Instance corresponding to the RREQ-DIO that arrived at TargNode has the S bit set to 1, there is a symmetric route both of whose directions satisfy the Objective Function. Other RREQ-DIOs might later provide better upward routes. The method of selection between a qualified symmetric route and an asymmetric route that might have better performance is implementation-specific and out of scope.

For a symmetric route, the RREP-DIO message is unicast to the next hop according to the Address Vector (H=0) or the route entry (H=1); the DODAG in RREP-Instance does not need to be built. The RPLInstanceID in the RREP-Instance is paired as defined in Section 6.3.3. In case the H bit is set to 0, the address vector from the RREQ-DIO MUST be included in the RREP-DIO.

### 6.3.2. RREP-DIO for Asymmetric Route

When a RREQ-DIO arrives at a TargNode with the S bit set to 0, the TargNode MUST build a DODAG in the RREP-Instance corresponding to the RREQ-DIO rooted at itself, in order to provide OrigNode with a downstream route to the TargNode. The RREP-DIO message is transmitted to multicast group all-RPL-nodes.

### 6.3.3. RPLInstanceID Pairing

Since the RPLInstanceID is assigned locally (i.e., there is no coordination between routers in the assignment of RPLInstanceID), the tuple (OrigNode, TargNode, RPLInstanceID) is needed to uniquely identify a discovered route. It is possible that multiple route discoveries with dissimilar Objective Functions are initiated simultaneously. Thus between the same pair of OrigNode and TargNode, there can be multiple AODV-RPL route discovery instances. So that OrigNode and Targnode can avoid any mismatch, they MUST pair the RREQ-Instance and the RREP-Instance in the same route discovery by using the RPLInstanceID.

When preparing the RREP-DIO, a TargNode could find the RPLInstanceID candidate for the RREP-Instance is already occupied by another RPL Instance from an earlier route discovery operation which is still active. This unlikely case might happen if two distinct OrigNodes need routes to the same TargNode, and they happen to use the same RPLInstanceID for RREQ-Instance. In such cases, the RPLInstanceID of an already active RREP-Instance MUST NOT be used again for assigning RPLInstanceID for the later RREP-Instance. Reusing the same RPLInstanceID for two distinct DODAGs originated with the same DODAGID (TargNode address) would prevent intermediate routers from

distinguishing between these DODAGs (and their associated Objective Functions). Instead, the RPLInstanceID MUST be replaced by another value so that the two RREP-instances can be distinguished. In the RREP-DIO option, the Delta field of the RREP-DIO message (Figure 2) indicates the increment to be applied to the pre-existing RPLInstanceID to obtain the value of the RPLInstanceID that is used in the RREP-DIO message. When the new RPLInstanceID after incrementation exceeds 255, it rolls over starting at 0. For example, if the RREQ-InstanceID is 252, and incremented by 6, the new RPLInstanceID will be 2. Related operations can be found in Section 6.4. RPLInstanceID collisions do not occur across RREQ-DIOs; the DODAGID equals the OrigNode address and is sufficient to disambiguate between DODAGs.

#### 6.4. Receiving and Forwarding Route Reply

Upon receiving a RREP-DIO, a router which already belongs to the RREP-Instance SHOULD drop the DIO. Otherwise the router performs the steps in the following subsections.

##### 6.4.1. Step 1: Receiving and Evaluation

If the Objective Function is not satisfied, the router MUST NOT join the DODAG; the router MUST discard the RREP-DIO, and does not execute the remaining steps in this section. An Intermediate Router MUST discard a RREP if one of its addresses is present in the Address Vector, and does not execute the remaining steps in this section.

If the S bit of the associated RREQ-Instance is set to 1, the router MUST proceed to Section 6.2.2.

If the S-bit of the RREQ-Instance is set to 0, the router MUST determine whether the downward direction of the link (towards the TargNode) over which the RREP-DIO is received satisfies the Objective Function, and the router's Rank would not exceed the RankLimit. If so, the router joins the DODAG of the RREP-Instance. The router that transmitted the received RREP-DIO is selected as the preferred parent. Afterwards, other RREP-DIO messages can be received.

##### 6.4.2. Step 2: OrigNode or Intermediate Router

The router updates its stored value of the TargNode's sequence number according to the value provided in the ART option. The router next checks if one of its addresses is included in the ART Option. If so, this router is the OrigNode of the route discovery. Otherwise, it is an intermediate router.

#### 6.4.3. Step 3: Build Route to TargNode

If the H bit is set to 1, then the router (OrigNode or intermediate) MUST build a downward route entry towards TargNode which includes at least the following items: OrigNode Address, RPLInstanceID, TargNode Address as destination, Next Hop, Lifetime and Sequence Number. For a symmetric route, the Next Hop in the route entry is the router from which the RREP-DIO is received. For an asymmetric route, the Next Hop is the preferred parent in the DODAG of RREP-Instance. The RPLInstanceID in the route entry MUST be the RREQ-InstanceID (i.e., after subtracting the Delta field value from the value of the RPLInstanceID). The source address is learned from the ART Option, and the destination address is learned from the DODAGID. The lifetime is set according to DODAG configuration (i.e., not the L field) and can be extended when the route is actually used. The sequence number represents the freshness of the route entry, and is copied from the Dest SeqNo field of the ART option of the RREP-DIO. A route entry with same source and destination address, same RPLInstanceID, but stale sequence number (i.e., incoming sequence number is less than the currently stored sequence number of the route entry), MUST be deleted.

#### 6.4.4. Step 4: RREP Propagation

If the receiver is the OrigNode, it can start transmitting the application data to TargNode along the path as provided in RREP-Instance, and processing for the RREP-DIO is complete. Otherwise, the RREP will be propagated towards OrigNode. If H=0, the intermediate router MUST include the address of the interface receiving the RREP-DIO into the address vector. If H=1, according to the last step the intermediate router has set up a route entry for TargNode. If the intermediate router has a route to OrigNode, it uses that route to unicast the RREP-DIO to OrigNode. Otherwise, in case of a symmetric route, the RREP-DIO message is unicast to the Next Hop according to the address vector in the RREP-DIO (H=0) or the local route entry (H=1). Otherwise, in case of an asymmetric route, the intermediate router transmits the RREP-DIO to multicast group all-RPL-nodes. The RPLInstanceID in the transmitted RREP-DIO is the same as the value in the received RREP-DIO.

### 7. Gratuitous RREP

In some cases, an Intermediate router that receives a RREQ-DIO message MAY transmit a "Gratuitous" RREP-DIO message back to OrigNode instead of continuing to multicast the RREQ-DIO towards TargNode. The intermediate router effectively builds the RREP-Instance on behalf of the actual TargNode. The G bit of the RREP option is provided to distinguish the Gratuitous RREP-DIO (G=1) sent by the

Intermediate router from the RREP-DIO sent by TargNode (G=0).

The gratuitous RREP-DIO MAY be sent out when an intermediate router receives a RREQ-DIO for a TargNode, and the router has a pair of downward and upward routes to the TargNode which also satisfy the Objective Function and for which the destination sequence number is at least as large as the sequence number in the RREQ-DIO message.

In case of source routing, the intermediate router MUST unicast the received RREQ-DIO to TargNode including the address vector between the OrigNode and the router. Thus the TargNode can have a complete upward route address vector from itself to the OrigNode. Then the router MUST include the address vector from the TargNode to the router itself in the gratuitous RREP-DIO to be transmitted.

In case of hop-by-hop routing, the intermediate router MUST unicast the received RREQ-DIO to the Next Hop on the route. The Next Hop router along the route MUST build new route entries with the related RPLInstanceID and DODAGID in the downward direction. The above process will happen recursively until the RREQ-DIO arrives at the TargNode. Then the TargNode MUST unicast recursively the RREP-DIO hop-by-hop to the intermediate router, and the routers along the route SHOULD build new route entries in the upward direction. Upon receiving the unicast RREP-DIO, the intermediate router sends the gratuitous RREP-DIO to the OrigNode as defined in Section 6.3.

## 8. Operation of Trickle Timer

RREQ-Instance/RREP-Instance multicast uses trickle timer operations [RFC6206] to control RREQ-DIO and RREP-DIO transmissions. The Trickle control of these DIO transmissions follows the procedures described in the Section 8.3 of [RFC6550] entitled "DIO Transmission". If the route is symmetric, the RREP DIO does not need the Trickle timer mechanism.

## 9. IANA Considerations

Note to RFC editor:

The sentence "The parenthesized numbers are only suggestions." is to be removed prior publication.

A Subregistry in this section refers to a named sub-registry of the "Routing Protocol for Low Power and Lossy Networks (RPL)" registry.

AODV-RPL uses the "P2P Route Discovery Mode of Operation" (MOP == 4) with new Options as specified in this document. Please cite AODV-RPL and this document as one of the protocols using MOP 4.

IANA is asked to assign three new AODV-RPL options "RREQ", "RREP" and "ART", as described in Figure 6 from the "RPL Control Message Options" Subregistry. The parenthesized numbers are only suggestions.

Value	Meaning	Reference
TBD2 (0x0B)	RREQ Option	This document
TBD3 (0x0C)	RREP Option	This document
TBD4 (0x0D)	ART Option	This document

Figure 6: AODV-RPL Options

## 10. Security Considerations

The security considerations for the operation of AODV-RPL are similar to those for the operation of RPL (as described in Section 19 of the RPL specification [RFC6550]). Sections 6.1 and 10 of [RFC6550] describe RPL's optional security framework, which AODV-RPL relies on to provide data confidentiality, authentication, replay protection, and delay protection services. Additional analysis for the security threats to RPL can be found in [RFC7416].

A router can join a temporary DAG created for a secure AODV-RPL route discovery only if it can support the security configuration in use (see Section 6.1 of [RFC6550]), which also specifies the key in use. It does not matter whether the key is preinstalled or dynamically acquired. The router must have the key in use before it can join the DAG being created for secure route discovery.

If a rogue router knows the key for the security configuration in use, it can join the secure AODV-RPL route discovery and cause various types of damage. Such a rogue router could advertise false information in its DIOs in order to include itself in the discovered route(s). It could generate bogus RREQ-DIO, and RREP-DIO messages carrying bad routes or maliciously modify genuine RREP-DIO messages it receives. A rogue router acting as the OrigNode could launch denial-of-service attacks against the LLN deployment by initiating fake AODV-RPL route discoveries. When rogue routers might be present, RPL's preinstalled mode of operation, where the key to use for route discovery is preinstalled, SHOULD be used.

When a RREQ-DIO message uses the source routing option by setting the H bit to 0, a rogue router may populate the Address Vector field with a set of addresses that may result in the RREP-DIO traveling in a routing loop.

If a rogue router is able to forge a gratuitous RREP, it could mount denial-of-service attacks.

## 11. Acknowledgements

The authors thank Pascal Thubert, Rahul Jadhav, and Lijo Thomas for their support and valuable inputs. The authors specially thank Lavanya H.M for implementing AODV-RPL in Contiki and conducting extensive simulation studies.

The authors would like to acknowledge the review, feedback and comments from the following people, in alphabetical order: Roman Danyliw, Lars Eggert, Benjamin Kaduk, Tero Kivinen, Erik Kline, Murray Kucherawy, Warren Kumari, Francesca Palombini, Alvaro Retana, Ines Robles, John Scudder, Meral Shirazipour, Peter Van der Stok, Eric Vyncke, and Robert Wilton.

## 12. References

### 12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6206] Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko, "The Trickle Algorithm", RFC 6206, DOI 10.17487/RFC6206, March 2011, <<https://www.rfc-editor.org/info/rfc6206>>.
- [RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <<https://www.rfc-editor.org/info/rfc6550>>.
- [RFC6551] Vasseur, JP., Ed., Kim, M., Ed., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks", RFC 6551, DOI 10.17487/RFC6551, March 2012, <<https://www.rfc-editor.org/info/rfc6551>>.

- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

## 12.2. Informative References

- [co-ioam] Ballamajalu, Rashmi., S.V.R., Anand., and Malati Hegde, "Co-iOAM: In-situ Telemetry Metadata Transport for Resource Constrained Networks within IETF Standards Framework", 2018 10th International Conference on Communication Systems & Networks (COMSNETS) pp.573-576, January 2018.
- [contiki] Contiki contributors, "The Contiki Open Source OS for the Internet of Things (Contiki Version 2.7)", November 2013, <<https://github.com/contiki-os/contiki>>.
- [Contiki-ng] Contiki-NG contributors, "Contiki-NG: The OS for Next Generation IoT Devices (Contiki-NG Version 4.6)", December 2020, <<https://github.com/contiki-ng/contiki-ng>>.
- [cooja] Contiki/Cooja contributors, "Cooja Simulator for Wireless Sensor Networks (Contiki/Cooja Version 2.7)", November 2013, <<https://github.com/contiki-os/contiki/tree/master/tools/cooja>>.
- [RFC3561] Perkins, C., Belding-Royer, E., and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", RFC 3561, DOI 10.17487/RFC3561, July 2003, <<https://www.rfc-editor.org/info/rfc3561>>.
- [RFC6997] Goyal, M., Ed., Baccelli, E., Philipp, M., Brandt, A., and J. Martocci, "Reactive Discovery of Point-to-Point Routes in Low-Power and Lossy Networks", RFC 6997, DOI 10.17487/RFC6997, August 2013, <<https://www.rfc-editor.org/info/rfc6997>>.
- [RFC6998] Goyal, M., Ed., Baccelli, E., Brandt, A., and J. Martocci, "A Mechanism to Measure the Routing Metrics along a Point-to-Point Route in a Low-Power and Lossy Network", RFC 6998, DOI 10.17487/RFC6998, August 2013, <<https://www.rfc-editor.org/info/rfc6998>>.



- [RFC7276] Mizrahi, T., Sprecher, N., Bellagamba, E., and Y. Weingarten, "An Overview of Operations, Administration, and Maintenance (OAM) Tools", RFC 7276, DOI 10.17487/RFC7276, June 2014, <<https://www.rfc-editor.org/info/rfc7276>>.
- [RFC7416] Tsao, T., Alexander, R., Dohler, M., Daza, V., Lozano, A., and M. Richardson, Ed., "A Security Threat Analysis for the Routing Protocol for Low-Power and Lossy Networks (RPLs)", RFC 7416, DOI 10.17487/RFC7416, January 2015, <<https://www.rfc-editor.org/info/rfc7416>>.
- [RFC7548] Ersue, M., Ed., Romascanu, D., Schoenwaelder, J., and A. Sehgal, "Management of Networks with Constrained Devices: Use Cases", RFC 7548, DOI 10.17487/RFC7548, May 2015, <<https://www.rfc-editor.org/info/rfc7548>>.
- [RFC9030] Thubert, P., Ed., "An Architecture for IPv6 over the Time-Slotted Channel Hopping Mode of IEEE 802.15.4 (6TiSCH)", RFC 9030, DOI 10.17487/RFC9030, May 2021, <<https://www.rfc-editor.org/info/rfc9030>>.

#### Appendix A. Example: Using ETX/RSSI Values to determine value of S bit

The combination of Received Signal Strength Indication(downstream) (RSSI) and Expected Number of Transmissions(upstream) (ETX) has been tested to determine whether a link is symmetric or asymmetric at intermediate routers. We present two methods to obtain an ETX value from RSSI measurement.

Method 1: In the first method, we constructed a table measuring RSSI

vs ETX using the Cooja simulation [cooja] setup in the Contiki OS environment[contiki]. We used Contiki-2.7 running 6LoWPAN/RPL protocol stack for the simulations. For approximating the number of packet drops based on the RSSI values, we implemented simple logic that drops transmitted packets with certain pre-defined ratios before handing over the packets to the receiver. The packet drop ratio is implemented as a table lookup of RSSI ranges mapping to different packet drop ratios with lower RSSI ranges resulting in higher values. While this table has been defined for the purpose of capturing the overall link behavior, it is highly recommended to conduct physical radio measurement experiments, in general. By keeping the receiving node at different distances, we let the packets experience different packet drops as per the described method. The ETX value computation is done by another module which is part of RPL Objective Function implementation. Since ETX value is reflective of the extent of packet drops, it allowed us to prepare a useful ETX vs RSSI table. ETX versus RSSI values obtained in this way may be used as explained below:

Source----->NodeA----->NodeB----->Destination

Figure 7: Communication link from Source to Destination

RSSI at NodeA for NodeB	Expected ETX at NodeA for NodeB->NodeA
> -60	150
-70 to -60	192
-80 to -70	226
-90 to -80	662
-100 to -90	3840

Table 1: Selection of S bit based on Expected ETX value

Method 2: One could also make use of the function `guess_etx_from_rssi()` defined in the 6LoWPAN/RPL protocol stack of Contiki-ng OS [Contiki-ng] to obtain RSSI-ETX mapping. This function outputs ETX value ranging between 128 and 3840 for  $-60 \leq \text{rssi} \leq -89$ . The function description is beyond the scope of this document.

We tested the operations in this specification by making the following experiment, using the above parameters. In our experiment, a communication link is considered as symmetric if the ETX value of NodeA->NodeB and NodeB->NodeA (see Figure 7) are within, say, a 1:3 ratio. This ratio should be understood as determining the link's symmetric/asymmetric nature. NodeA can typically know the ETX value in the direction of NodeA -> NodeB but it has no direct way of knowing the value of ETX from NodeB->NodeA. Using physical testbed experiments and realistic wireless channel propagation models, one can determine a relationship between RSSI and ETX representable as an expression or a mapping table. Such a relationship in turn can be used to estimate ETX value at nodeA for link NodeB--->NodeA from the received RSSI from NodeB. Whenever nodeA determines that the link towards the nodeB is bi-directional asymmetric then the S bit is set to 0. Afterwards, the link from NodeA to Destination remains designated as asymmetric and the S bit remains set to 0.

Determination of asymmetry versus bidirectionality remains a topic of lively discussion in the IETF.

## Appendix B. Changelog

Note to the RFC Editor: please remove this section before publication.

### B.1. Changes from version 12 to version 13

- \* Changed name of "Shift" field to be the "Delta" field.
- \* Specified that if a node does not have resources, it MUST drop the RREQ.
- \* Changed name of MaxUseRank to MaxUsefulRank.
- \* Revised a sentence that was not clear about when a TargNode can delay transmission of the RREP in response to a RREQ.
- \* Provided advice about running AODV-RPL at same time as P2P-RPL or native RPL.
- \* Small reorganization and enlargement of the description of Trickle time operation in Section 8.
- \* Added definition for "RREQ-InstanceID" to Terminology section.
- \* Specified that once a node leaves an RREQ-Instance, it MUST NOT rejoin the same RREQ-Instance.

## B.2. Changes from version 11 to version 12

- \* Defined RREP\_WAIT\_TIME for asymmetric as well as symmetric handling of RREP-DIO.
- \* Clarified link-local multicast transmission to use link-local multicast group all-RPL nodes.
- \* Identified some security threats more explicitly.
- \* Specified that the pairing between RREQ-DIO and RREP-DIO happens at OrigNode and TargNode. Intermediate routers do not necessarily maintain the pairing.
- \* When RREQ-DIO is received with H=0 and S=1, specified that intermediate routers MAY store symmetric Address Vector information for possible use when a machine RREP-DIO is received.
- \* Specified that AODV-RPL uses the "P2P Route Discovery Mode of Operation" (MOP == 4), instead of requesting the allocation of a new MOP. Clarified that there is no conflict with [RFC6997].
- \* Fixed several important typos and improved language in numerous places.
- \* Reorganized the steps in the specification for handling RREQ and RREP at an intermediate router, to more closely follow the order of processing actions to be taken by the router.

## B.3. Changes from version 10 to version 11

- \* Numerous editorial improvements.
- \* Replace  $\text{Floor}((7 + (\text{Prefix Length})) / 8)$  by  $\text{Ceil}(\text{Prefix Length} / 8)$  for simplicity and ease of understanding.
- \* Use "L field" instead of "L bit" since L is a two-bit field.
- \* Improved the procedures in section 6.2.1.
- \* Define the S bit of the data structure a router uses to represent whether or not the RREQ instance is for a symmetric or an asymmetric route. This replaces text in the document that was a holdover from earlier versions in which the RREP had an S bit for that purpose.

- \* Quote terminology from AODV that has been identified as possibly originating in language reflecting various kinds of bias against certain cultures.
- \* Clarified the relationship of AODV-RPL to RPL.
- \* Eliminated the "Point-to-Point" terminology to avoid suggesting only a single link.
- \* Modified certain passages to better reflect the possibility that a router might have multiple IP addresses.
- \* "Rsv" replaced by "X X" for reserved field.
- \* Added mandates for reserved fields, and replaces some ambiguous language phraseology by mandates.
- \* Replaced "retransmit" terminology by more correct "propagate" terminology.
- \* Added text about determining link symmetry near Figure 5.
- \* Mandated checking the Address Vector to avoid routing loops.
- \* Improved specification for use of the Delta value in Section 6.3.3.
- \* Corrected the wrong use of RREQ-Instance to be RREP-Instance.
- \* Referred to Subregistry values instead of Registry values in Section 9.
- \* Sharpened language in Section 10, eliminated misleading use of capitalization in the words "Security Configuration".
- \* Added acknowledgements and contributors.

#### B.4. Changes from version 09 to version 10

- \* Changed the title for brevity and to remove acronyms.
- \* Added "Note to the RFC Editor" in Section 9.
- \* Expanded DAO and P2MP in Section 1.
- \* Reclassified [RFC6998] and [RFC7416] as Informational.
- \* SHOULD changed to MUST in Section 4.1 and Section 4.2.

- \* Several editorial improvements and clarifications.

#### B.5. Changes from version 08 to version 09

- \* Removed section "Link State Determination" and put some of the relevant material into Section 5.
- \* Cited security section of [RFC6550] as part of the RREP-DIO message description in Section 2.
- \* SHOULD has been changed to MUST in Section 4.2.
- \* Expanded the terms ETX and RSSI in Section 5.
- \* Section 6.4 has been expanded to provide a more precise explanation of the handling of route reply.
- \* Added [RFC7416] in the Security Considerations (Section 10) for RPL security threats. Cited [RFC6550] for authenticated mode of operation.
- \* Appendix A has been mostly re-written to describe methods to determine whether or not the S bit should be set to 1.
- \* For consistency, adjusted several mandates from SHOULD to MUST and from SHOULD NOT to MUST NOT.
- \* Numerous editorial improvements and clarifications.

#### B.6. Changes from version 07 to version 08

- \* Instead of describing the need for routes to "fulfill the requirements", specify that routes need to "satisfy the Objective Function".
- \* Removed all normative dependencies on [RFC6997]
- \* Rewrote Section 10 to avoid duplication of language in cited specifications.
- \* Added a new section "Link State Determination" with text and citations to more fully describe how implementations determine whether links are symmetric.
- \* Modified text comparing AODV-RPL to other protocols to emphasize the need for AODV-RPL instead of the problems with the other protocols.

- \* Clarified that AODV-RPL uses some of the base RPL specification but does not require an instance of RPL to run.
- \* Improved capitalization, quotation, and spelling variations.
- \* Specified behavior upon reception of a RREQ-DIO or RREP-DIO message for an already existing DODAGID (e.g, Section 6.4).
- \* Fixed numerous language issues in IANA Considerations Section 9.
- \* For consistency, adjusted several mandates from SHOULD to MUST and from SHOULD NOT to MUST NOT.
- \* Numerous editorial improvements and clarifications.

#### B.7. Changes from version 06 to version 07

- \* Added definitions for all fields of the ART option (see Section 4.3). Modified definition of Prefix Length to prohibit Prefix Length values greater than 127.
- \* Modified the language from [RFC6550] Target Option definition so that the trailing zero bits of the Prefix Length are no longer described as "reserved".
- \* Reclassified [RFC3561] and [RFC6998] as Informative.
- \* Added citation for [RFC8174] to Terminology section.

#### B.8. Changes from version 05 to version 06

- \* Added Security Considerations based on the security mechanisms defined in [RFC6550].
- \* Clarified the nature of improvements due to P2P route discovery versus bidirectional asymmetric route discovery.
- \* Editorial improvements and corrections.

#### B.9. Changes from version 04 to version 05

- \* Add description for sequence number operations.
- \* Extend the residence duration L in section 4.1.
- \* Change AODV-RPL Target option to ART option.

## B.10. Changes from version 03 to version 04

- \* Updated RREP option format. Remove the T bit in RREP option.
- \* Using the same RPLInstanceID for RREQ and RREP, no need to update [RFC6550].
- \* Explanation of Delta field in RREP.
- \* Multiple target options handling during transmission.

## B.11. Changes from version 02 to version 03

- \* Include the support for source routing.
- \* Import some features from [RFC6997], e.g., choice between hop-by-hop and source routing, the L field which determines the duration of residence in the DAG, RankLimit, etc.
- \* Define new target option for AODV-RPL, including the Destination Sequence Number in it. Move the TargNode address in RREQ option and the OrigNode address in RREP option into ADOV-RPL Target Option.
- \* Support route discovery for multiple targets in one RREQ-DIO.
- \* New RPLInstanceID pairing mechanism.

## Appendix C. Contributors

Abdur Rashid Sangi

Huaiyin Institute of Technology

No.89 North Beijing Road, Qinghe District

Huaian 223001

P.R. China

Email: sangi\_bahrian@yahoo.com

Malati Hegde

Indian Institute of Science

Bangalore 560012



India

Email: malati@iisc.ac.in

Mingui Zhang

Huawei Technologies

No. 156 Beiqing Rd. Haidian District

Beijing 100095

P.R. China

Email: zhangmingui@huawei.com

#### Authors' Addresses

Charles E. Perkins

Lupin Lodge

Los Gatos, 95033

United States

Email: charliep@lupinlodge.com

S.V.R Anand

Indian Institute of Science

Bangalore 560012

India

Email: anandsvr@iisc.ac.in

Satish Anamalamudi

SRM University-AP

Amaravati Campus

Amaravati, Andhra Pradesh 522 502

India

Email: satishnaidu80@gmail.com

Bing Liu

Huawei Technologies

No. 156 Beiqing Rd. Haidian District

Beijing

100095

China

Email: remy.liubing@huawei.com

ROLL  
Internet-Draft  
Intended status: Standards Track  
Expires: December 21, 2018

P. Thubert, Ed.  
Cisco  
R. Jadhav  
Huawei Tech  
J. Pylakutty  
Cisco  
June 19, 2018

Root initiated routing state in RPL  
draft-ietf-roll-dao-projection-04

Abstract

This document proposes a protocol extension to RPL that enables to install a limited amount of centrally-computed routes in a RPL graph, enabling loose source routing down a non-storing mode DODAG, or transversal routes inside the DODAG. As opposed to the classical route injection in RPL that are injected by the end devices, this draft enables the root of the DODAG to projects the routes that are needed on the nodes where they should be installed.

Status of This Memo

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

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

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

This Internet-Draft will expire on December 21, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	3
2.1. BCP 14 . . . . .	3
2.2. References . . . . .	4
2.3. Subset of a 6LoWPAN Glossary . . . . .	4
2.4. New Terms . . . . .	5
3. Extending RFC 6550 . . . . .	5
4. New RPL Control Message Options . . . . .	5
5. Projected DAO . . . . .	7
5.1. Non-storing Mode Projected Route . . . . .	8
5.2. Storing-Mode Projected Route . . . . .	9
6. Applications . . . . .	11
6.1. Loose Source Routing in Non-storing Mode . . . . .	11
6.2. Transversal Routes in storing and non-storing modes . . . . .	13
7. RPL Instances . . . . .	15
8. Security Considerations . . . . .	15
9. IANA Considerations . . . . .	15
10. Acknowledgments . . . . .	16
11. References . . . . .	16
11.1. Normative References . . . . .	16
11.2. Informative References . . . . .	17
Appendix A. Examples . . . . .	18
A.1. Using storing mode P-DAO in non-storing mode MOP . . . . .	18
A.2. Projecting a storing-mode transversal route . . . . .	19
Authors' Addresses . . . . .	20

## 1. Introduction

The "Routing Protocol for Low Power and Lossy Networks" [RFC6550] (LLN)(RPL) is a generic Distance Vector protocol that is well suited for application in a variety of low energy Internet of Things (IoT) networks. RPL forms Destination Oriented Directed Acyclic Graphs (DODAGs) in which the root often acts as the Border Router to connect the RPL domain to the Internet. The root is responsible to select the RPL Instance that is used to forward a packet coming from the Internet into the RPL domain and set the related RPL information in the packets.

The 6TiSCH architecture [I-D.ietf-6tisch-architecture] leverages RPL for its routing operation and considers the Deterministic Networking

Architecture [I-D.ietf-detnet-architecture] as one possible model whereby the device resources and capabilities are exposed to an external controller which installs routing states into the network based on some objective functions that reside in that external entity.

Based on heuristics of usage, path length, and knowledge of device capacity and available resources such as battery levels and reservable buffers, a Path Computation Element ([PCE]) with a global visibility on the system could install additional P2P routes that are more optimized for the current needs as expressed by the objective function.

This draft enables a RPL root to install and maintain projected routes (P-routes) within its DODAG, along a selected set of nodes that may or may not include self, for a chosen duration. This potentially enables routes that are more optimized than those obtained with the distributed operation of RPL, either in terms of the size of a source-route header or in terms of path length, which impacts both the latency and the packet delivery ratio. P-routes may be installed in either Storing and Non-Storing Modes Instances of the classical RPL operation, resulting in potentially hybrid situations where the mode of some P-routes is different from that of the other routes in the RPL Instance.

Projected routes must be used with the parsimony to limit the amount of state that is installed in each device to fit within its resources, and to limit the amount of rerouted traffic to fit within the capabilities of the transmission links. The algorithm used to compute the paths and the protocol used to learn the topology of the network and the resources that are available in devices and in the network are out of scope for this document. Possibly with the assistance of a Path Computation Element ([PCE]) that could have a better visibility on the larger system, the root computes which segment could be optimized and uses this draft to install the corresponding projected routes.

## 2. Terminology

### 2.1. BCP 14

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 BCP 14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

## 2.2. References

In this document, readers will encounter terms and concepts that are discussed in the following documents:

- o "Routing Protocol for Low Power and Lossy Networks" [RFC6550], and
- o "Terminology in Low power And Lossy Networks" [RFC7102].

## 2.3. Subset of a 6LoWPAN Glossary

This document often uses the following acronyms:

6BBR: 6LoWPAN Backbone Router

6LBR: 6LoWPAN Border Router

6LN: 6LoWPAN Node

6LR: 6LoWPAN Router

6CIO: Capability Indication Option

EARO: (Extended) Address Registration Option -- (E)ARO

EDAR: (Extended) Duplicate Address Request -- (E)DAR

EDAC: (Extended) Duplicate Address Confirmation -- (E)DAC

DAD: Duplicate Address Detection

DODAG: Destination-Oriented Directed Acyclic Graph

LLN: Low-Power and Lossy Network

NA: Neighbor Advertisement

NCE: Neighbor Cache Entry

ND: Neighbor Discovery

NDP: Neighbor Discovery Protocol

NS: Neighbor Solicitation

RPL: IPv6 Routing Protocol for LLNs (pronounced ripple) [RFC6550]

RA: Router Advertisement

RS: Router Solicitation

#### 2.4. New Terms

Projected Route: A route that is installed remotely by a RPL root.

#### 3. Extending RFC 6550

Section 6.7 of RPL [RFC6550] specifies Control Message Options (CMO) to be placed in RPL messages such as the Destination Advertisement Object (DAO) message. The RPL Target Option and the Transit Information Option (TIO) are such options; the former indicates a node to be reached and the latter specifies a parent that can be used to reach that node. Options may be factorized; one or more contiguous TIOs apply to the one or more contiguous Target options that immediately precede the TIOs in the RPL message.

This specification introduces 2 new Control Message Options referred to as Route Projection Options (RPO). One RPO is the Information option (VIO) and the other is the Source-Routed VIO (SRVIO). The VIO installs a route on each hop along a projected route (in a fashion analogous to RPL Storing Mode) whereas the SRVIO installs a source-routing state at the ingress node, which uses it to insert a routing header in a fashion similar to Non-Storing Mode.

Like the TIO, the RPOs MUST be preceded by one or more RPL Target Options to which they apply, and they can be factorized: multiple contiguous RPOs indicate alternate paths to the target(s).

#### 4. New RPL Control Message Options

The format of RPOs is as follows:

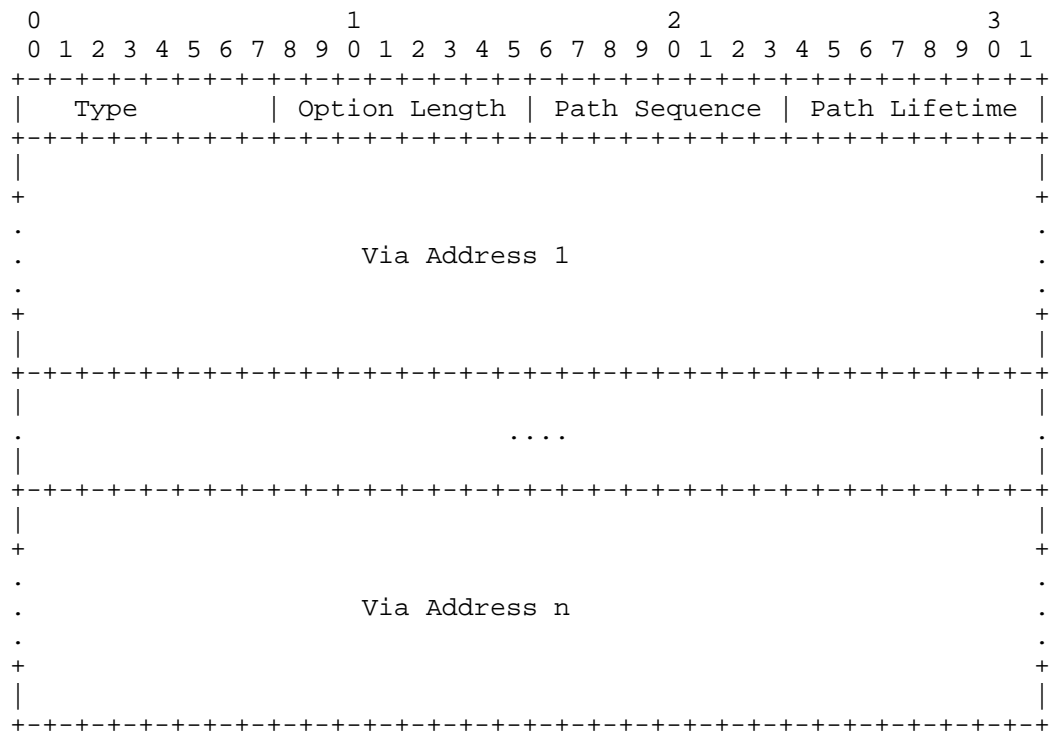


Figure 1: Via Information option format

Option Type: 0x0A for VIO, 0x0B for SRVIO (to be confirmed by IANA)

Option Length: In bytes; variable, depending on the number of Via Addresses.

Path Sequence: 8-bit unsigned integer. When a RPL Target option is issued by the root of the DODAG (i.e. in a DAO message), that root sets the Path Sequence and increments the Path Sequence each time it issues a RPL Target option with updated information. The indicated sequence deprecates any state for a given Target that was learned from a previous sequence and adds to any state that was learned for that sequence.

Path Lifetime: 8-bit unsigned integer. The length of time in Lifetime Units (obtained from the Configuration option) that the prefix is valid for route determination. The period starts when a new Path Sequence is seen. A value of all one bits (0xFF) represents infinity. A value of all zero bits (0x00) indicates a loss of reachability. A DAO message that contains

a Via Information option with a Path Lifetime of 0x00 for a Target is referred as a No-Path (for that Target) in this document.

Via Address: 16 bytes. IPv6 Address of the next hop towards the destination(s) indicated in the target option that immediately precede the RPO. Via Addresses are indicated in the order of the data path from the ingress to the egress nodes. TBD: See how the /64 prefix can be elided if it is the same as that of (all of) the target(s). In that case, the Next-Hop Address could be expressed as the 8-bytes suffix only.

An RPO MUST contain at least one Via Address, and a Via Address MUST NOT be present more than once, otherwise the RPO MUST be ignored.

## 5. Projected DAO

This draft adds a capability to RPL whereby the root of a DODAG projects a route by sending an extended DAO message called a Projected-DAO (P-DAO) to an arbitrary router in the DODAG, indicating one or more sequence(s) of routers inside the DODAG via which the target(s) indicated in the Target Information Option(s) (TIO) can be reached.

A P-DAO is sent from a global address of the root to a global address of the recipient, and MUST be confirmed by a DAO-ACK, which is sent back to a global address of the root.

A P-DAO message MUST contain at least one TIO and at least one RPO following it. There can be at most one such sequence of TIOs and then RPOs.

Like a classical DAO message, a P-DAO is processed only if it is "new" per section 9.2.2. "Generation of DAO Messages" of the RPL specification [RFC6550]; this is determined using the Path Sequence information from the RPO as opposed to a TIO. Also, a Path Lifetime of 0 in an RPO indicates that a route is to be removed.

There are two kinds of operation for the projected routes, the Storing Mode and the Non-Storing Mode.

The Non-Storing Mode is discussed in section Section 5.1. It uses an SRVIO that carries a list of Via Addresses to be used as a source-routed path to the target. The recipient of the P-DAO is the ingress router of the source-routed path. Upon a Non-Storing Mode P-DAO, the ingress router installs a source-routed state to the target and replies to the root directly with a DAO-ACK message.



The Storing Mode is discussed in section Section 5.2. It uses a VIO with one Via Address per consecutive hop, from the ingress to the egress of the path, including the list of all intermediate routers in the data path order. The Via Addresses indicate the routers in which the routing state to the target have to be installed via the next Via Address in the VIO. In normal operations, the P-DAO is propagated along the chain of Via Routers from the egress router of the path till the ingress one, which confirms the installation to the root with a DAO-ACK message. Note that the root may be the ingress and it may be the egress of the path, that it can also be neither but it cannot be both.

### 5.1. Non-storing Mode Projected Route

As illustrated in Figure 2, a P-DAO that carries an SRVIO enables the root to install a source-routed path towards a target in any particular router; with this path information the router can add a source routed header reflecting the P-route to any packet for which the current destination either is the said target or can be reached via the target.

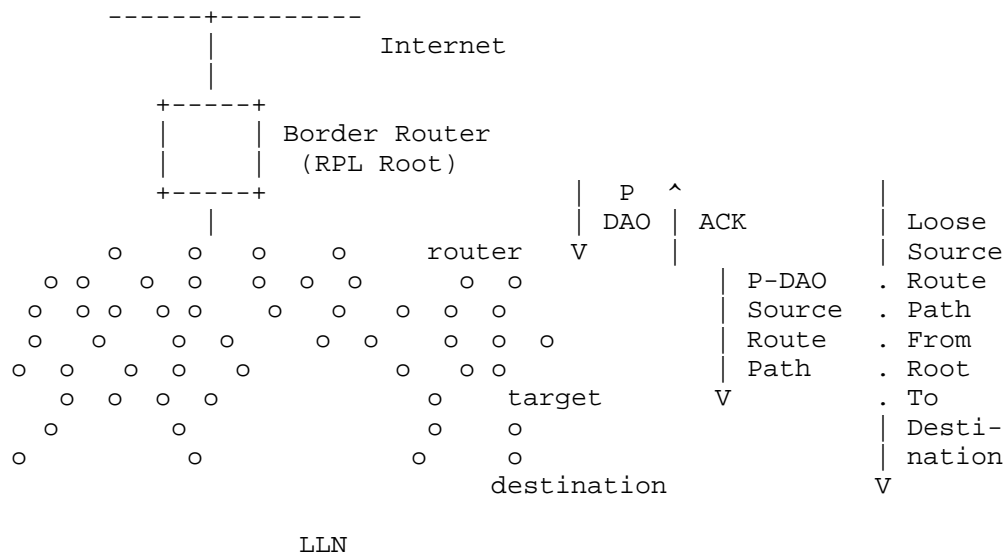


Figure 2: Projecting a Non-Storing Route

A route indicated by an SRVIO may be loose, meaning that the node that owns the next listed Via Address is not necessarily a neighbor. Without proper loop avoidance mechanisms, the interaction of loose source routing and other mechanisms may effectively cause loops. In order to avoid those loops, if the router that installs a P-route

does not have a connected route (a direct adjacency) to the next source routed hop and fails to locate it as a neighbor or a neighbor of a neighbor, then it MUST ensure that it has another projected route to the next loose hop under the control of the same route computation system, otherwise the P-DAO is rejected.

When forwarding a packet to a destination for which the router determines that routing happens via the target, the router inserts the source routing header in the packet to reach the target. In the case of a loose source-routed path, there MUST be either a neighbor that is adjacent to the loose next hop, on which case the packet is forwarded to that neighbor, or a source-routed path to the loose next hop; in the latter case, another encapsulation takes place and the process possibly recurses; otherwise the packet is dropped.

In order to add a source-routing header, the router encapsulates the packet with an IP-in-IP header and a non-storing mode source routing header (SRH) [RFC6554].

In the uncompressed form the source of the packet would be self, the destination would be the first Via Address in the SRVIO, and the SRH would contain the list of the remaining Via Addresses and then the target.

In practice, the router will normally use the "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Paging Dispatch" [RFC8025] to compress the RPL artifacts as indicated in the "6LoWPAN Routing Header" [RFC8138] specification. In that case, the router indicates self as encapsulator in an IP-in-IP 6LoRH Header, and places the list of Via Addresses in the order of the VIO and then the target in the SRH 6LoRH Header.

## 5.2. Storing-Mode Projected Route

As illustrated in Figure 3, the Storing Mode projected route is used by the root to install a routing state towards a target in the routers along a segment between an ingress and an egress router; this enables the routers to forward along that segment any packet for which the next loose hop is the said target, for instance a loose source routed packet for which the next loose hop is the target, or a packet for which the router has a routing state to the final destination via the target.

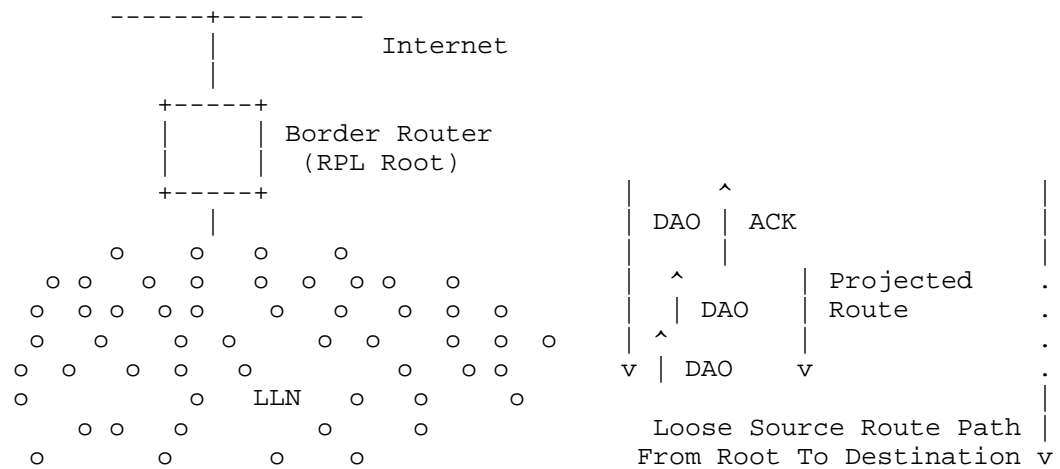


Figure 3: Projecting a route

In order to install the relevant routing state along the segment between an ingress and an egress routers, the root sends a unicast P-DAO message to the egress router of the routing segment that must be installed. The P-DAO message contains the ordered list of hops along the segment as a direct sequence of Via Information options that are preceded by one or more RPL Target options to which they relate. Each Via Information option contains a Path Lifetime for which the state is to be maintained.

The root sends the P-DAO directly to the egress node of the segment. In that P-DAO, the destination IP address matches the Via Address in the last VIO. This is how the egress recognizes its role. In a similar fashion, the ingress node recognizes its role as it matches Via Address in the first VIO.

The egress node of the segment is the only node in the path that does not install a route in response to the P-DAO; it is expected to be already able to route to the target(s) on its own. It may either be the target, or may have some existing information to reach the target(s), such as a connected route or an already installed projected route. If one of the targets cannot be located, the node MUST answer to the root with a negative DAO-ACK listing the target(s) that could not be located (suggested status 10 to be confirmed by IANA).

If the egress node can reach all the targets, then it forwards the P-DAO with unchanged content to its loose predecessor in the segment as indicated in the list of Via Information options, and recursively

the message is propagated unchanged along the sequence of routers indicated in the P-DAO, but in the reverse order, from egress to ingress.

The address of the predecessor to be used as destination of the propagated DAO message is found in the Via Information option the precedes the one that contain the address of the propagating node, which is used as source of the packet.

Upon receiving a propagated DAO, an intermediate router as well as the ingress router install a route towards the DAO target(s) via its successor in the P-DAO; the router locates the VIO that contains its address, and uses as next hop the address found in the Via Address field in the following VIO. The router MAY install additional routes towards the addresses that are located in VIOs that are after the next one, if any, but in case of a conflict or a lack of resource, a route to a target installed by the root has precedence.

The process recurses till the P-DAO is propagated to ingress router of the segment, which answers with a DAO-ACK to the root.

Also, the path indicated in a P-DAO may be loose, in which case the reachability to the next hop has to be asserted. Each router along the path indicated in a P-DAO is expected to be able to reach its successor, either with a connected route (direct neighbor), or by routing, for instance following a route installed previously by a DAO or a P-DAO message. If that route is not connected then a recursive lookup may take place at packet forwarding time to find the next hop to reach the target(s). If it does not and cannot reach the next router in the P-DAO, the router MUST answer to the root with a negative DAO-ACK indicating the successor that is unreachable (suggested status 11 to be confirmed by IANA).

A Path Lifetime of 0 in a Via Information option is used to clean up the state. The P-DAO is forwarded as described above, but the DAO is interpreted as a No-Path DAO and results in cleaning up existing state as opposed to refreshing an existing one or installing a new one.

## 6. Applications

### 6.1. Loose Source Routing in Non-storing Mode

A RPL implementation operating in a very constrained LLN typically uses the Non-Storing Mode of Operation as represented in Figure 4. In that mode, a RPL node indicates a parent-child relationship to the root, using a Destination Advertisement Object (DAO) that is unicast from the node directly to the root, and the root typically builds a

source routed path to a destination down the DODAG by recursively concatenating this information.

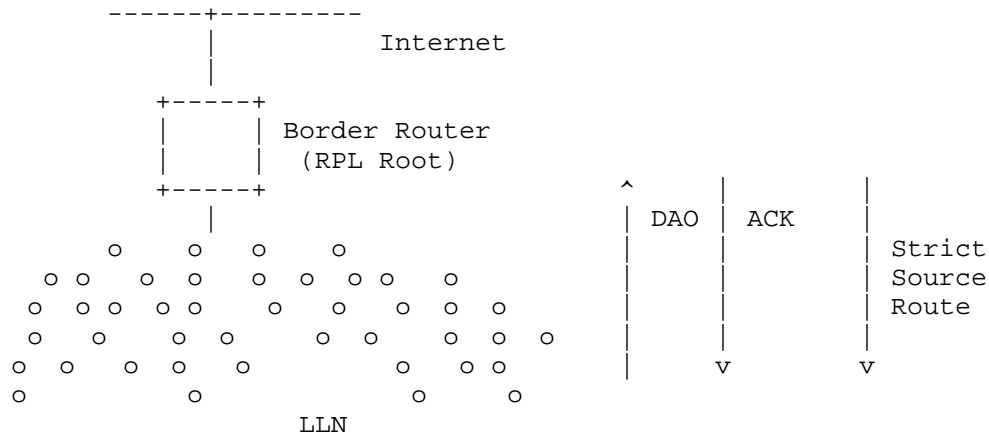


Figure 4: RPL non-storing mode of operation

Based on the parent-children relationships expressed in the non-storing DAO messages, the root possesses topological information about the whole network, though this information is limited to the structure of the DODAG for which it is the destination. A packet that is generated within the domain will always reach the root, which can then apply a source routing information to reach the destination if the destination is also in the DODAG. Similarly, a packet coming from the outside of the domain for a destination that is expected to be in a RPL domain reaches the root.

It results that the root, or then some associated centralized computation engine such as a PCE, can determine the amount of packets that reach a destination in the RPL domain, and thus the amount of energy and bandwidth that is wasted for transmission, between itself and the destination, as well as the risk of fragmentation, any potential delays because of a paths longer than necessary (shorter paths exist that would not traverse the root).

As a network gets deep, the size of the source routing header that the root must add to all the downward packets becomes an issue for nodes that are many hops away. In some use cases, a RPL network forms long lines and a limited amount of well-targeted routing state would allow to make the source routing operation loose as opposed to strict, and save packet size. Limiting the packet size is directly beneficial to the energy budget, but, mostly, it reduces the chances of frame loss and/or packet fragmentation, which is highly detrimental to the LLN operation. Because the capability to store a

routing state in every node is limited, the decision of which route is installed where can only be optimized with a global knowledge of the system, a knowledge that the root or an associated PCE may possess by means that are outside of the scope of this specification.

This specification enables to store source-routed or storing mode state in intermediate routers, which enables to limit the excursion of the source route headers in deep networks. Once a P-DAO exchange has taken place for a given target, if the root operates in non storing mode, then it may elide the sequence of routers that is installed in the network from its source route headers to destination that are reachable via that target, and the source route headers effectively become loose.

## 6.2. Transversal Routes in storing and non-storing modes

RPL is optimized for Point-to-Multipoint (P2MP), root to leaves and Multipoint-to-Point (MP2P) leaves to root operations, whereby routes are always installed along the RPL DODAG. Transversal Peer to Peer (P2P) routes in a RPL network will generally suffer from some stretch since routing between 2 peers always happens via a common parent, as illustrated in Figure 5:

- o in non-storing mode, all packets routed within the DODAG flow all the way up to the root of the DODAG. If the destination is in the same DODAG, the root must encapsulate the packet to place a Routing Header that has the strict source route information down the DODAG to the destination. This will be the case even if the destination is relatively close to the source and the root is relatively far off.
- o In storing mode, unless the destination is a child of the source, the packets will follow the default route up the DODAG as well. If the destination is in the same DODAG, they will eventually reach a common parent that has a route to the destination; at worse, the common parent may also be the root. From that common parent, the packet will follow a path down the DODAG that is optimized for the Objective Function that was used to build the DODAG.

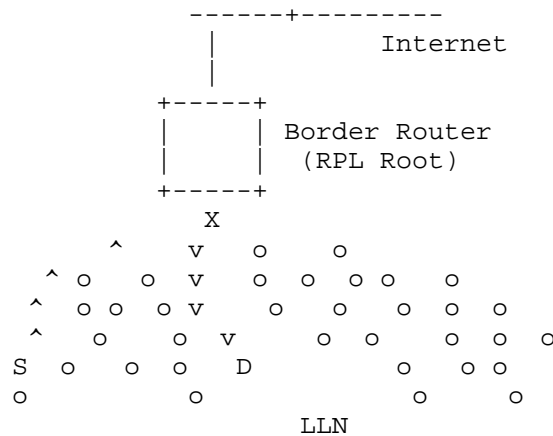


Figure 5: Routing Stretch between S and D via common parent X

It results that it is often beneficial to enable transversal P2P routes, either if the RPL route presents a stretch from shortest path, or if the new route is engineered with a different objective. For that reason, earlier work at the IETF introduced the "Reactive Discovery of Point-to-Point Routes in Low Power and Lossy Networks" [RFC6997], which specifies a distributed method for establishing optimized P2P routes. This draft proposes an alternate based on a centralized route computation.

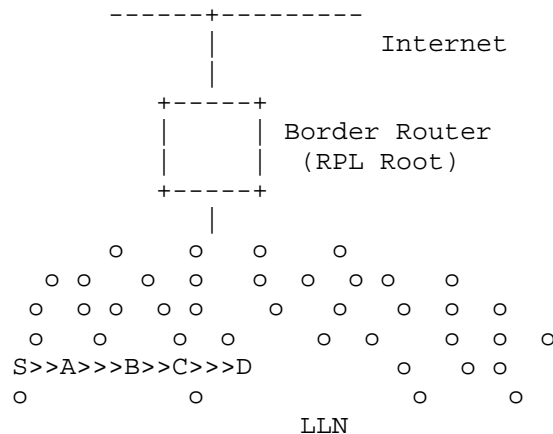


Figure 6: Projected Transversal Route

This specification enables to store source-routed or storing mode state in intermediate routers, which enables to limit the stretch of a P2P route and maintain the characteristics within a given SLA. An

example of service using this mechanism could be a control loop that would be installed in a network that uses classical RPL for asynchronous data collection. In that case, the P2P path may be installed in a different RPL Instance, with a different objective function.

## 7. RPL Instances

It must be noted that RPL has a concept of instance but does not have a concept of an administrative distance, which exists in certain proprietary implementations to sort out conflicts between multiple sources of routing information. This draft conforms the instance model as follows:

- o If the PCE needs to influence a particular instance to add better routes in conformance with the routing objectives in that instance, it may do so. When the PCE modifies an existing instance then the added routes must not create a loop in that instance. This is achieved by always preferring a route obtained from the PCE over a route that is learned via RPL.
- o If the PCE installs a more specific (say, Traffic Engineered) route between a particular pair of nodes then it SHOULD use a Local Instance from the ingress node of that path. A packet associated with that instance will be routed along that path and MUST NOT be placed over a Global Instance again. A packet that is placed on a Global Instance may be injected in the Local Instance based on node policy and the Local Instance parameters.

In all cases, the path is indicated by a new Via Information option, and the flow is similar to the flow used to obtain loose source routing.

## 8. Security Considerations

This draft uses messages that are already present in RPL [RFC6550] with optional secured versions. The same secured versions may be used with this draft, and whatever security is deployed for a given network also applies to the flows in this draft.

## 9. IANA Considerations

This document extends the IANA registry created by RFC 6550 for RPL Control Codes as follows:



Code	Description	Reference
0x0A	Via	This document
0x0B	Source-Routed Via	This document

#### RPL Control Codes

This document is updating the registry created by RFC 6550 for the RPL 3-bit Mode of Operation (MOP) as follows:

MOP value	Description	Reference
5	Non-Storing mode of operation with Projected routes	This document
6	Storing mode of operation with Projected routes	This document

#### DIO Mode of operation

## 10. Acknowledgments

The authors wish to acknowledge JP Vasseur and Patrick Wetterwald for their contributions to the ideas developed here.

## 11. References

### 11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <<https://www.rfc-editor.org/info/rfc6550>>.

- [RFC6551] Vasseur, JP., Ed., Kim, M., Ed., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks", RFC 6551, DOI 10.17487/RFC6551, March 2012, <<https://www.rfc-editor.org/info/rfc6551>>.
- [RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, DOI 10.17487/RFC6554, March 2012, <<https://www.rfc-editor.org/info/rfc6554>>.
- [RFC8025] Thubert, P., Ed. and R. Cragie, "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Paging Dispatch", RFC 8025, DOI 10.17487/RFC8025, November 2016, <<https://www.rfc-editor.org/info/rfc8025>>.
- [RFC8138] Thubert, P., Ed., Bormann, C., Toutain, L., and R. Cragie, "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing Header", RFC 8138, DOI 10.17487/RFC8138, April 2017, <<https://www.rfc-editor.org/info/rfc8138>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

## 11.2. Informative References

- [I-D.ietf-6tisch-architecture]  
Thubert, P., "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4", draft-ietf-6tisch-architecture-14 (work in progress), April 2018.
- [I-D.ietf-detnet-architecture]  
Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-05 (work in progress), May 2018.
- [PCE] IETF, "Path Computation Element", <<https://datatracker.ietf.org/doc/charter-ietf-pce/>>.
- [RFC6997] Goyal, M., Ed., Baccelli, E., Philipp, M., Brandt, A., and J. Martocci, "Reactive Discovery of Point-to-Point Routes in Low-Power and Lossy Networks", RFC 6997, DOI 10.17487/RFC6997, August 2013, <<https://www.rfc-editor.org/info/rfc6997>>.

[RFC7102] Vasseur, JP., "Terms Used in Routing for Low-Power and Lossy Networks", RFC 7102, DOI 10.17487/RFC7102, January 2014, <<https://www.rfc-editor.org/info/rfc7102>>.

## Appendix A. Examples

### A.1. Using storing mode P-DAO in non-storing mode MOP

In non-storing mode, the DAG root maintains the knowledge of the whole DODAG topology, so when both the source and the destination of a packet are in the DODAG, the root can determine the common parent that would have been used in storing mode, and thus the list of nodes in the path between the common parent and the destination. For instance in the diagram shown in Figure 7, if the source is node 41 and the destination is node 52, then the common parent is node 22.

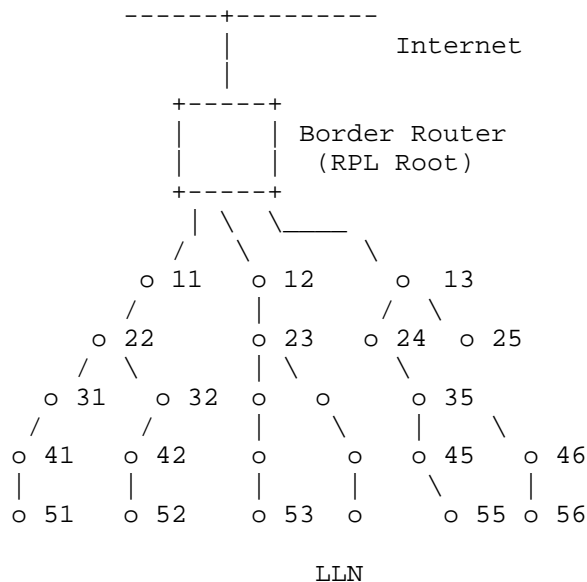


Figure 7: Example DODAG forming a logical tree topology

With this draft, the root can install a storing mode routing states along a segment that is either from itself to the destination, or from one or more common parents for a particular source/destination pair towards that destination (in this particular example, this would be the segment made of nodes 22, 32, 42).

In the example below, say that there is a lot of traffic to nodes 55 and 56 and the root decides to reduce the size of routing headers to those destinations. The root can first send a DAO to node 45



The process recurses till the P-DAO reaches S, the ingress of the segment, which installs a route to D via A and sends a DAO-ACK to the root.

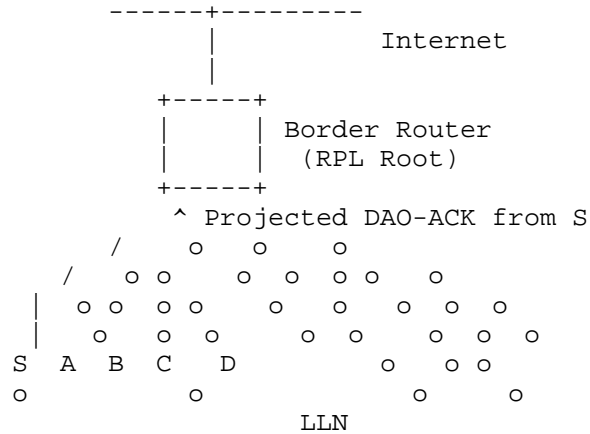


Figure 9: Projected DAO-ACK to root

As a result, a transversal route is installed that does not need to follow the DODAG structure.

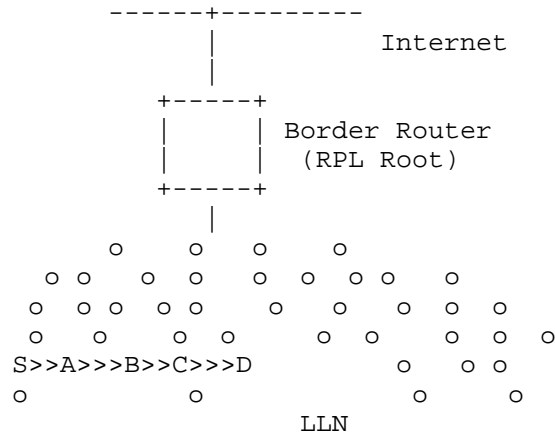


Figure 10: Projected Transversal Route

Authors' Addresses

Pascal Thubert (editor)  
Cisco Systems  
Village d'Entreprises Green Side  
400, Avenue de Roumanille  
Batiment T3  
Biot - Sophia Antipolis 06410  
FRANCE

Phone: +33 4 97 23 26 34  
Email: pthubert@cisco.com

Rahul Arvind Jadhav  
Huawei Tech  
Kundalahalli Village, Whitefield,  
Bangalore, Karnataka 560037  
India

Phone: +91-080-49160700  
Email: rahul.ietf@gmail.com

James Pylakutty  
Cisco Systems  
Cessna Business Park  
Kadubeesanahalli  
Marathalli ORR  
Bangalore, Karnataka 560087  
INDIA

Phone: +91 80 4426 4140  
Email: mundenma@cisco.com

ROLL  
Internet-Draft  
Intended status: Standards Track  
Expires: 24 September 2022

P. Thubert, Ed.  
Cisco Systems  
R.A. Jadhav  
Huawei Tech  
M. Richardson  
Sandelman  
23 March 2022

Root initiated routing state in RPL  
draft-ietf-roll-dao-projection-25

Abstract

THIS RFC extends RFC 6550, RFC 6553, and RFC 8138 to enable a RPL Root to install and maintain Projected Routes within its DODAG, along a selected set of nodes that may or may not include self, for a chosen duration. This potentially enables routes that are more optimized or resilient than those obtained with the classical distributed operation of RPL, either in terms of the size of a Routing Header or in terms of path length, which impacts both the latency and the packet delivery ratio.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://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 24 September 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	4
2. Terminology . . . . .	4
2.1. Requirements Language . . . . .	4
2.2. References . . . . .	5
2.3. Glossary . . . . .	5
2.4. Domain Terms . . . . .	5
2.4.1. Projected Route . . . . .	6
2.4.2. Projected DAO . . . . .	6
2.4.3. Path . . . . .	6
2.4.4. Routing Stretch . . . . .	6
2.4.5. Track . . . . .	7
3. Context and Goal . . . . .	9
3.1. RPL Applicability . . . . .	10
3.2. RPL Routing Modes . . . . .	11
3.3. Requirements . . . . .	12
3.3.1. Loose Source Routing . . . . .	12
3.3.2. East-West Routes . . . . .	13
3.4. On Tracks . . . . .	15
3.4.1. Building Tracks With RPL . . . . .	15
3.4.2. Tracks and RPL Instances . . . . .	16
3.5. Serial Track Signaling . . . . .	16
3.5.1. Using Storing Mode Segments . . . . .	18
3.5.2. Using Non-Storing Mode joining Tracks . . . . .	24
3.6. Complex Tracks . . . . .	31
3.7. Scope and Expectations . . . . .	33
3.7.1. External Dependencies . . . . .	33
3.7.2. Positioning vs. Related IETF Standards . . . . .	33
4. Extending existing RFCs . . . . .	35
4.1. Extending RFC 6550 . . . . .	35
4.1.1. Projected DAO . . . . .	36
4.1.2. Projected DAO-ACK . . . . .	38
4.1.3. Via Information Option . . . . .	39
4.1.4. Sibling Information Option . . . . .	39
4.1.5. P-DAO Request . . . . .	39
4.1.6. Amending the RPI . . . . .	40
4.1.7. Additional Flag in the RPL DODAG Configuration Option . . . . .	40
4.2. Extending RFC 6553 . . . . .	41
4.3. Extending RFC 8138 . . . . .	42
5. New RPL Control Messages and Options . . . . .	43



5.1.	New P-DAO Request Control Message . . . . .	43
5.2.	New PDR-ACK Control Message . . . . .	45
5.3.	Via Information Options . . . . .	46
5.4.	Sibling Information Option . . . . .	49
6.	Root Initiated Routing State . . . . .	51
6.1.	RPL Network Setup . . . . .	51
6.2.	Requesting a Track . . . . .	52
6.3.	Identifying a Track . . . . .	53
6.4.	Installing a Track . . . . .	54
6.4.1.	Signaling a Projected Route . . . . .	55
6.4.2.	Installing a Track Segment with a Storing Mode P-Route . . . . .	56
6.4.3.	Installing a Track Leg with a Non-Storing Mode P-Route . . . . .	58
6.5.	Tearing Down a P-Route . . . . .	60
6.6.	Maintaining a Track . . . . .	60
6.6.1.	Maintaining a Track Segment . . . . .	61
6.6.2.	Maintaining a Track Leg . . . . .	61
6.7.	Encapsulating and Forwarding Along a Track . . . . .	62
6.8.	Compression of the RPL Artifacts . . . . .	64
7.	Lesser Constrained Variations . . . . .	66
7.1.	Storing Mode Main DODAG . . . . .	66
7.2.	A Track as a Full DODAG . . . . .	68
8.	Profiles . . . . .	69
9.	Backwards Compatibility . . . . .	71
10.	Security Considerations . . . . .	72
11.	IANA Considerations . . . . .	72
11.1.	RPL DODAG Configuration Option Flag . . . . .	72
11.2.	Elective 6LoWPAN Routing Header Type . . . . .	73
11.3.	Critical 6LoWPAN Routing Header Type . . . . .	73
11.4.	Subregistry For The RPL Option Flags . . . . .	73
11.5.	RPL Control Codes . . . . .	74
11.6.	RPL Control Message Options . . . . .	74
11.7.	SubRegistry for the Projected DAO Request Flags . . . . .	75
11.8.	SubRegistry for the PDR-ACK Flags . . . . .	75
11.9.	Subregistry for the PDR-ACK Acceptance Status Values . . . . .	76
11.10.	Subregistry for the PDR-ACK Rejection Status Values . . . . .	76
11.11.	SubRegistry for the Via Information Options Flags . . . . .	77
11.12.	SubRegistry for the Sibling Information Option Flags . . . . .	77
11.13.	Destination Advertisement Object Flag . . . . .	77
11.14.	Destination Advertisement Object Acknowledgment Flag . . . . .	78
11.15.	New ICMPv6 Error Code . . . . .	78
11.16.	RPL Rejection Status values . . . . .	78
12.	Acknowledgments . . . . .	79
13.	Normative References . . . . .	79
14.	Informative References . . . . .	81
	Authors' Addresses . . . . .	83

## 1. Introduction

RPL, the "Routing Protocol for Low Power and Lossy Networks" [RPL] (LLNs), is an anisotropic Distance Vector protocol that is well-suited for application in a variety of low energy Internet of Things (IoT) networks where stretched P2P paths are acceptable vs. the signaling and state overhead involved in maintaining shortest paths across.

RPL forms destination Oriented Directed Acyclic Graphs (DODAGs) in which the Root often acts as the Border router to connect the RPL domain to the IP backbone and routes along that graph up, towards the Root, and down towards the nodes.

With this specification, an abstract routing function called a Path Computation Element [PCE] (e.g., located in a central controller or collocated with the Root) interacts with the RPL Root to compute Peer to Peer (P2P) paths within a pre-existing RPL Main DODAG. The topological information that is passed to the PCE is derived from the DODAG that is already available at the Root in RPL Non-Storing Mode. This specification introduces protocol extensions that enrich the topological information that is available at the Root and passed to the PCE.

Based on usage, path length, and knowledge of available resources such as battery levels and reservable buffers in the nodes, the PCE with a global visibility on the system can optimize the computed routes for the application needs, including the capability to provide path redundancy. This specification also introduces protocol extensions that enable the Root to translates the computed paths into RPL and install them as Projected Routes (aka P-Routes) inside the DODAG on behalf of a PCE.

## 2. Terminology

### 2.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in THIS RFC are to be interpreted as described in BCP 14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

In addition, the terms "Extends" and "Amends" are used as per [I-D.kuehlewind-update-tag] section 3.

## 2.2. References

In THIS RFC, readers will encounter terms and concepts that are discussed in the "Routing Protocol for Low Power and Lossy Networks" [RPL], the "6TiSCH Architecture" [RFC9030], the "Deterministic Networking Architecture" [RFC8655], the "Reliable and Available Wireless (RAW) Architecture" [RAW-ARCHI], and "Terminology in Low power And Lossy Networks" [RFC7102].

## 2.3. Glossary

THIS RFC often uses the following acronyms:

CMO: Control Message Option  
DAO: destination Advertisement Object  
DAG: Directed Acyclic Graph  
DODAG: destination-Oriented Directed Acyclic Graph; A DAG with only one vertex (i.e., node) that has no outgoing edge (i.e., link)  
GUA: IPv6 Global Unicast Address  
LLN: Low-Power and Lossy Network  
MOP: RPL Mode of Operation  
P-DAO: Projected DAO  
P-Route: Projected Route  
PDR: P-DAO Request  
RAN: RPL-Aware Node (either a RPL router or a RPL-Aware Leaf)  
RAL: RPL-Aware Leaf  
RH: Routing Header  
RPI: RPL Packet Information  
RTO: RPL Target Option  
RUL: RPL-Unaware Leaf  
SIO: RPL Sibling Information Option  
ULA: IPv6 Unique Local Address  
NSM-VIO: A Source-Routed Via Information Option, used in Non-Storing Mode P-DAO messages.  
SLO: Service Level Objective  
TIO: RPL Transit Information Option  
SM-VIO: A strict Via Information Option, used in Storing Mode P-DAO messages.  
VIO: A Via Information Option; it can be a SM-VIO or an NSM-VIO.

## 2.4. Domain Terms

This specification uses the following terminology:

#### 2.4.1. Projected Route

A RPL P-Route is a RPL route that is computed remotely by a PCE, and installed and maintained by a RPL Root on behalf of the PCE. It is installed as a state that signals that destinations (aka Targets) are reachable along a sequence of nodes.

#### 2.4.2. Projected DAO

A DAO message used to install a P-Route.

#### 2.4.3. Path

Quoting section 1.1.3 of [INT-ARCHI]:

At a given moment, all the IP datagrams from a particular source host to a particular destination host will typically traverse the same sequence of gateways. We use the term "path" for this sequence. Note that a path is uni-directional; it is not unusual to have different paths in the two directions between a given host pair.

Section 2 of [I-D.irtf-panrg-path-properties] points to a longer, more modern definition of path, which begins as follows:

A sequence of adjacent path elements over which a packet can be transmitted, starting and ending with a node. A path is unidirectional. Paths are time-dependent, i.e., the sequence of path elements over which packets are sent from one node to another may change. A path is defined between two nodes.

It follows that the general acceptance of a path is a linear sequence of nodes, as opposed to a multi-dimensional graph. In the context of this document, a path is observed by following one copy of a packet that is injected in a Track and possibly replicated within.

#### 2.4.4. Routing Stretch

RPL is anisotropic, meaning that it is directional, or more exactly polar. RPL does not behave the same way "down" with multicast DIO messages that form the DODAG and "up" with unicast DAO messages that follow the DODAG. This is in contrast with traditional IGPs that operate the same in all directions and are thus called isotropic.

The term Routing Stretch denotes the length of a path, as compared with a shortest path, which can be an abstract concept in RPL when the metrics are statistical and dynamic, and the concept of short varies with the Objective Function.

The RPL DODAG optimizes the P2MP (from Root) and MP2P (to Root) paths, but the P2P (node to node) traffic has to follow the same DODAG. Following the DODAG, the RPL datapath passes via a common parent in Storing Mode and via the Root in Non-Storing Mode. This typically involves more hops and more latency than the minimum possible for a direct P2P path that an isotropic protocol would compute. We refer to this elongated path as stretched.

### 2.4.5. Track

A networking graph that can be followed to transport packets with equivalent treatment; as opposed to the definition of a path above, a Track is not necessarily linear. It may contain multiple paths that may fork and rejoin, and may enable the RAW Packet ARQ, Replication, Elimination, and Overhearing (PAREO) operations.

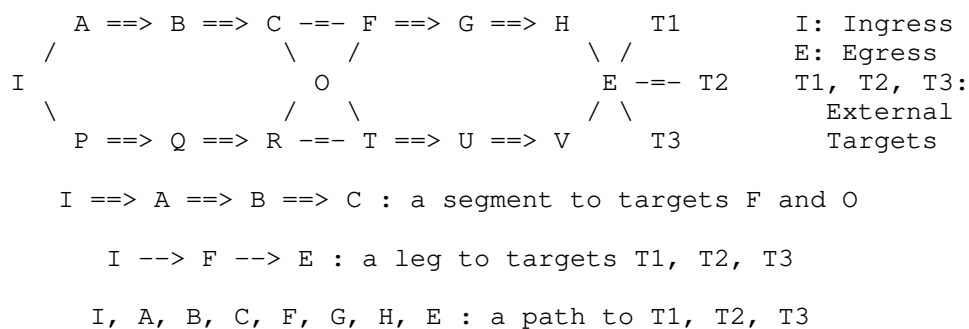


Figure 1: A Track and its Components

This specification builds Tracks that are DODAGs oriented towards a Track Ingress, and the forward direction for packets (aka East-West) is from the Track Ingress to one of the possibly multiple Track Egress Nodes, which is also down the DODAG.

The Track may be strictly connected, meaning that the vertices are adjacent, or loosely connected, meaning that the vertices are connected using Segments that are associated to the same Track.

#### 2.4.5.1. TrackID

A RPL Local InstanceID that identifies a Track using the namespace owned by the Track Ingress. The TrackID is associated with the IPv6 Address of the Track Ingress that is used as DODAGID, and together they form a unique identification of the Track (see the definition of DODAGID in section 2 of [RPL]).

#### 2.4.5.2. Namespace

The term namespace is used to refer to the scope of the TrackID. The TrackID is locally significant within its namespace. The namespace is identified by the DODAGID for the Track. The tuple (DODAGID, TrackID) is globally unique.

#### 2.4.5.3. Serial Track

A Track that has only one path.

#### 2.4.5.4. Stand-Alone

A single P-DAO that fully defines a Track, e.g., a Serial Track installed with a single Storing Mode Via Information option (SM-VIO).

#### 2.4.5.5. Stitching

This specification using the term stitching to indicate that a track is piped to another one, meaning that traffic out of the first is injected in the other.

#### 2.4.5.6. Leg

An end-to-end East-West serial path. A leg can be a serial Track by itself or a subTrack of a complex Track with the same Ingress and Egress Nodes. With this specification, a Leg is installed by the Root of the main DODAG using a Non-Storing Mode P-DAO message, and it is expressed as a loose sequence of nodes that are joined by Track Segments.

As the Non-Storing Mode Via Information option (NSM-VIO) can only signal sequences of nodes, it takes one Non-Storing Mode P-DAO message per Leg to signal the structure of a complex Track.

Each NSM-VIO for the same TrackId but a different Segment ID signals a different leg that the Track Ingress adds to the topology.

#### 2.4.5.7. subTrack

A Track within a Track, formed by a non-empty collection of Legs of the Track.

#### 2.4.5.8. Segment

A serial path formed by a strict sequence of nodes, along which a P-Route is installed. With this specification, a Segment is typically installed by the Root of the main DODAG using Storing Mode P-DAO messages. A Segment is used as the topological edge of a Track joining the loose steps along the Legs that form the structure of a complex Track. The same segment may be leveraged by more than one Leg where the Legs overlap.

Since this specification builds only DODAGs, all Segments are oriented from Ingress (East) to Egress (West), as opposed to the general Track model in the RAW Architecture [RAW-ARCHI], which allows North/South Segments that can be bidirectional as well.

##### 2.4.5.8.1. Section of a Segment

A continuous subset of a segment that may be replaced while the segment remains. for instance, in segment  $A \Rightarrow B \Rightarrow C \Rightarrow D \Rightarrow E \Rightarrow F$ , say that the link C to D might be misbehaving. The section  $B \Rightarrow C \Rightarrow D \Rightarrow E$  in the segment may be replaced by  $B \Rightarrow C' \Rightarrow D' \Rightarrow E$  to route around the problem. The segment becomes  $A \Rightarrow B \Rightarrow C' \Rightarrow D' \Rightarrow E \Rightarrow F$ .

##### 2.4.5.8.2. Segment Routing and SRH

The terms Segment Routing and SRH refer to using source-routing to hop over segments. In a Non-Storing mode RPL domain, the SRH is typically a RPL Source Route Header (the IPv6 RH of type 3) as defined in [RFC6554].

If the network is a 6LoWPAN Network, the expectation is that the SRH is compressed and encoded as a 6LoWPAN Routing Header (6LoRH), as specified in section 5 of [RFC8138].

On the other hand, if the RPL Network is less constrained and operated in Storing Mode, as discussed in Section 7.1, the Segment Routing operation and the SRH could be as specified in [RFC8754]. This specification applies equally to both forms of source routing and SRH.

### 3. Context and Goal

### 3.1. RPL Applicability

RPL is optimized for situations where the power is scarce, the bandwidth constrained and the transmissions unreliable. This matches the use case of an IoT LLN where RPL is typically used today, but also situations of high relative mobility between the nodes in the network (aka swarming), e.g., within a variable set of vehicles with a similar global motion, or a toon of drones.

To reach this goal, RPL is primarily designed to minimize the control plane activity, that is the relative amount of routing protocol exchanges vs. data traffic, and the amount of state that is maintained in each node. RPL does not need converge, and provides connectivity to most nodes most of the time.

RPL may form multiple topologies called instances. Instances can be created to enforce various optimizations through objective functions, or to reach out through different Root Nodes. The concept of objective function allows to adapt the activity of the routing protocol to the use case, e.g., type, speed, and quality of the LLN links.

RPL instances operate as ships passing in the night, unbeknownst of one another. The RPL Root is responsible to select the RPL Instance that is used to forward a packet coming from the Backbone into the RPL domain and set the related RPL information in the packets. 6TiSCH leverages RPL for its distributed routing operations.

To reduce the routing exchanges, RPL leverages an anisotropic Distance Vector approach, which does not need a global knowledge of the topology, and only optimizes the routes to and from the RPL Root, allowing P2P paths to be stretched. Although RPL installs its routes proactively, it only maintains them lazily, in reaction to actual traffic, or as a slow background activity.

This is simple and efficient in situations where the traffic is mostly directed from or to a central node, such as the control traffic between routers and a controller of a Software Defined Networking (SDN) infrastructure or an Autonomic Control Plane (ACP).

But stretch in P2P routing is counter-productive to both reliability and latency as it introduces additional delay and chances of loss. As a result, [RPL] is not a good fit for the use cases listed in the RAW use cases document [USE-CASES], which demand high availability and reliability, and as a consequence require both short and diverse paths.



### 3.2. RPL Routing Modes

RPL first forms a default route in each node towards the a Root, and those routes together coalesce as a Directed Acyclic Graph upwards. RPL then constructs routes to destinations signaled as Targets in the reverse direction, down the same DODAG. So do so, a RPL Instance can be operated either in RPL Storing or Non-Storing Mode of Operation (MOP). The default route towards the Root is maintained aggressively and may change while a packet progresses without causing loops, so the packet will still reach the Root.

In Non-Storing Mode, each node advertises itself as a Target directly to the Root, indicating the parents that may be used to reach self. Recursively, the Root builds and maintains an image of the whole DODAG in memory, and leverages that abstraction to compute source route paths for the packets to their destinations down the DODAG. When a node changes its point(s) of attachment to the DODAG, it takes single unicast packet to the Root along the default route to update it, and the connectivity is restored immediately; this mode is preferable for use cases where internet connectivity is dominant, or when, like here, the Root controls the network activity in the nodes.

In Storing Mode, the routing information percolates upwards, and each node maintains the routes to the subDAG of its descendants down the DODAG. The maintenance is lazy, either reactive upon traffic or as a slow background process. Packets flow via the common parent and the routing stretch is reduced vs. Non-Storing, for a better P2P connectivity. On the other hand, a new route takes a longer time to propagate to the Root, time for the Distance-Vector protocol to operate hop-by-hop, and the Internet connectivity is restored more slowly upon movement.

Either way, the RPL routes are injected by the Target nodes, in a distributed fashion. To complement RPL and eliminate routing stretch, this specification introduces an hybrid mode that combines Storing and Non-Storing operations to build and project routes onto the nodes where they should be installed. This specification uses the term Projected Route (P-Route) to refer to those routes.

A P-Route may be installed in either Storing and Non-Storing Mode, potentially resulting in hybrid situations where the Mode of the P-Route is different from that of the RPL Main DODAG. P-Routes can be used as stand-alone segments to reduce the size of the source routing headers with loose source routing operations down the main RPL DODAG. P-Routes can also be combined with other P-Routes to form a more complex forwarding graph called a Track.

### 3.3. Requirements

#### 3.3.1. Loose Source Routing

A RPL implementation operating in a very constrained LLN typically uses the Non-Storing Mode of Operation as represented in Figure 2. In that mode, a RPL node indicates a parent-child relationship to the Root, using a destination Advertisement Object (DAO) that is unicast from the node directly to the Root, and the Root typically builds a source routed path to a destination down the DODAG by recursively concatenating this information.

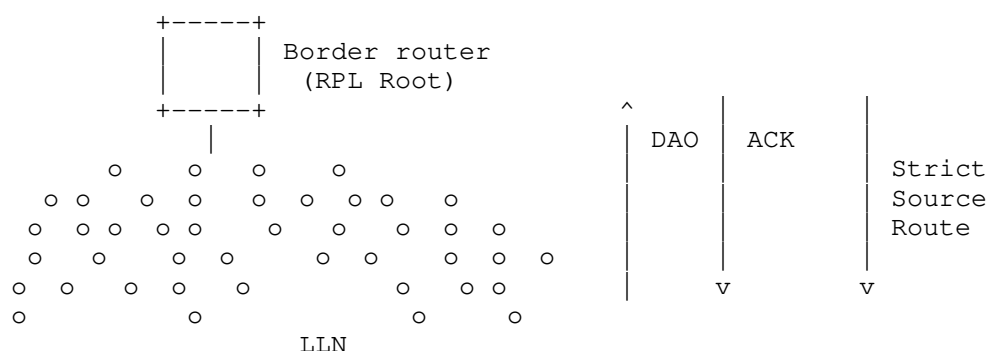


Figure 2: RPL Non-Storing Mode of operation

Based on the parent-children relationships expressed in the Non-Storing DAO messages, the Root possesses topological information about the whole network, though this information is limited to the structure of the DODAG for which it is the destination. A packet that is generated within the domain will always reach the Root, which can then apply a source routing information to reach the destination if the destination is also in the DODAG. Similarly, a packet coming from the outside of the domain for a destination that is expected to be in a RPL domain reaches the Root. It results that the wireless bandwidth near the Root is the gating factor for all transmissions towards or within the domain, and that the Root is a single point of failure for all connectivity to nodes within its domain.

The RPL Root must add a source routing header to all downward packets. As a network grows, the size of the source routing header augments with the depth of the nodes. In some use cases, a RPL network forms long lines along physical structures such as streets for lighting. Limiting the packet size is directly beneficial to the energy budget, but, mostly, it reduces the chances of frame loss and packet fragmentation, which are highly detrimental to the LLN operation. A limited amount of well-targeted routing state would

allow the source routing operation to be loose as opposed to strict, and save packet size. Because the capability to store a routing state in every node is limited, the decision of which route is installed where can only be optimized with a global knowledge of the system, a knowledge that the Root or an associated PCE may possess by means that are outside of the scope of this specification.

Being on path for all packets in Non-Storing mode, the Root may determine the number of P2P packets in its RPL domain per source and destination, the latency incurred, and the amount of energy and bandwidth that is consumed to reach the self and then down, including a possible fragmentation when encapsulating larger packets. Enabling a shorter path that would not traverse the Root for select P2P source/destinations may improve the latency, lower the consumption of constrained resources, free bandwidth at the bottleneck near the Root, improve the delivery ratio and reduce the latency for those P2P flows with a global benefit for all flows of reducing the load at the Root.

This requirement is to store a routing state associated with the Main DODAG in selected RPL routers, to limit the excursion of the source route headers in deep networks. The Root may elide the sequence of routers that is installed in the network from its source route header, which becomes loose while it is strict in [RPL].

### 3.3.2. East-West Routes

[RPL] optimizes Point-to-Multipoint (P2MP) routes from the Root, Multipoint-to-Point (MP2P) routes to the DODAG Root, and Internet access when the Root also serves as Border Router. All routes are installed North-South (aka up/down) along the RPL DODAG. Peer to Peer (P2P) East-West routes in a RPL network will generally suffer from some elongated (stretched) path versus a direct (optimized) path, since routing between two nodes always happens via a common parent, as illustrated in Figure 3:

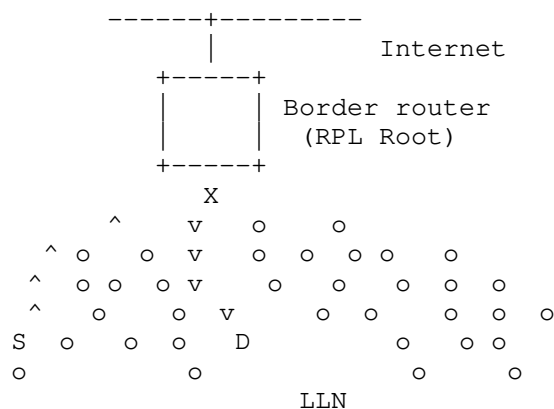


Figure 3: Routing Stretch between S and D via common parent X  
along North-South Paths

As described in [RFC9008], the amount of stretch depends on the Mode of Operation:

- \* In Non-Storing Mode, all packets routed within the DODAG flow all the way up to the Root of the DODAG. If the destination is in the same DODAG, the Root must encapsulate the packet to place an RH that has the strict source route information down the DODAG to the destination. This will be the case even if the destination is relatively close to the source and the Root is relatively far off.
- \* In Storing Mode, unless the destination is a child of the source, the packets will follow the default route up the DODAG as well. If the destination is in the same DODAG, they will eventually reach a common parent that has a route to the destination; at worse, the common parent may also be the Root. From that common parent, the packet will follow a path down the DODAG that is optimized for the Objective Function that was used to build the DODAG.

It results that it is often beneficial to enable East-West P2P routes, either if the RPL route presents a stretch from shortest path, or if the new route is engineered with a different objective, and that it is even more critical in Non-Storing Mode than it is in Storing Mode, because the routing stretch is wider. For that reason, earlier work at the IETF introduced the "Reactive Discovery of Point-to-Point Routes in Low Power and Lossy Networks" [RFC6997], which specifies a distributed method for establishing optimized P2P routes. This draft proposes an alternate based on a centralized route computation.

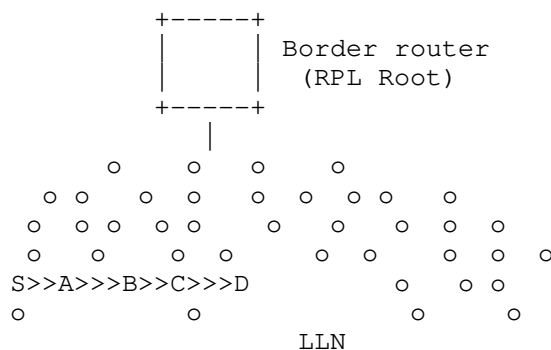


Figure 4: More direct East-West Route between S and D

The requirement is to install additional routes in the RPL routers, to reduce the stretch of some P2P routes and maintain the characteristics within a given SLO, e.g., in terms of latency and/or reliability.

### 3.4. On Tracks

#### 3.4.1. Building Tracks With RPL

The concept of a Track was introduced in the "6TiSCH Architecture" [RFC9030], as a collection of potential paths that leverage redundant forwarding solutions along the way. This can be a DODAG or a more complex structure that is only partially acyclic (e.g., per packet).

With this specification, a Track is shaped as a DODAG, and following the directed edges leads to a Track Ingress. Storing Mode P-DAO messages follow the direction of the edges to set up routes for traffic that flows the other way, towards the Track Egress(es). If there is a single Track Egress, then the Track is reversible to form another DODAG by reversing the direction of each edge. A node at the Ingress of more than one Segment in a Track may use one or more of these Segments to forward a packet inside the Track.

A RPL Track is a collection of (one or more) parallel loose source routed sequences of nodes ordered from Ingress to Egress, each forming a Track Leg. The nodes that are directly connected, reachable via existing Tracks as illustrated in Section 3.5.2.3 or joined with strict Segments of other nodes as shown in Section 3.5.1.3. The Legs are expressed in RPL Non-Storing Mode and require an encapsulation to add a Source Route Header, whereas the Segments are expressed in RPL Storing Mode.

A Serial Track comprises provides only one path between Ingress and Egress. It comprises at most one Leg. A Stand-Alone Segment implicitly defines a Serial Track from its Ingress to Egress.

A complex Track forms a graph that provides a collection of potential paths to provide redundancy for the packets, either as a collection of Legs that may be parallel or cross at certain points, or as a more generic DODAG.

### 3.4.2. Tracks and RPL Instances

Section 5.1. of [RPL] describes the RPL Instance and its encoding. There can be up to 128 global RPL Instances, for which there can be one or more DODAGs, and there can be 64 local RPL Instances, with a namespace that is indexed by a DODAGID, where the DODAGID is a Unique Local Address (ULA) or a Global Unicast Address (GUA) of the Root of the DODAG. Bit 0 (most significant) is set to 1 to signal a Local RPLInstanceID, as shown in Figure 5. By extension, this specification expresses the value of the RPLInstanceID as a single integer between 128 and 191, representing both the Local RPLInstanceID in 0..63 and Bit 0 set.

```

0 1 2 3 4 5 6 7
+---+---+---+---+---+---+
|1|D|   ID   |  Local RPLInstanceID in 0..63
+---+---+---+---+---+---+

```

Figure 5: Local RPLInstanceID Encoding

A Track is normally associated with a Local RPL Instance which RPLInstanceID is used as the TrackID, more in Section 6.3. A Track Leg may also be used as a subTrack that extends the RPL main DODAG. In that case, the TrackID is set to the global RPLInstanceID of the main DODAG, which suffices to identify the routing topology. As opposed to local RPL instances, the Track Ingress that encapsulates the packets over a subtrack is not Root, and that the source address of the encapsulated packet is not used to determine the Track.

### 3.5. Serial Track Signaling

This specification enables to set up a P-Route along either a Track Leg or a Segment. A P-Route is installed and maintained by the Root of the main DODAG using an extended RPL DAO message called a Projected DAO (P-DAO), and a Track is composed of the combination of one or more P-Routes.

A P-DAO message for a Track signals the TrackID in the RPLInstanceID field. In the case of a local RPL Instance, the address of the Track Ingress is used as source to encapsulate packets along the Track. The Track is signaled in the DODAGID field of the Projected DAO Base Object, see Figure 8.

This specification introduces the Via Information Option (VIO) to signal a sequence of hops in a Leg or a Segment in the P-DAO messages, either in Storing Mode (SM-VIO) or Non-Storing Mode (NSM-VIO). One P-DAO messages contains a single VIO, associated to one or more RPL Target Options that signal the destination IPv6 addresses that can be reached along the Track, more in Section 5.3.

Before diving deeper into Track Legs and Segments signaling and operation, this section provides examples of what how route projection works through variations of a simple example. This simple example illustrates the case of host routes, though RPL Targets can be prefixes.

Say we want to build a Serial Track from node A to E in Figure 6, so A can route packets to E's neighbors F and G along A, B, C, D and E as opposed to via the Root:

```

                                     /==> F
A ==> B ==> C ==> D==> E <
                                     \==> G

```

Figure 6: Reference Track

Conventionally we use ==> to represent a strict hop and --> for a loose hop. We use "-to-", such as in C==>D==>E-to-F to represent coma-separated Targets, e.g., F is a Target for Segment C==>D==>E. In this example, A is Track Ingress, E is Track Egress. C is a stitching point. F and G are "external" Targets for the Track, and become reachable from A via the Track A(ingress) to E (Egress and implicit Target in Non-Storing Mode) leading to F and G (explicit Targets).

Figure 5 depicts the format of the RPLInstanceID encoding for a local RPLInstanceID .

In a general manner the desired outcome is as follows:

- \* Targets are E, F, and G
- \* P-DAO 1 signals C==>D==>E

- \* P-DAO 2 signals  $A \Rightarrow B \Rightarrow C$

- \* P-DAO 3 signals F and G via the  $A \dashrightarrow E$  Track

P-DAO 3 may be omitted if P-DAO 1 and 2 signal F and G as Targets.

Loose sequences of hops must be expressed in Non-Storing Mode, so P-DAO 3 contains a NSM-VIO. With this specification, the DODAGID to be used by the Ingress as source address is signaled if needed in the DAO base object, the via list starts at the first loose hop and matches the source route header, and the Egress of a Non-Storing Mode P-DAO is an implicit Target that is not listed in the RTO.

### 3.5.1. Using Storing Mode Segments

$A \Rightarrow B \Rightarrow C$  and  $C \Rightarrow D \Rightarrow E$  are segments of a same Track. Note that the Storing Mode signaling imposes strict continuity in a segment, since the P-DAO is passed hop by hop, as a classical DAO is, along the reverse datapath that it signals. One benefit of strict routing is that loops are avoided along the Track.

#### 3.5.1.1. Stitched Segments

In this formulation:

- \* P-DAO 1 signals  $C \Rightarrow D \Rightarrow E$ -to-F,G

- \* P-DAO 2 signals  $A \Rightarrow B \Rightarrow C$ -to-F,G

Storing Mode P-DAO 1 is sent to E and when it is successfully acknowledged, Storing Mode P-DAO 2 is sent to C, as follows:



Field	P-DAO 1 to E	P-DAO 2 to C
Mode	Storing	Storing
Track Ingress	A	A
(DODAGID, TrackID)	(A, 129)	(A, 129)
SegmentID	1	2
VIO	C, D, E	A, B, C
Targets	F, G	F, G

Table 1: P-DAO Messages

As a result the RIBs are set as follows:

Node	destination	Origin	Next Hop(s)	TrackID
E	F, G	P-DAO 1	Neighbor	(A, 129)
D	E	P-DAO 1	Neighbor	(A, 129)
"	F, G	P-DAO 1	E	(A, 129)
C	D	P-DAO 1	Neighbor	(A, 129)
"	F, G	P-DAO 1	D	(A, 129)
B	C	P-DAO 2	Neighbor	(A, 129)
"	F, G	P-DAO 2	C	(A, 129)
A	B	P-DAO 2	Neighbor	(A, 129)
"	F, G	P-DAO 2	B	(A, 129)

Table 2: RIB setting

Packets originated by A to F or G do not require an encapsulation as the RPI can be placed in the native header chain. For packets that it routes, A must encapsulate to add the RPI that signals the trackID; the outer headers of the packets that are forwarded along the Track have the following settings:

Header	IPv6 Source Addr.	IPv6 Dest. Addr.	TrackID in RPI
Outer	A	F or G	(A, 129)
Inner	X != A	F or G	N/A

Table 3: Packet Header Settings

As an example, say that A has a packet for F. Using the RIB above:

- \* From P-DAO 2: A forwards to B and B forwards to C.
- \* From P-DAO 1: C forwards to D and D forwards to E.
- \* From Neighbor Cache Entry: E delivers the packet to F.

#### 3.5.1.2. External routes

In this example, we consider F and G as destinations that are external to the Track as a DODAG, as discussed in section 4.1.1. of [RFC9008]. We then apply the directives for encapsulating in that case, more in Section 6.7.

In this formulation, we set up the Track Leg explicitly, which creates less routing state in intermediate hops at the expense of larger packets to accommodate source routing:

- \* P-DAO 1 signals C==>D==>E-to-E
- \* P-DAO 2 signals A==>B==>C-to-E
- \* P-DAO 3 signals F and G via the A-->E-to-F,G Track

Storing Mode P-DAO 1 and 2, and Non-Storing Mode P-DAO 3, are sent to E, C and A, respectively, as follows:

	P-DAO 1 to E	P-DAO 2 to C	P-DAO 3 to A
Mode	Storing	Storing	Non-Storing
Track Ingress	A	A	A
(DODAGID, TrackID)	(A, 129)	(A, 129)	(A, 129)
SegmentID	1	2	3
VIO	C, D, E	A, B, C	E
Targets	E	E	F, G

Table 4: P-DAO Messages

Note in the above that E is not an implicit Target in Storing mode, so it must be added in the RTO.

As a result the RIBs are set as follows:

Node	destination	Origin	Next Hop(s)	TrackID
E	F, G	P-DAO 1	Neighbor	(A, 129)
D	E	P-DAO 1	Neighbor	(A, 129)
C	D	P-DAO 1	Neighbor	(A, 129)
"	E	P-DAO 1	D	(A, 129)
B	C	P-DAO 2	Neighbor	(A, 129)
"	E	P-DAO 2	C	(A, 129)
A	B	P-DAO 2	Neighbor	(A, 129)
"	E	P-DAO 2	B	(A, 129)
"	F, G	P-DAO 3	E	(A, 129)

Table 5: RIB setting

Packets from A to E do not require an encapsulation. The outer headers of the packets that are forwarded along the Track have the following settings:

Header	IPv6 Source Addr.	IPv6 Dest. Addr.	TrackID in RPI
Outer	A	E	(A, 129)
Inner	X	E (X != A), F or G	N/A

Table 6: Packet Header Settings

As an example, say that A has a packet for F. Using the RIB above:

- \* From P-DAO 3: A encapsulates the packet the Track signaled by P-DAO 3, with the outer header above. Now the packet destination is E.
- \* From P-DAO 2: A forwards to B and B forwards to C.
- \* From P-DAO 1: C forwards to D and D forwards to E; E decapsulates the packet.
- \* From Neighbor Cache Entry: E delivers packets to F or G.

#### 3.5.1.3. Segment Routing

In this formulation leverages Track Legs to combine Segments and form a Graph. The packets are source routed from a Segment to the next to adapt the path. As such, this can be seen as a form of Segment Routing [RFC8402]:

- \* P-DAO 1 signals C==>D==>E-to-E
- \* P-DAO 2 signals A==>B-to-B,C
- \* P-DAO 3 signals F and G via the A-->C-->E-to-F,G Track

Storing Mode P-DAO 1 and 2, and Non-Storing Mode P-DAO 3, are sent to E, B and A, respectively, as follows:

	P-DAO 1 to E	P-DAO 2 to B	P-DAO 3 to A
Mode	Storing	Storing	Non-Storing
Track Ingress	A	A	A
(DODAGID, TrackID)	(A, 129)	(A, 129)	(A, 129)
SegmentID	1	2	3
VIO	C, D, E	A, B	C, E
Targets	E	C	F, G

Table 7: P-DAO Messages

Note in the above that the Segment can terminate at the loose hop as used in the example of P-DAO 1 or at the previous hop as done with P-DAO 2. Both methods are possible on any Segment joined by a loose Track Leg. P-DAO 1 generates more signaling since E is the Segment Egress when D could be, but has the benefit that it validates that the connectivity between D and E still exists.

As a result the RIBs are set as follows:

Node	destination	Origin	Next Hop(s)	TrackID
E	F, G	P-DAO 1	Neighbor	(A, 129)
D	E	P-DAO 1	Neighbor	(A, 129)
C	D	P-DAO 1	Neighbor	(A, 129)
"	E	P-DAO 1	D	(A, 129)
B	C	P-DAO 2	Neighbor	(A, 129)
A	B	P-DAO 2	Neighbor	(A, 129)
"	C	P-DAO 2	B	(A, 129)
"	E, F, G	P-DAO 3	C, E	(A, 129)

Table 8: RIB setting

Packets originated at A to E do not require an encapsulation, but carry a SRH via C. The outer headers of the packets that are forwarded along the Track have the following settings:

Header	IPv6 Source Addr.	IPv6 Dest. Addr.	TrackID in RPI
Outer	A	C till C then E	(A, 129)
Inner	X	E (X != A), F or G	N/A

Table 9: Packet Header Settings

As an example, say that A has a packet for F. Using the RIB above:

- \* From P-DAO 3: A encapsulates the packet the Track signaled by P-DAO 3, with the outer header above. Now the destination in the IPv6 Header is C, and a SRH signals the final destination is E.
- \* From P-DAO 2: A forwards to B and B forwards to C.
- \* From P-DAO 3: C processes the SRH and sets the destination in the IPv6 Header to E.
- \* From P-DAO 1: C forwards to D and D forwards to E; E decapsulates the packet.
- \* From the Neighbor Cache Entry: E delivers packets to F or G.

### 3.5.2. Using Non-Storing Mode joining Tracks

In this formulation:

- \* P-DAO 1 signals C==>D==>E-to-F,G
- \* P-DAO 2 signals A==>B==>C-to-E,F,G

A==>B==>C and C==>D==>E are Tracks expressed as Non-Storing P-DAOs.

#### 3.5.2.1. Stitched Tracks

Non-Storing Mode P-DAO 1 and 2 are sent to C and A respectively, as follows:

	P-DAO 1 to C	P-DAO 2 to A
Mode	Non-Storing	Non-Storing
Track Ingress	C	A
(DODAGID, TrackID)	(C, 131)	(A, 131)
SegmentID	1	1
VIO	D, E	B, C
Targets	F, G	E, F, G

Table 10: P-DAO Messages

As a result the RIBs are set as follows:

Node	destination	Origin	Next Hop(s)	TrackID
E	F, G	ND	Neighbor	Any
D	E	ND	Neighbor	Any
C	D	ND	Neighbor	Any
"	E, F, G	P-DAO 1	D, E	(C, 131)
B	C	ND	Neighbor	Any
A	B	ND	Neighbor	Any
"	C, E, F, G	P-DAO 2	B, C	(A, 131)

Table 11: RIB setting

Packets originated at A to E, F and G do not require an encapsulation, though it is preferred that A encapsulates and C decapsulates. Either way, they carry a SRH via B and C, and C needs to encapsulate to E, F, or G to add an SRH via D and E. The encapsulating headers of packets that are forwarded along the Track between C and E have the following settings:

Header	IPv6 Source Addr.	IPv6 Dest. Addr.	TrackID in RPI
Outer	C	D till D then E	(C, 131)
Inner	X	E, F, or G	N/A

Table 12: Packet Header Settings between C and E

As an example, say that A has a packet for F. Using the RIB above:

- \* From P-DAO 2: A encapsulates the packet with destination of F in the Track signaled by P-DAO 2. The outer header has source A, destination B, an SRH that indicates C as the next loose hop, and a RPI indicating a TrackId of 131 from A's namespace, which is distinct from TrackId of 131 from C's.
- \* From the SRH: Packets forwarded by B have source A, destination C, a consumed SRH, and a RPI indicating a TrackId of 131 from A's namespace. C decapsulates.
- \* From P-DAO 1: C encapsulates the packet with destination of F in the Track signaled by P-DAO 1. The outer header has source C, destination D, an SRH that indicates E as the next loose hop, and a RPI indicating a TrackId of 131 from C's namespace. E decapsulates.

#### 3.5.2.2. External routes

In this formulation:

- \* P-DAO 1 signals C==>D==>E-to-E
- \* P-DAO 2 signals A==>B==>C-to-C,E
- \* P-DAO 3 signals F and G via the A-->E-to-F,G Track

Non-Storing Mode P-DAO 1 is sent to C and Non-Storing Mode P-DAO 2 and 3 are sent A, as follows:



	P-DAO 1 to C	P-DAO 2 to A	P-DAO 3 to A
Mode	Non-Storing	Non-Storing	Non-Storing
Track Ingress	C	A	A
(DODAGID, TrackID)	(C, 131)	(A, 129)	(A, 141)
SegmentID	1	1	1
VIO	D, E	B, C	E
Targets	E	E	F, G

Table 13: P-DAO Messages

As a result the RIBs are set as follows:

Node	destination	Origin	Next Hop(s)	TrackID
E	F, G	ND	Neighbor	Any
D	E	ND	Neighbor	Any
C	D	ND	Neighbor	Any
"	E	P-DAO 1	D, E	(C, 131)
B	C	ND	Neighbor	Any
A	B	ND	Neighbor	Any
"	C, E	P-DAO 2	B, C	(A, 129)
"	F, G	P-DAO 3	E	(A, 141)

Table 14: RIB setting

The encapsulating headers of packets that are forwarded along the Track between C and E have the following settings:

Header	IPv6 Source Addr.	IPv6 Dest. Addr.	TrackID in RPI
Outer	C	D till D then E	(C, 131)
Middle	A	E	(A, 141)
Inner	X	E, F or G	N/A

Table 15: Packet Header Settings

As an example, say that A has a packet for F. Using the RIB above:

- \* From P-DAO 3: A encapsulates the packet with destination of F in the Track signaled by P-DAO 3. The outer header has source A, destination E, and a RPI indicating a TrackId of 141 from A's namespace. This recurses with:
- \* From P-DAO 2: A encapsulates the packet with destination of E in the Track signaled by P-DAO 2. The outer header has source A, destination B, an SRH that indicates C as the next loose hop, and a RPI indicating a TrackId of 129 from A's namespace.
- \* From the SRH: Packets forwarded by B have source A, destination C, a consumed SRH, and a RPI indicating a TrackId of 129 from A's namespace. C decapsulates.
- \* From P-DAO 1: C encapsulates the packet with destination of E in the Track signaled by P-DAO 1. The outer header has source C, destination D, an SRH that indicates E as the next loose hop, and a RPI indicating a TrackId of 131 from C's namespace. E decapsulates.

### 3.5.2.3. Segment Routing

In this formulation:

- \* P-DAO 1 signals C==>D==>E-to-E
- \* P-DAO 2 signals A==>B-to-C
- \* P-DAO 3 signals F and G via the A-->C-->E-to-F,G Track

Non-Storing Mode P-DAO 1 is sent to C and Non-Storing Mode P-DAO 2 and 3 are sent A, as follows:

	P-DAO 1 to C	P-DAO 2 to A	P-DAO 3 to A
Mode	Non-Storing	Non-Storing	Non-Storing
Track Ingress	C	A	A
(DODAGID, TrackID)	(C, 131)	(A, 129)	(A, 141)
SegmentID	1	1	1
VIO	D, E	B	C, E
Targets		C	F, G

Table 16: P-DAO Messages

As a result the RIBs are set as follows:

Node	destination	Origin	Next Hop(s)	TrackID
E	F, G	ND	Neighbor	Any
D	E	ND	Neighbor	Any
C	D	ND	Neighbor	Any
"	E	P-DAO 1	D, E	(C, 131)
B	C	ND	Neighbor	Any
A	B	ND	Neighbor	Any
"	C	P-DAO 2	B, C	(A, 129)
"	E, F, G	P-DAO 3	C, E	(A, 141)

Table 17: RIB setting

The encapsulating headers of packets that are forwarded along the Track between A and B have the following settings:

Header	IPv6 Source Addr.	IPv6 Dest. Addr.	TrackID in RPI
Outer	A	B till D then E	(A, 129)
Middle	A	C	(A, 141)
Inner	X	E, F or G	N/A

Table 18: Packet Header Settings

The encapsulating headers of packets that are forwarded along the Track between B and C have the following settings:

Header	IPv6 Source Addr.	IPv6 Dest. Addr.	TrackID in RPI
Outer	A	C	(A, 141)
Inner	X	E, F or G	N/A

Table 19: Packet Header Settings

The encapsulating headers of packets that are forwarded along the Track between C and E have the following settings:

Header	IPv6 Source Addr.	IPv6 Dest. Addr.	TrackID in RPI
Outer	C	D till D then E	(C, 131)
Middle	A	E	(A, 141)
Inner	X	E, F or G	N/A

Table 20: Packet Header Settings

As an example, say that A has a packet for F. Using the RIB above:

- \* From P-DAO 3: A encapsulates the packet with destination of F in the Track signaled by P-DAO 3. The outer header has source A, destination C, an SRH that indicates E as the next loose hop, and a RPI indicating a TrackId of 141 from A's namespace. This recurses with:

- \* From P-DAO 2: A encapsulates the packet with destination of C in the Track signaled by P-DAO 2. The outer header has source A, destination B, and a RPI indicating a TrackId of 129 from A's namespace. B decapsulates forwards to C based on a sibling connected route.
- \* From the SRH: C consumes the SRH and makes the destination E.
- \* From P-DAO 1: C encapsulates the packet with destination of E in the Track signaled by P-DAO 1. The outer header has source C, destination D, an SRH that indicates E as the next loose hop, and a RPI indicating a TrackId of 131 from C's namespace. E decapsulates.

### 3.6. Complex Tracks

To increase the reliability of the P2P transmission, this specification enables to build a collection of Legs between the same Ingress and Egress Nodes and combine them with the same TrackID, as shown in Figure 7. Legs may cross at the edges of loose hops or remain parallel.

The Segments that join the loose hops of a Leg are installed with the same TrackID as the Leg. But each individual Leg and Segment has its own P-RouteID which allows it to be managed separately. When Legs cross within respective Segment, the next loose hop (the current destination of the packet) indicates which Leg is being followed and a Segment that can reach that next loose hop is selected.

CPF

CPF

CPF

CPF

## Southbound API

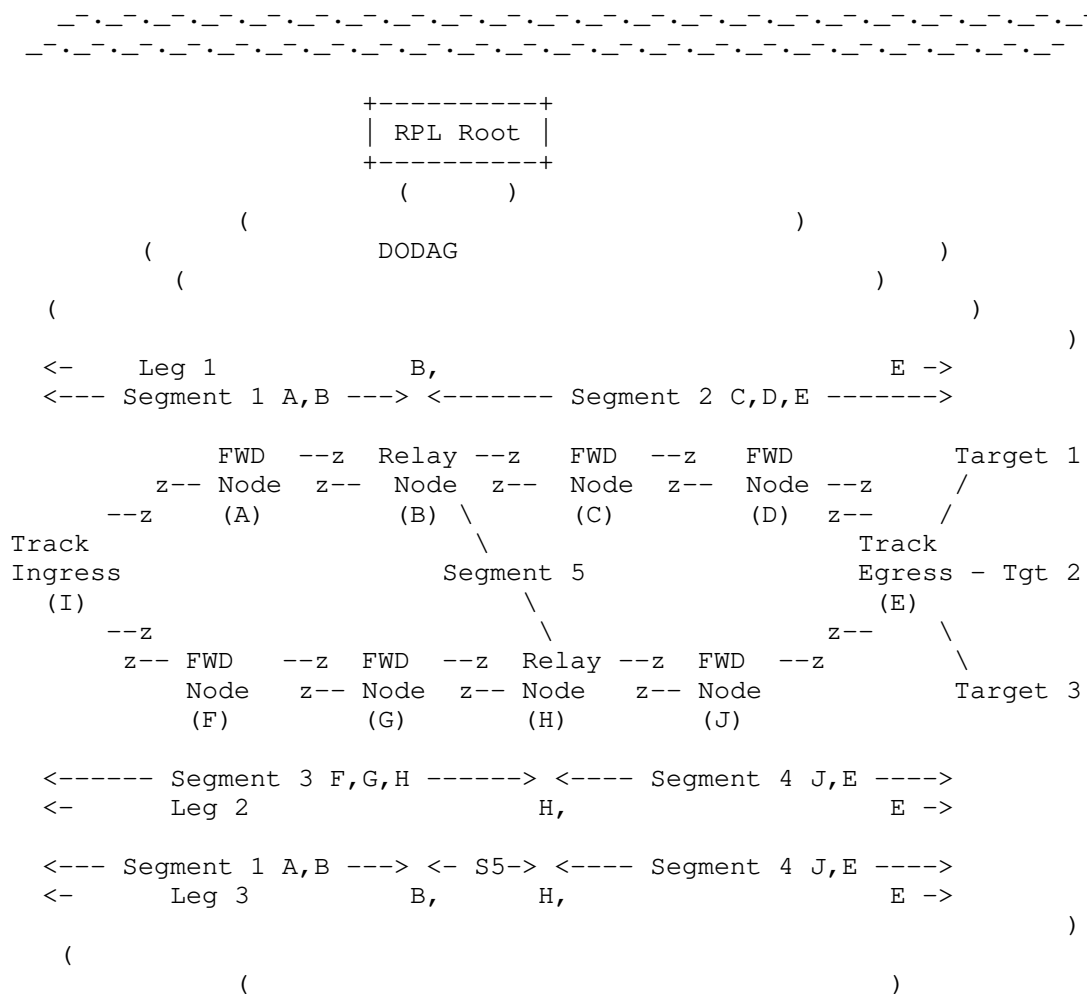


Figure 7: Segments and Tracks

Note that while this specification enables to build both Segments inside a Leg (aka East-West), such as Segment 2 above which is within Leg 1, and Inter-Leg Segments (aka North-South), such as Segment 2 above which joins Leg 1 and Leg 2, it does not signal to the Ingress which Inter-Leg Segments are available, so the use of North-South Segments and associated PAREO functions is currently limited. The only possibility available at this time is to define overlapping Legs

as illustrated in Figure 7, with Leg 3 that is congruent with Leg 1 till node B and congruent with Leg 2 from node H on, abstracting Segment 5 as an East-West Segment.

### 3.7. Scope and Expectations

#### 3.7.1. External Dependencies

This specification expects that the RPL Main DODAG is operated in RPL Non-Storing Mode to sustain the exchanges with the Root. Based on its comprehensive knowledge of the parent-child relationship, the Root can form an abstracted view of the whole DODAG topology. THIS RFC adds the capability for nodes to advertise additional sibling information to complement the topological awareness of the Root to be passed on to the PCE, and enable the PCE to build more / better paths that traverse those siblings.

P-Routes require resources such as routing table space in the routers and bandwidth on the links; the amount of state that is installed in each node must be computed to fit within the node's memory, and the amount of rerouted traffic must fit within the capabilities of the transmission links. The methods used to learn the node capabilities and the resources that are available in the devices and in the network are out of scope for THIS RFC. The method to capture and report the LLN link capacity and reliability statistics are also out of scope. They may be fetched from the nodes through network management functions or other forms of telemetry such as OAM.

#### 3.7.2. Positioning vs. Related IETF Standards

##### 3.7.2.1. Extending 6TiSCH

The "6TiSCH Architecture" [RFC9030] leverages a centralized model that is similar to that of "Deterministic Networking Architecture" [RFC8655], whereby the device resources and capabilities are exposed to an external controller which installs routing states into the network based on its own objective functions that reside in that external entity.

##### 3.7.2.2. Mapping to DetNet

DetNet Forwarding Nodes only understand the simple 1-to-1 forwarding sublayer transport operation along a segment whereas the more sophisticated Relay nodes can also provide service sublayer functions such as Replication and Elimination.

One possible mapping between DetNet and this specification is to signal the Relay Nodes as the hops of a Leg and the forwarding Nodes as the hops in a Segment that join the Relay nodes as illustrated in Figure 7.

#### 3.7.2.3. Leveraging PCE

With DetNet and 6TiSCH, the component of the controller that is responsible of computing routes is a PCE. The PCE computes its routes based on its own objective functions such as described in [RFC4655], and typically controls the routes using the PCE Protocol (PCEP) by [RFC5440]. While this specification expects a PCE and while PCEP might effectively be used between the Root and the PCE, the control protocol between the PCE and the Root is out of scope.

This specification also expects a single PCE with a full view of the network. Distributing the PCE function for a large network is out of scope. This specification uses the RPL Root as a proxy to the PCE. The PCE may be collocated with the Root, or may reside in an external Controller. In that case, the protocol between the Root and the PCE is out of scope and abstracted by / mapped to RPL inside the DODAG; one possibility is for the Root to transmit the RPL DAOs with the SIOs that detail the parent/child and sibling information.

The algorithm to compute the paths and the protocol used by the PCE and the metrics and link statistics involved in the computation are also out of scope. The effectiveness of the route computation by the PCE depends on the quality of the metrics that are reported from the RPL network. Which metrics are used and how they are reported is out of scope, but the expectation is that they are mostly of long-term, statistical nature, and provide visibility on link throughput, latency, stability and availability over relatively long periods.

#### 3.7.2.4. Providing for RAW

The RAW Architecture [RAW-ARCHI] extends the definition of Track, as being composed of East-West directional segments and North-South bidirectional segments, to enable additional path diversity, using Packet ARQ, Replication, Elimination, and Overhearing (PAREO) functions over the available paths, to provide a dynamic balance between the reliability and availability requirements of the flows and the need to conserve energy and spectrum. This specification prepares for RAW by setting up the Tracks, but only forms DODAGs, which are composed of aggregated end-to-end loose source routed Legs, joined by strict routed Segments, all oriented East-West.



The RAW Architecture defines a dataplane extension of the PCE called the Path Selection Engine (PSE), that adapts the use of the path redundancy within a Track to defeat the diverse causes of packet loss. The PSE controls the forwarding operation of the packets within a Track. This specification can use but does not impose a PSE and does not provide the policies that would select which packets are routed through which path within a Track, IOW, how the PSE may use the path redundancy within the Track. By default, the use of the available redundancy is limited to simple load balancing, and all the segments are East-West unidirectional only.

A Track may be set up to reduce the load around the Root, or to enable urgent traffic to flow more directly. This specification does not provide the policies that would decide which flows are routed through which Track. In a Non-Storing Mode RPL Instance, the Main DODAG provides a default route via the Root, and the Tracks provide more specific routes to the Track Targets.

#### 4. Extending existing RFCs

This section explains which changes are extensions to existing specifications, and which changes are amendments to existing specification. It is expected that extensions to existing specifications do not cause existing code on legacy 6LRs to malfunction, as the extensions will simply be ignored. New code is required for an extension. Those 6LRs will be unable to participate in the new mechanisms, but may also cause projected DAOs to be impossible to install. Amendments to existing specifications are situations where there are semantic changes required to existing code, and which may require new unit tests to confirm that legacy operations will continue unaffected.

##### 4.1. Extending RFC 6550

This specification Extends RPL [RPL] to enable the Root to install East-West routes inside a Main DODAG that is operated as Non-Storing Mode. The Root issues a Projected DAO (P-DAO) message (see Section 4.1.1) to the Track Ingress; the P-DAO message contains a new Via Information Option (VIO) that installs a strict or a loose sequence of hops to form respectively a Track Segment or a Track Leg.

The new P-DAO Request (PDR) is a new message detailed in Section 5.1. As per [RPL] section 6, if a node receives this message and it does not understand this new Code, then discards the message. When the root initiates to a node that it has not communicated with before, and to which it does not know if this specification has been implemented (by means such as capabilities), then the root SHOULD request a PDR-ACK.

A P-DAO Request (PDR) message enables a Track Ingress to request the Track from the Root. The resulting Track is also a DODAG for which the Track Ingress is the Root, the owner the address that serves as DODAGID and authoritative for the associated namespace from which the TrackID is selected. In the context of this specification, the installed route appears as a more specific route to the Track Targets, and the Track Ingress routes the packets towards the Targets via the Track using the longest match as usual.

To ensure that the PDR and P-DAO messages can flow at most times, it is RECOMMENDED that the nodes involved in a Track maintain multiple parents in the Main DODAG, advertise them all to the Root, and use them in turn to retry similar packets. It is also RECOMMENDED that the Root uses diverse source route paths to retry similar messages to the nodes in the Track.

#### 4.1.1. Projected DAO

Section 6 of [RPL] introduces the RPL Control Message Options (CMO), including the RPL Target Option (RTO) and Transit Information Option (TIO), which can be placed in RPL messages such as the destination Advertisement Object (DAO). A DAO message signals routing information to one or more Targets indicated in RTOs, providing one hop information at a time in the TIO.

THIS RFC Amends the specification of the DAO to create the P-DAO message. This Amended DAO is signaled with a new "Projected DAO" (P) flag, see Figure 8.

A Projected DAO (P-DAO) is a special DAO message generated by the Root to install a P-Route formed of multiple hops in its DODAG. This provides a RPL-based method to install the Tracks as expected by the 6TiSCH Architecture [RFC9030] as a collection of multiple P-Routes.

The Root MUST source the P-DAO message with its address that serves as DODAGID for the main DODAG. The receiver MUST NOT accept a P-DAO message that is not sent by the Root of its DODAG and MUST ignore such message silently.

The 'P' flag is encoded in bit position 2 (to be confirmed by IANA) of the Flags field in the DAO Base Object. The Root MUST set it to 1 in a Projected DAO message. Otherwise it MUST be set to 0. It is set to 0 in Legacy implementations as specified respectively in Sections 20.11 and 6.4 of [RPL].

The P-DAO is control plane signaling and should not be stuck behind high traffic levels. The expectation is that the P-DAO message is sent as high QoS level, above that of data traffic, typically with the Network Control precedence.

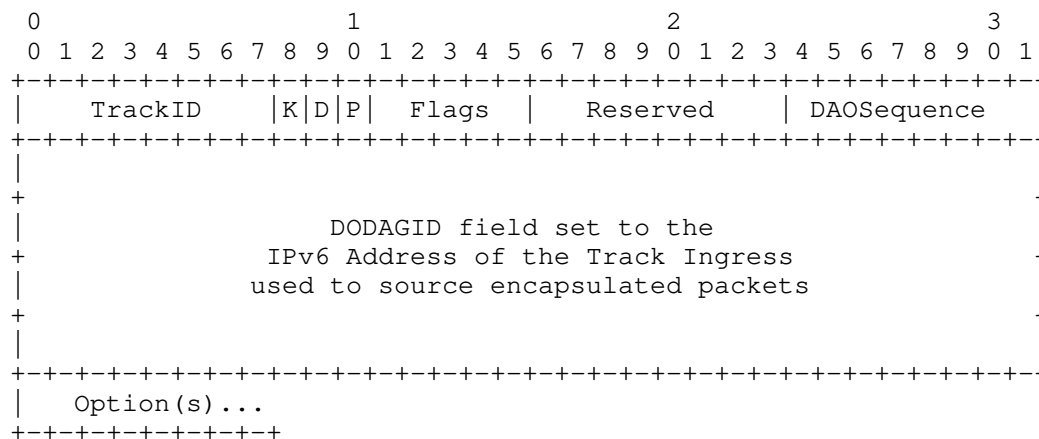


Figure 8: Projected DAO Base Object

New fields:

**TrackID:** The local or global RPLInstanceID of the DODAG that serves as Track, more in Section 6.3

**P:** 1-bit flag (position to be confirmed by IANA).

The 'P' flag is set to 1 by the Root to signal a Projected DAO, and it is set to 0 otherwise.

The D flag is set to one to signal that the DODAGID field is present. It may be set to zero if and only if the destination address of the P-DAO-ACK message is set to the IPv6 address that serves as DODAGID and it MUST be set to one otherwise, meaning that the DODAGID field MUST then be present.

In RPL Non-Storing Mode, the TIO and RTO are combined in a DAO message to inform the DODAG Root of all the edges in the DODAG, which are formed by the directed parent-child relationships. The DAO message signals to the Root that a given parent can be used to reach a given child. The P-DAO message generalizes the DAO to signal to the Track Ingress that a Track for which it is Root can be used to reach children and siblings of the Track Egress. In both cases, options may be factorized and multiple RTOs may be present to signal a collection of children that can be reached through the parent or the Track, respectively.

#### 4.1.2. Projected DAO-ACK

THIS RFC also Amends the DAO-ACK message. The new P flag signals the projected form.

The format of the P-DAO-ACK message is thus as illustrated in Figure 9:

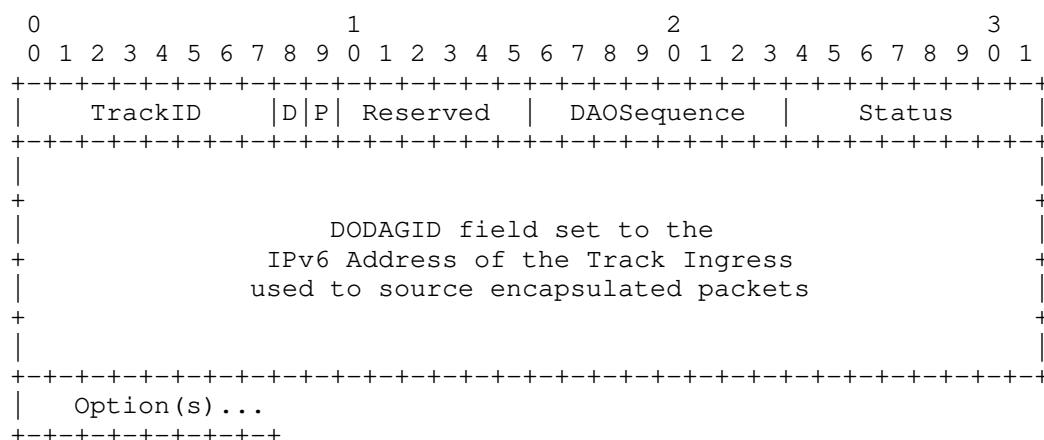


Figure 9: Projected DAO-ACK Base Object

New fields:

**TrackID:** The local or global RPLInstanceID of the DODAG that serves as Track, more in Section 6.3

**P:** 1-bit flag (position to be confirmed by IANA).

The 'P' flag is set to 1 by the Root to signal a Projected DAO, and it is set to 0 otherwise.

The D flag is set to one to signal that the DODAGID field is present. It may be set to zero if and only if the source address of the P-DAO-ACK message is set to the IPv6 address that serves as DODAGID and it MUST be set to one otherwise, meaning that the DODAGID field MUST then be present.

#### 4.1.3. Via Information Option

THIS RFC Extends the CMO to create new objects called the Via Information Options (VIO). The VIOs are the multihop alternative to the TIO, more in Section 5.3. One VIO is the stateful Storing Mode VIO (SM-VIO); an SM-VIO installs a strict hop-by-hop P-Route called a Track Segment. The other is the Non-Storing Mode VIO (NSM-VIO); the NSM-VIO installs a loose source-routed P-Route called a Track Leg at the Track Ingress, which uses that state to encapsulate a packet IPv6\_in\_IPv6 with a new Routing Header (RH) to the Track Egress, more in Section 6.7.

A P-DAO contains one or more RTOs to indicate the Target (destinations) that can be reached via the P-Route, followed by exactly one VIO that signals the sequence of nodes to be followed, more in Section 6. There are two modes of operation for the P-Routes, the Storing Mode and the Non-Storing Mode, see Section 6.4.2 and Section 6.4.3 respectively for more.

#### 4.1.4. Sibling Information Option

This specification Extends the CMO to create the Sibling Information Option (SIO). The SIO is used by a RPL Aware Node (RAN) to advertise a selection of its candidate neighbors as siblings to the Root, more in Section 5.4. The SIO is placed in DAO messages that are sent directly to the Root of the main DODAG.

#### 4.1.5. P-DAO Request

The set of RPL Control Messages is Extended to include the P-DAO Request (PDR) and P-DAO Request Acknowledgement (PDR-ACK). These two new RPL Control Messages enable an RPL-Aware Node to request the establishment of a Track between itself as the Track Ingress Node and a Track Egress. The node makes its request by sending a new P-DAO Request (PDR) Message to the Root. The Root confirms with a new PDR-ACK message back to the requester RAN, see Section 5.1 for more.

#### 4.1.6. Amending the RPI

Sending a Packet within a RPL Local Instance requires the presence of the abstract RPL Packet Information (RPI) described in section 11.2. of [RPL] in the outer IPv6 Header chain (see [RFC9008]). The RPI carries a local RPLInstanceID which, in association with either the source or the destination address in the IPv6 Header, indicates the RPL Instance that the packet follows.

This specification Amends [RPL] to create a new flag that signals that a packet is forwarded along a P-Route.

Projected-Route 'P': 1-bit flag. It is set to 1 in the RPI that is added in the encapsulation when a packet is sent over a Track. It is set to 0 when a packet is forwarded along the main Track, including when the packet follows a Segment that joins loose hops of the Main DODAG. The flag is not mutable en-route.

The encoding of the 'P' flag in native format is shown in Section 4.2 while the compressed format is indicated in Section 4.3.

#### 4.1.7. Additional Flag in the RPL DODAG Configuration Option

The DODAG Configuration Option is defined in Section 6.7.6 of [RPL]. Its purpose is extended to distribute configuration information affecting the construction and maintenance of the DODAG, as well as operational parameters for RPL on the DODAG, through the DODAG. This Option was originally designed with 4 bit positions reserved for future use as Flags.

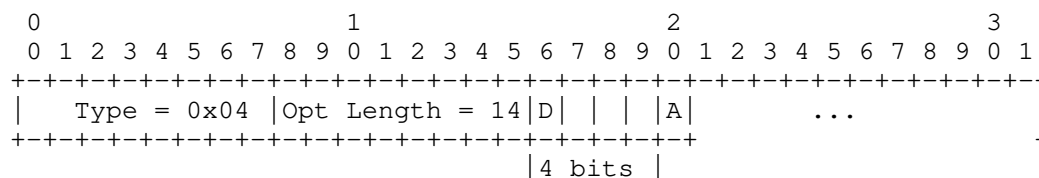


Figure 10: DODAG Configuration Option (Partial View)

This specification Amends the specification to define a new flag "Projected Routes Support" (D). The 'D' flag is encoded in bit position 0 of the reserved Flags in the DODAG Configuration Option (this is the most significant bit) (to be confirmed by IANA but there's little choice). It is set to 0 in legacy implementations as specified respectively in Sections 20.14 and 6.7.6 of [RPL].

The 'D' flag is set to 1 to indicate that this specification is enabled in the network and that the Root will install the requested Tracks when feasible upon a PDR message.

Section 4.1.2. of [RFC9008] updates [RPL] to indicate that the definition of the Flags applies to Mode of Operation values from zero (0) to six (6) only. For a MOP value of 7, the implementation MUST consider that the Root accepts PDR messages and will install Projected Routes.

The RPL DODAG Configuration option is typically placed in a DODAG Information Object (DIO) message. The DIO message propagates down the DODAG to form and then maintain its structure. The DODAG Configuration option is copied unmodified from parents to children.

[RPL] states that:

```
| Nodes other than the DODAG root MUST NOT modify this information
| when propagating the DODAG Configuration option.
```

Therefore, a legacy parent propagates the 'D' flag as set by the root, and when the 'D' flag is set to 1, it is transparently flooded to all the nodes in the DODAG.

#### 4.2. Extending RFC 6553

"The RPL Option for Carrying RPL Information in Data-Plane Datagrams" [RFC6553] describes the RPL Option for use among RPL routers to include the abstract RPL Packet Information (RPI) described in section 11.2. of [RPL] in data packets.

The RPL Option is commonly referred to as the RPI though the RPI is really the abstract information that is transported in the RPL Option. [RFC9008] updated the Option Type from 0x63 to 0x23.

This specification Amends the RPL Option to encode the 'P' flag as follows:

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
                                     +-----+-----+-----+
                                     | Option Type | Opt Data Len |
+-----+-----+-----+-----+-----+-----+-----+-----+
| O|R|F|P|0|0|0|0| RPLInstanceID | SenderRank |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     (sub-TLVs)                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 11: Amended RPL Option Format

Option Type: 0x23 or 0x63, see [RFC9008]

Opt Data Len: See [RFC6553]

'O', 'R' and 'F' flags: See [RFC6553]. Those flags MUST be set to 0 by the sender and ignored by the receiver if the 'P' flag is set.

Projected-Route 'P': 1-bit flag as defined in Section 4.1.6.

RPLInstanceID: See [RFC6553]. Indicates the TrackId if the 'P' flag is set, as discussed in Section 4.1.1.

SenderRank: See [RFC6553]. This field MUST be set to 0 by the sender and ignored by the receiver if the 'P' flag is set.

#### 4.3. Extending RFC 8138

The 6LoWPAN Routing Header [RFC8138] specification introduces a new IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) [RFC6282] dispatch type for use in 6LoWPAN route-over topologies, which initially covers the needs of RPL data packet compression.

Section 4 of [RFC8138] presents the generic formats of the 6LoWPAN Routing Header (6LoRH) with two forms, one Elective that can be ignored and skipped when the router does not understand it, and one Critical which causes the packet to be dropped when the router cannot process it. The 'E' Flag in the 6LoRH indicates its form. In order to skip the Elective 6LoRHs, their format imposes a fixed expression of the size, whereas the size of a Critical 6LoRH may be signaled in variable forms to enable additional optimizations.

When the [RFC8138] compression is used, the Root of the Main DODAG that sets up the Track also constructs the compressed routing header (SRH-6LoRH) on behalf of the Track Ingress, which saves the complexities of optimizing the SRH-6LoRH encoding in constrained code. The SRH-6LoRH is signaled in the NSM-VIO, in a fashion that it is ready to be placed as is in the packet encapsulation by the Track Ingress.

Section 6.3 of [RFC8138] presents the formats of the 6LoWPAN Routing Header of type 5 (RPI-6LoRH) that compresses the RPI for normal RPL operation. The format of the RPI-6LoRH is not suited for P-Routes since the O,R,F flags are not used and the Rank is unknown and ignored.



This specification extends [RFC8138] to introduce a new 6LoRH, the P-RPI-6LoRH that can be used in either Elective or Critical 6LoRH form, see Table 22 and Table 23 respectively. The new 6LoRH MUST be used as a Critical 6LoRH, unless an SRH-6LoRH is present and controls the routing decision, in which case it MAY be used in Elective form.

The P-RPI-6LoRH is designed to compress the RPI along RPL P-Routes. Its format is as follows:

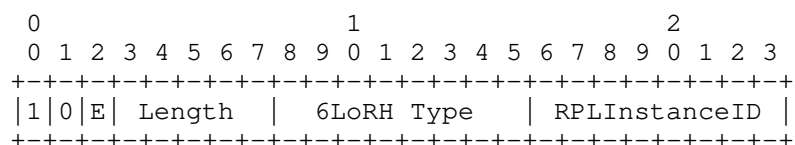


Figure 12: P-RPI-6LoRH Format

Type: IANA is requested to define the same value of the type for both Elective and Critical forms. A type of 8 is suggested.

Elective 'E': See [RFC8138]. The 'E' flag is set to 1 to indicate an Elective 6LoRH, meaning that it can be ignored when forwarding.

RPLInstanceID : In the context of this specification, the RPLInstanceID field signals the TrackID, see Section 3.4 and Section 6.3 .

Section 6.8 details how a a Track Ingress leverages the P-RPI-6LoRH Header as part of the encapsulation of a packet to place it into a Track.

## 5. New RPL Control Messages and Options

### 5.1. New P-DAO Request Control Message

The P-DAO Request (PDR) message is sent by a Node in the Main DODAG to the Root. It is a request to establish or refresh a Track where this node is Track Ingress, and signals whether an acknowledgment called PDR-ACK is requested or not. A positive PDR-ACK indicates that the Track was built and that the Roots commits to maintain the Track for the negotiated lifetime.

The main Root MAY indicate to the Track Ingress that the Track was terminated before its time and to do so, it MUST uses an asynchronous PDR-ACK with an negative status. A status of "Transient Failure" (see Section 11.10) is an indication that the PDR may be retried after a reasonable time that depends on the deployment. Other

negative status values indicate a permanent error; the tentative must be abandoned until a corrective action is taken at the application layer or through network management.

The source IPv6 address of the PDR signals the Track Ingress to-be of the requested Track, and the TrackID is indicated in the message itself. At least one RPL Target Option MUST be present in the message. If more than one RPL Target Option is present, the Root will provide a Track that reaches the first listed Target and a subset of the other Targets; the details of the subset selection are out of scope. The RTO signals the Track Egress, more in Section 6.2.

The RPL Control Code for the PDR is 0x09, to be confirmed by IANA. The format of PDR Base Object is as follows:

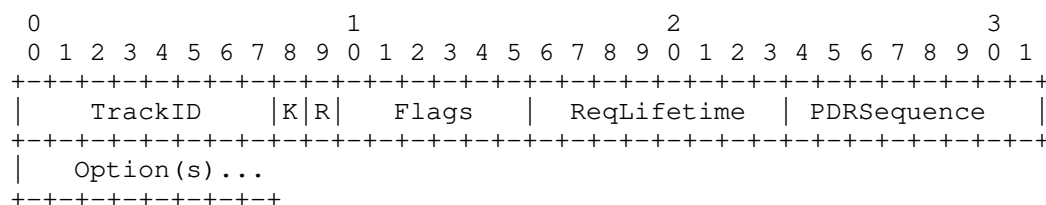


Figure 13: New P-DAO Request Format

**TrackID:** 8-bit field. In the context of this specification, the TrackID field signals the RPLInstanceID of the DODAG formed by the Track, see Section 3.4 and Section 6.3. To allocate a new Track, the Ingress Node must provide a value that is not in use at this time.

**K:** The 'K' flag is set to indicate that the recipient is expected to send a PDR-ACK back.

**R:** The 'R' flag is set to request a Complex Track for redundancy.

**Flags:** Reserved. The Flags field MUST be initialized to zero by the sender and MUST be ignored by the receiver

**ReqLifetime:** 8-bit unsigned integer. The requested lifetime for the Track expressed in Lifetime Units (obtained from the DODAG Configuration option).

A PDR with a fresher PDRSequence refreshes the lifetime, and a PDRLifetime of 0 indicates that the Track should be destroyed, e.g., when the application that requested the Track terminates.

**PDRSequence:** 8-bit wrapping sequence number, obeying the operation

in section 7.2 of [RPL]. The PDRSequence is used to correlate a PDR-ACK message with the PDR message that triggered it. It is incremented at each PDR message and echoed in the PDR-ACK by the Root.

## 5.2. New PDR-ACK Control Message

The new PDR-ACK is sent as a response to a PDR message with the 'K' flag set. The RPL Control Code for the PDR-ACK is 0x0A, to be confirmed by IANA. Its format is as follows:

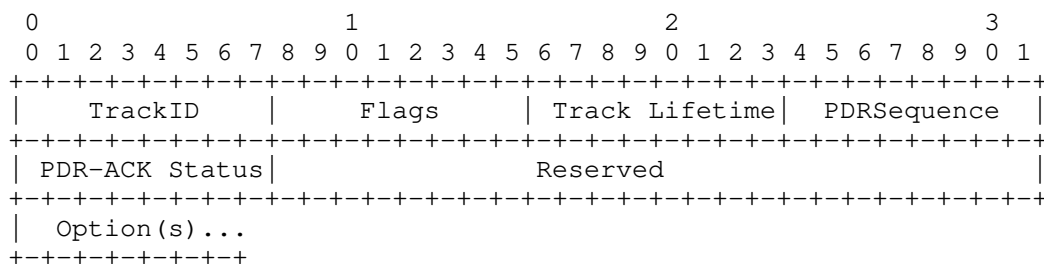


Figure 14: New PDR-ACK Control Message Format

**TrackID:** Set to the TrackID indicated in the TrackID field of the PDR messages that this replies to.

**Flags:** Reserved. The Flags field MUST initialized to zero by the sender and MUST be ignored by the receiver

**Track Lifetime:** Indicates that remaining Lifetime for the Track, expressed in Lifetime Units; the value of zero (0x00) indicates that the Track was destroyed or not created.

**PDRSequence:** 8-bit wrapping sequence number. It is incremented at each PDR message and echoed in the PDR-ACK.

**PDR-ACK Status:** 8-bit field indicating the completion. The PDR-ACK Status is substructured as indicated in Figure 15:

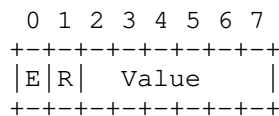


Figure 15: PDR-ACK status Format

**E:** 1-bit flag. Set to indicate a rejection. When not set, the

value of 0 indicates Success/Unqualified Acceptance and other values indicate "not an outright rejection".

R: 1-bit flag. Reserved, MUST be set to 0 by the sender and ignored by the receiver.

Status Value: 6-bit unsigned integer. Values depending on the setting of the 'E' flag, see Table 28 and Table 29.

Reserved: The Reserved field MUST be initialized to zero by the sender and MUST be ignored by the receiver

### 5.3. Via Information Options

A VIO signals the ordered list of IPv6 Via Addresses that constitutes the hops of either a Leg (using Non-Storing Mode) a Segment (using storing mode) of a Track. A Storing Mode P-DAO contains one Storing Mode VIO (SM-VIO) whereas a Non-Storing Mode P-DAO contains one Non-Storing Mode VIO (NSM-VIO)

The duration of the validity of a VIO is indicated in a Segment Lifetime field. A P-DAO message that contains a VIO with a Segment Lifetime of zero is referred as a No-Path P-DAO.

The VIO contains one or more SRH-6LoRH header(s), each formed of a SRH-6LoRH head and a collection of compressed Via Addresses, except in the case of a Non-Storing Mode No-Path P-DAO where the SRH-6LoRH header is not present.

In the case of a SM-VIO, or if [RFC8138] is not used in the data packets, then the Root MUST use only one SRH-6LoRH per Via Information Option, and the compression is the same for all the addresses, as shown in Figure 16, for simplicity.

In case of an NSM-VIO and if [RFC8138] is in use in the Main DODAG, the Root SHOULD optimize the size of the NSM-VIO if using different SRH-6LoRH Types make the VIO globally shorter; this means that more than one SRH-6LoRH may be present.

The format of the Via Information Options is as follows:

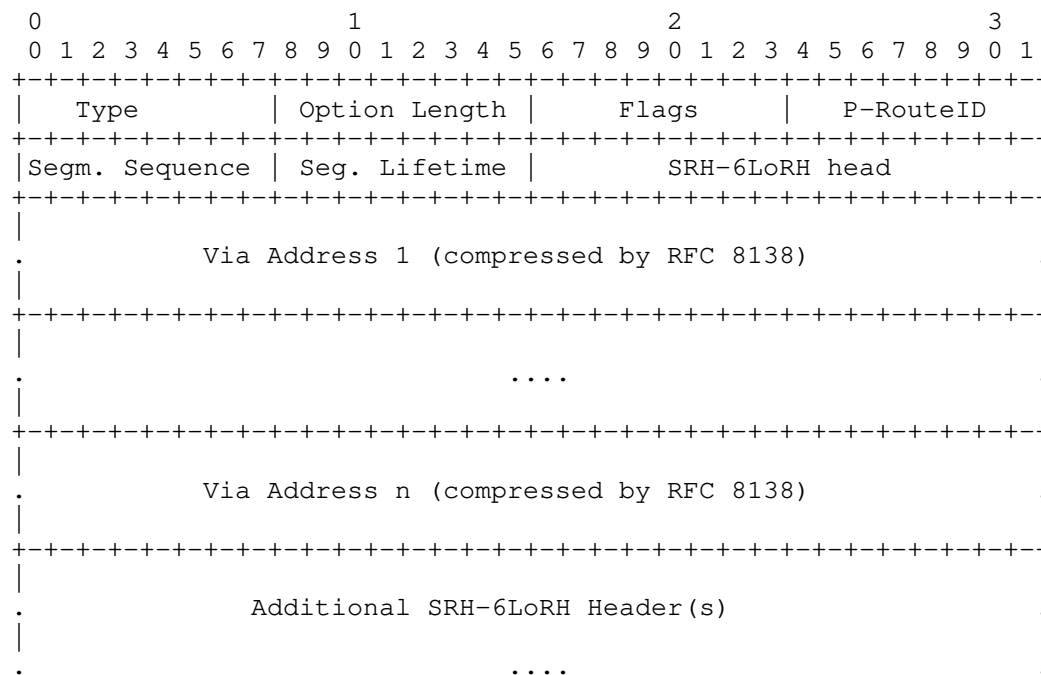


Figure 16: VIO format (uncompressed form)

Option Type: 0x0E for SM-VIO, 0x0F for NSM-VIO (to be confirmed by IANA), see Table 26

Option Length: 8-bit unsigned integer, representing the length in octets of the option, not including the Option Type and Length fields, see section 6.7.1. of [RPL]; the Option Length is variable, depending on the number of Via Addresses and the compression applied.

P-RouteID: 8-bit field that identifies a component of a Track or the Main DODAG as indicated by the TrackID field. The value of 0 is used to signal a Serial Track, i.e., made of a single segment/Leg. In an SM-VIO, the P-RouteID indicates an actual Segment. In an NSM-VIO, it indicates a Leg, that is a serial subTrack that is added to the overall topology of the Track.

Segment Sequence: 8-bit unsigned integer. The Segment Sequence obeys the operation in section 7.2 of [RPL] and the lollipop starts at 255.

When the Root of the DODAG needs to refresh or update a Segment in a Track, it increments the Segment Sequence individually for that Segment.

The Segment information indicated in the VIO deprecates any state for the Segment indicated by the P-RouteID within the indicated Track and sets up the new information.

A VIO with a Segment Sequence that is not as fresh as the current one is ignored.

A VIO for a given DODAGID with the same (TrackID, P-RouteID, Segment Sequence) indicates a retry; it MUST NOT change the Segment and MUST be propagated or answered as the first copy.

Segment Lifetime: 8-bit unsigned integer. The length of time in Lifetime Units (obtained from the Configuration option) that the Segment is usable.

The period starts when a new Segment Sequence is seen. The value of 255 (0xFF) represents infinity. The value of zero (0x00) indicates a loss of reachability.

SRH-6LoRH head: The first 2 bytes of the (first) SRH-6LoRH as shown in Figure 6 of [RFC8138]. As an example, a 6LoRH Type of 4 means that the VIA Addresses are provided in full with no compression.

Via Address: An IPv6 ULA or GUA of a node along the Segment. The VIO contains one or more IPv6 Via Addresses listed in the datapath order from Ingress to Egress. The list is expressed in a compressed form as signaled by the preceding SRH-6LoRH header.

In a Storing Mode P-DAO that updates or removes a section of an already existing Segment, the list in the SM-VIO may represent only the section of the Segment that is being updated; at the extreme, the SM-VIO updates only one node, in which case it contains only one IPv6 address. In all other cases, the list in the VIO MUST be complete.

In the case of an SM-VIO, the list indicates a sequential (strict) path through direct neighbors, the complete list starts at Ingress and ends at Egress, and the nodes listed in the VIO, including the Egress, MAY be considered as implicit Targets.

In the case of an NSM-VIO, the complete list can be loose and excludes the Ingress node, starting at the first loose hop and ending at a Track Egress; the Track Egress MUST be considered as an implicit Target, so it MUST NOT be signaled in a RPL Target Option.

#### 5.4. Sibling Information Option

The Sibling Information Option (SIO) provides indication on siblings that could be used by the Root to form P-Routes. One or more SIO(s) may be placed in the DAO messages that are sent to the Root in Non-Storing Mode.

To advertise a neighbor node, the router MUST have an active Address Registration from that sibling using [RFC8505], for an address (ULA or GUA) that serves as identifier for the node. If this router also registers an address to that sibling, and the link has similar properties in both directions, only the router with the lowest Interface ID in its registered address needs report the SIO, with the B flag set, and the Root will assume symmetry.

The SIO carries a flag (B) that is set when similar performances can be expected both directions, so the routing can consider that the information provided for one direction is valid for both. If the SIO is effectively received from both sides then the B flag MUST be ignored. The policy that describes the performance criteria, and how they are asserted is out of scope. In the absence of an external protocol to assert the link quality, the flag SHOULD NOT be set.

The format of the SIO is as follows:

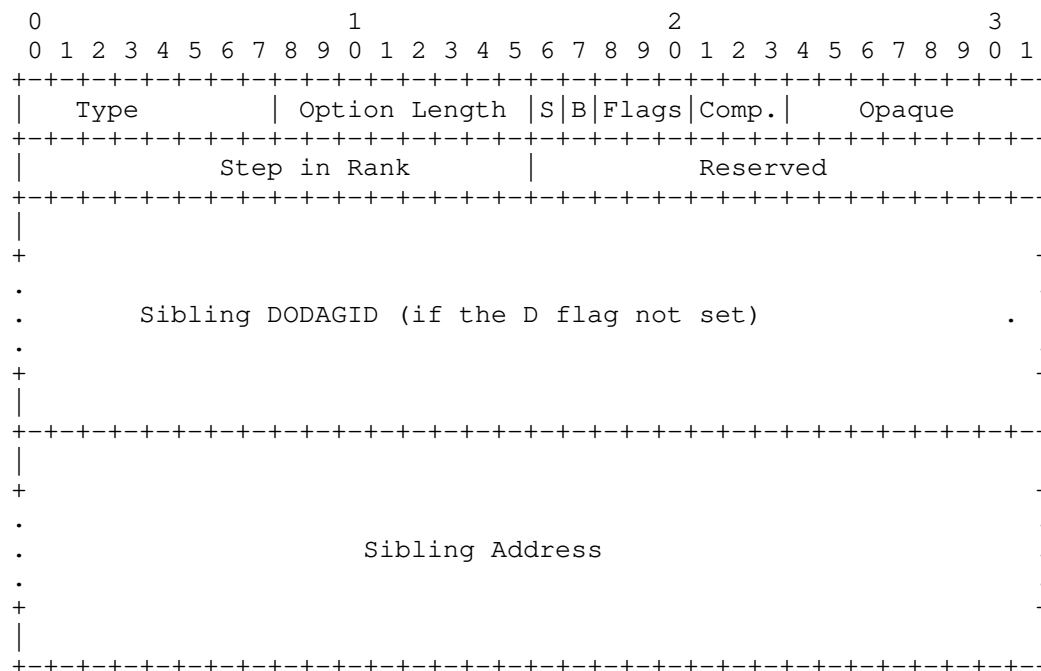


Figure 17: Sibling Information Option Format

Option Type: 0x10 for SIO (to be confirmed by IANA), see =Table 26

Option Length: 8-bit unsigned integer, representing the length in octets of the option, not including the Option Type and Length fields, see section 6.7.1. of [RPL].

Reserved for Flags: MUST be set to zero by the sender and MUST be ignored by the receiver.

B: 1-bit flag that is set to indicate that the connectivity to the sibling is bidirectional and roughly symmetrical. In that case, only one of the siblings may report the SIO for the hop. If 'B' is not set then the SIO only indicates connectivity from the sibling to this node, and does not provide information on the hop from this node to the sibling.

S: 1-bit flag that is set to indicate that sibling belongs to the same DODAG. When not set, the Sibling DODAGID is indicated.

Flags: Reserved. The Flags field MUST be initialized to zero by the sender and MUST be ignored by the receiver



Opaque: MAY be used to carry information that the node and the Root understand, e.g., a particular representation of the Link properties such as a proprietary Link Quality Information for packets received from the sibling. An industrial Alliance that uses RPL for a particular use / environment MAY redefine the use of this field to fit its needs.

Compression Type: 3-bit unsigned integer. This is the SRH-6LoRH Type as defined in figure 7 in section 5.1 of [RFC8138] that corresponds to the compression used for the Sibling Address and its DODAGID if resent. The Compression reference is the Root of the Main DODAG.

Step in Rank: 16-bit unsigned integer. This is the Step in Rank [RPL] as computed by the Objective Function between this node and the sibling, that reflects the abstract Rank increment that would be computed by the OF if the sibling was the preferred parent.

Reserved: The Reserved field MUST be initialized to zero by the sender and MUST be ignored by the receiver

Sibling DODAGID: 2 to 16 bytes, the DODAGID of the sibling in a [RFC8138] compressed form as indicated by the Compression Type field. This field is present if and only if the D flag is not set.

Sibling Address: 2 to 16 bytes, an IPv6 Address of the sibling, with a scope that MUST be make it reachable from the Root, e.g., it cannot be a Link Local Address. The IPv6 address is encoded in the [RFC8138] compressed form indicated by the Compression Type field.

An SIO MAY be immediately followed by a DAG Metric Container. In that case the DAG Metric Container provides additional metrics for the hop from the Sibling to this node.

## 6. Root Initiated Routing State

### 6.1. RPL Network Setup

To avoid the need of Path MTU Discovery, 6LoWPAN links are normally defined with a MTU of 1280 (see section 4 of [6LoWPAN]). Injecting packets in a Track typically involves an IP-in-IP encapsulation and additional IPv6 Extension Headers. This may cause a fragmentation if the resulting packets exceeds the MTU that is defined for the RPL domain.

Though fragmentation is possible in a 6LoWPAN LLN, e.g., using [6LoWPAN], [RFC8930], and/or [RFC8931], it is RECOMMENDED to allow an MTU that is larger than 1280 in the main DODAG and allows for the additional headers while exposing only 1280 to the 6LoWPAN Nodes.

## 6.2. Requesting a Track

This specification introduces the PDR message, used by an LLN node to request the formation of a new Track for which this node is Ingress. Note that the namespace for the TrackID is owned by the Ingress node, and in the absence of a PDR, there must be some procedure for the Root to assign TrackIDs in that namespace while avoiding collisions, more in Section 6.3.

The PDR signals the desired TrackID and the duration for which the Track should be established. Upon a PDR, the Root MAY install the Track as requested, in which case it answers with a PDR-ACK indicating the granted Track Lifetime. All the Segments MUST be of a same mode, either Storing or Non-Storing. All the Segments MUST be created with the same TrackID and the same DODAGID signaled in the P-DAO.

The Root designs the Track as it sees best, and updates / changes the Segments overtime to serve the Track as needed. Note that there is no protocol element to notify to the requesting Track Ingress when changes happen deeper down the Track, so they are transparent to the Track Ingress. If the main Root cannot maintain an expected service level, then it needs to tear down the Track completely. The Segment Lifetime in the P-DAO messages does not need to be aligned to the Requested Lifetime in the PDR, or between P-DAO messages for different Segments. The Root may use shorter lifetimes for the Segments and renew them faster than the Track is, or longer lifetimes in which case it will need to tear down the Segments if the Track is not renewed.

When the Track Lifetime that was returned in the PDR-ACK is close to elapse - vs. the trip time from the node to the Root, the requesting node SHOULD resend a PDR using the TrackID in the PDR-ACK to extend the lifetime of the Track, else the Track will time out and the Root will tear down the whole structure.

If the Track fails and cannot be restored, the Root notifies the requesting node asynchronously with a PDR-ACK with a Track Lifetime of 0, indicating that the Track has failed, and a PDR-ACK Status indicating the reason of the fault.

### 6.3. Identifying a Track

RPL defines the concept of an Instance to signal an individual routing topology, and multiple topologies can coexist in the same network. The RPLInstanceID is tagged in the RPI of every packet to signal which topology the packet actually follows.

This draft leverages the RPL Instance model as follows:

- \* The Root MAY use P-DAO messages to add better routes in the main (Global) RPL Instance in conformance with the routing objectives in that Instance.

To achieve this, the Root MAY install a Segment along a path down the main Non-Storing Mode DODAG. This enables a loose source routing and reduces the size of the Routing Header, see Section 3.3.1. The Root MAY also install a Track Leg across the Main DODAG to complement the routing topology.

When adding a P-Route to the RPL Main DODAG, the Root MUST set the RPLInstanceID field of the P-DAO Base Object (see section 6.4.1. of [RPL]) to the RPLInstanceID of the Main DODAG, and MUST NOT use the DODAGID field. A P-Route provides a longer match to the Target Address than the default route via the Root, so it is preferred.

- \* The Root MAY also use P-DAO messages to install a Track as an independent routing topology (say, Traffic Engineered) to achieve particular routing characteristics from an Ingress to an Egress Endpoints. To achieve this, the Root MUST set up a local RPL Instance (see section 5 of [RPL]), and the Local RPLInstanceID serves as TrackID. The TrackID MUST be unique for the IPv6 ULA or GUA of the Track Ingress that serves as DODAGID for the Track.

This way, a Track is uniquely identified by the tuple (DODAGID, TrackID) where the TrackID is always represented with the D flag set to 0 (see also section 5.1. of [RPL]), indicating when used in an RPI that the source address of the IPv6 packet signals the DODAGID.

The P-DAO Base Object MUST indicate the tuple (DODAGID, TrackID) that identifies the Track as shown in Figure 8, and the P-RouteID that identifies the P-Route MUST be signaled in the VIO as shown in Figure 16.

The Track Ingress is the Root of the DODAG ID formed by the local RPL Instance. It owns the namespace of its TrackIDs, so it can pick any unused value to request a new Track with a PDR. In a

particular deployment where PDR are not used, a portion of the namespace can be administratively delegated to the main Root, meaning that the main Root is authoritative for assigning the TrackIDs for the Tracks it creates.

With this specification, the Root is aware of all the active Tracks, so it can also pick any unused value to form Tracks without a PDR. To avoid a collision of the Root and the Track Ingress picking the same value at the same time, it is RECOMMENDED that the Track Ingress starts allocating the ID value of the Local RPLInstanceID (see section 5.1. of [RPL]) used as TrackIDs with the value 0 incrementing, while the Root starts with 63 decrementing.

#### 6.4. Installing a Track

A Serial Track can be installed by a single P-Route that signals the sequence of consecutive nodes, either in Storing Mode as a single-Segment Track, or in Non-Storing Mode as a single-Leg Track. A single-Leg Track can be installed as a loose Non-Storing Mode P-Route, in which case the next loose entry must recursively be reached over a Serial Track.

A Complex Track can be installed as a collection of P-Routes with the same DODAGID and Track ID. The Ingress of a Non-Storing Mode P-Route is the owner and Root of the DODAGID. The Ingress of a Storing Mode P-Route must be either the owner of the DODAGID, or a hop of a Leg of the same Track. In the latter case, the Targets of the P-Route must include the next hop of the Leg if there is one, to ensure forwarding continuity. In the case of a Complex Track, each Segment is maintained independently and asynchronously by the Root, with its own lifetime that may be shorter, the same, or longer than that of the Track.

A route along a Track for which the TrackID is not the RPLInstanceID of the Main DODAG MUST be installed with a higher precedence than the routes along the Main DODAG, meaning that:

- \* Longest match MUST be the prime comparison for routing.
- \* In case of equal length match, the route along the Track MUST be preferred vs. the one along the Main DODAG.
- \* There SHOULD NOT be 2 different Tracks leading to the same Target from same Ingress node, unless there's a policy for selecting which packets use which Track; such policy is out of scope.

- \* A packet that was routed along a Track MUST NOT be routed along the main DODAG again; if the destination is not reachable as a neighbor by the node where the packet exits the Track then the packet MUST be dropped.

#### 6.4.1. Signaling a Projected Route

This draft adds a capability whereby the Root of a main RPL DODAG installs a Track as a collection of P-Routes, using a Projected-DAO (P-DAO) message for each individual Track Leg or Segment. The P-DAO signals a collection of Targets in the RPL Target Option(s) (RTO). Those Targets can be reached via a sequence of routers indicated in a VIO.

Like a classical DAO message, a P-DAO causes a change of state only if it is "new" per section 9.2.2. "Generation of DAO Messages" of the RPL specification [RPL]; this is determined using the Segment Sequence information from the VIO as opposed to the Path Sequence from a TIO. Also, a Segment Lifetime of 0 in a VIO indicates that the P-Route associated to the Segment is to be removed. There are two Modes of operation for the P-Routes, the Storing and the Non-Storing Modes.

A P-DAO message MUST be sent from the address of the Root that serves as DODAGID for the Main DODAG. It MUST contain either exactly one sequence of one or more RTOs followed one VIO, or any number of sequences of one or more RTOs followed by one or more TIOs. The former is the normal expression for this specification, where as the latter corresponds to the variation for lesser constrained environments described in Section 7.2.

A P-DAO that creates or updates a Track Leg MUST be sent to a GUA or a ULA of the Ingress of the Leg; it must contain the full list of hops in the Leg unless the Leg is being removed. A P-DAO that creates a new Track Segment MUST be sent to a GUA or a ULA of the Segment Egress and MUST signal the full list of hops in Segment; a P-DAO that updates (including deletes) a section of a Segment MUST be sent to the first node after the modified Segment and signal the full list of hops in the section starting at the node that immediately precedes the modified section.

In Non-Storing Mode, as discussed in Section 6.4.3, the Root sends the P-DAO to the Track Ingress where the source-routing state is applied, whereas in Storing Mode, the P-DAO is sent to the last node on the installed path and forwarded in the reverse direction, installing a Storing Mode state at each hop, as discussed in Section 6.4.2. In both cases the Track Ingress is the owner of the Track, and it generates the P-DAO-ACK when the installation is successful.

If the 'K' Flag is present in the P-DAO, the P-DAO must be acknowledged using a DAO-ACK that is sent back to the address of the Root from which the P-DAO was received. In most cases, the first node of the Leg, Segment, or updated section of the Segment is the node that sends the acknowledgment. The exception to the rule is when an intermediate node in a Segment fails to forward a Storing Mode P-DAO to the previous node in the SM-VIO.

In a No-Path Non-Storing Mode P-DAO, the SRH-6LoRH MUST NOT be present in the NSM-VIO; the state in the Ingress is erased regardless. In all other cases, a VIO MUST contain at least one Via Address, and a Via Address MUST NOT be present more than once, which would create a loop.

A node that processes a VIO MAY verify whether one of these conditions happen, and when so, it MUST ignore the P-DAO and reject it with a RPL Rejection Status of "Error in VIO" in the DAO-ACK, see Section 11.16.

Other errors than those discussed explicitly that prevent the installing the route are acknowledged with a RPL Rejection Status of "Unqualified Rejection" in the DAO-ACK.

#### 6.4.2. Installing a Track Segment with a Storing Mode P-Route

As illustrated in Figure 18, a Storing Mode P-DAO installs a route along the Segment signaled by the SM-VIO towards the Targets indicated in the Target Options. The Segment is to be included in a DODAG indicated by the P-DAO Base Object, that may be the one formed by the RPL Main DODAG, or a Track associated with a local RPL Instance.

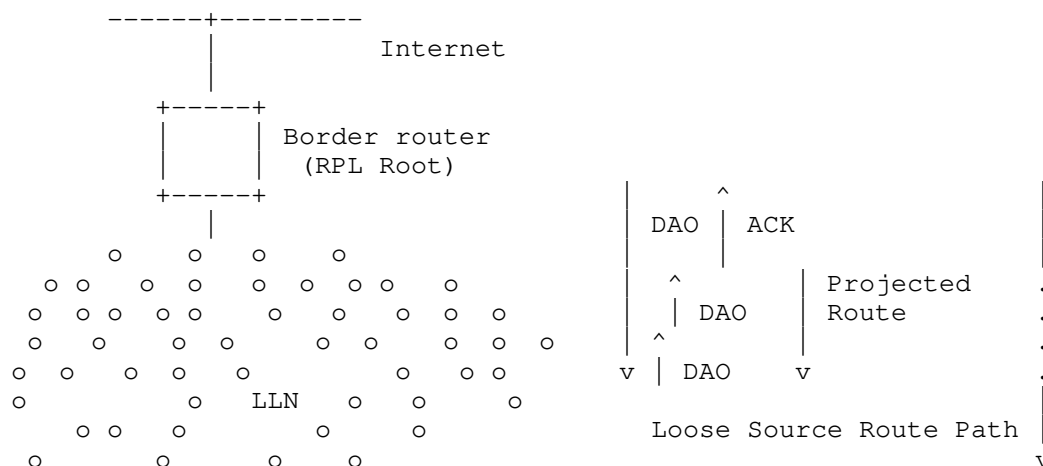


Figure 18: Projecting a route

In order to install the relevant routing state along the Segment , the Root sends a unicast P-DAO message to the Track Egress router of the routing Segment that is being installed. The P-DAO message contains a SM-VIO with the strict sequence of Via Addresses. The SM-VIO follows one or more RTOs indicating the Targets to which the Track leads. The SM-VIO contains a Segment Lifetime for which the state is to be maintained.

The Root sends the P-DAO directly to the Egress node of the Segment. In that P-DAO, the destination IP address matches the last Via Address in the SM-VIO. This is how the Egress recognizes its role. In a similar fashion, the Segment Ingress node recognizes its role as it matches first Via Address in the SM-VIO.

The Egress node of the Segment is the only node in the path that does not install a route in response to the P-DAO; it is expected to be already able to route to the Target(s) based on its existing tables. If one of the Targets is not known, the node MUST answer to the Root with a DAO-ACK listing the unreachable Target(s) in an RTO and a rejection status of "Unreachable Target".

If the Egress node can reach all the Targets, then it forwards the P-DAO with unchanged content to its predecessor in the Segment as indicated in the list of Via Information options, and recursively the message is propagated unchanged along the sequence of routers indicated in the P-DAO, but in the reverse order, from Egress to Ingress.

The address of the predecessor to be used as destination of the propagated DAO message is found in the Via Address the precedes the one that contain the address of the propagating node, which is used as source of the message.

Upon receiving a propagated DAO, all except the Egress router MUST install a route towards the DAO Target(s) via their successor in the SM-VIO. A router that cannot store the routes to all the Targets in a P-DAO MUST reject the P-DAO by sending a DAO-ACK to the Root with a Rejection Status of "Out of Resources" as opposed to forwarding the DAO to its predecessor in the list. The router MAY install additional routes towards the VIA Addresses that are the SM-VIO after self, if any, but in case of a conflict or a lack of resource, the route(s) to the Target(s) are the ones that must be installed in priority.

If a router cannot reach its predecessor in the SM-VIO, the router MUST send the DAO-ACK to the Root with a Rejection Status of "Predecessor Unreachable".

The process continues till the P-DAO is propagated to Ingress router of the Segment, which answers with a DAO-ACK to the Root. The Root always expects a DAO-ACK, either from the Track Ingress with a positive status or from any node along the segment with a negative status. If the DAO-ACK is not received, the Root may retry the DAO with the same TID, or tear down the route.

#### 6.4.3. Installing a Track Leg with a Non-Storing Mode P-Route

As illustrated in Figure 19, a Non-Storing Mode P-DAO installs a source-routed path within the Track indicated by the P-DAO Base Object, towards the Targets indicated in the Target Options. The source-routed path requires a Source-Routing header which implies an IP-in-IP encapsulation to add the SRH to an existing packet. It is sent to the Track Ingress which creates a tunnel associated with the Track, and connected routes over the tunnel to the Targets in the RTO. The tunnel encapsulation MUST incorporate a routing header via the list addresses listed in the VIO in the same order. The content of the NSM-VIO starting at the first SRH-6LoRH header MUST be used verbatim by the Track Ingress when it encapsulates a packet to forward it over the Track.



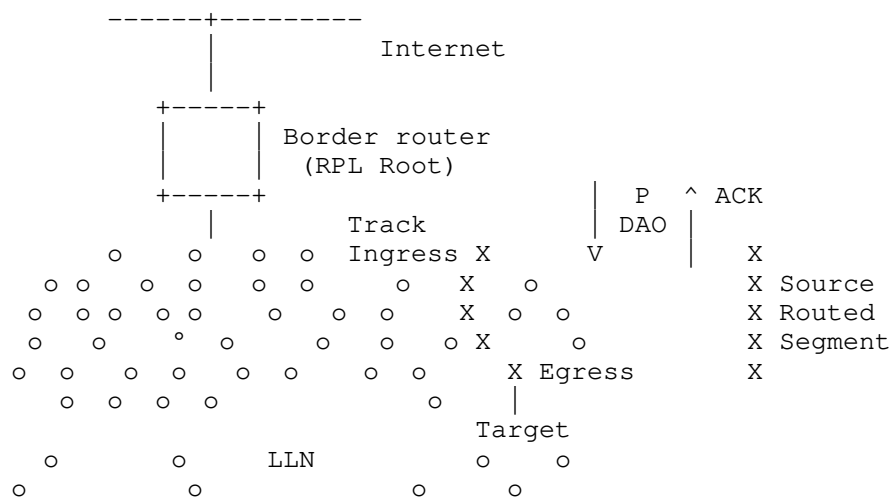


Figure 19: Projecting a Non-Storing Route

The next entry in the source-routed path must be either a neighbor of the previous entry, or reachable as a Target via another P-Route, either Storing or Non-Storing, which implies that the nested P-Route has to be installed before the loose sequence is, and that P-Routes must be installed from the last to the first along the datapath. For instance, a Segment of a Track must be installed before the Leg(s) of the same Track that use it, and stitched Segments must be installed in order from the last that reaches to the Targets to the first.

If the next entry in the loose sequence is reachable over a Storing Mode P-Route, it MUST be the Target of a Segment and the Ingress of a next segment, both already setup; the segments are associated with the same Track, which avoids the need of an additional encapsulation. For instance, in Section 3.5.1.3, Segments A==>B-to-C and C==>D==>E-to-F must be installed with Storing Mode P-DAO messages 1 and 2 before the Track A-->C-->E-to-F that joins them can be installed with Non-Storing Mode P-DAO 3.

Conversely, if it is reachable over a Non-Storing Mode P-Route, the next loose source-routed hop of the inner Track is a Target of a previously installed Track and the Ingress of a next Track, which requires a de- and a re-encapsulation when switching the outer Tracks that join the loose hops. This is exemplified in Section 3.5.2.3 where Non-Storing Mode P-DAO 1 and 2 install strict Tracks that Non-Storing Mode P-DAO 3 joins as a super Track. In such a case, packets are subject to double IP-in-IP encapsulation.

### 6.5. Tearing Down a P-Route

A P-DAO with a lifetime of 0 is interpreted as a No-Path DAO and results in cleaning up existing state as opposed to refreshing an existing one or installing a new one. To tear down a Track, the Root must tear down all the Track Segments and Legs that compose it one by one.

Since the state about a Leg of a Track is located only on the Ingress Node, the Root cleans up the Leg by sending an NSM-VIO to the Ingress indicating the TrackID and the P-RouteID of the Leg being removed, a Segment Lifetime of 0 and a newer Segment Sequence. The SRH-6LoRH with the Via Addresses in the NSM-VIO are not needed; it SHOULD not be placed in the message and MUST be ignored by the receiver. Upon that NSM-VIO, the Ingress node removes all state for that Track if any, and replies positively anyway.

The Root cleans up a section of a Segment by sending an SM-VIO to the last node of the Segment, with the TrackID and the P-RouteID of the Segment being updated, a Segment Lifetime of zero (0) and a newer Segment Sequence. The Via Addresses in the SM-VIO indicates the section of the Segment being modified, from the first to the last node that is impacted. This can be the whole Segment if it is totally removed, or a sequence of one or more nodes that have been bypassed by a Segment update.

The No-Path P-DAO is forwarded normally along the reverse list, even if the intermediate node does not find a Segment state to clean up. This results in cleaning up the existing Segment state if any, as opposed to refreshing an existing one or installing a new one.

### 6.6. Maintaining a Track

Repathing a Track Segment or Leg may cause jitter and packet misordering. For critical flows that require timely and/or in-order delivery, it might be necessary to deploy the PAREO functions [RAW-ARCHI] over a highly redundant Track. This specification allows to use more than one Leg for a Track, and 1+N packet redundancy.

This section provides the steps to ensure that no packet is lost due to the operation itself. This is ensured by installing the new section from its last node to the first, so when an intermediate node installs a route along the new section, all the downstream nodes in the section have already installed their own. The disabled section is removed when the packets in-flight are forwarded along the new section as well.

#### 6.6.1. Maintaining a Track Segment

To modify a section of a Segment between a first node and a second, downstream node (which can be the Ingress and Egress), while conserving those nodes in the Segment, the Root sends an SM-VIO to the second node indicating the sequence of nodes in the new section of the Segment. The SM-VIO indicates the TrackID and the P-RouteID of the Segment being updated, and a newer Segment Sequence. The P-DAO is propagated from the second to the first node and on the way, it updates the state on the nodes that are common to the old and the new section of the Segment and creates a state in the new nodes.

When the state is updated in an intermediate node, that node might still receive packets that were in flight from the Ingress to self over the old section of the Segment. Since the remainder of the Segment is already updated, the packets are forwarded along the new version of the Segment from that node on.

After a reasonable time to enable the deprecated sections to empty, the Root tears down the remaining section(s) of the old segments are torn down as described in Section 6.5.

#### 6.6.2. Maintaining a Track Leg

This specification allows the Root to add Legs to a Track by sending a Non-Storing Mode P-DAO to the Ingress associated to the same TrackID, and a new Segment ID. If the Leg is loose, then the Segments that join the hops must be created first. It makes sense to add a new Leg before removing one that is becoming excessively lossy, and switch to the new Leg before removing the old. Dropping a Track before the new one is installed would reroute the traffic via the root; this may augment the latency beyond acceptable thresholds, and load the network near the root. This may also cause loops in the case of stitched Tracks; the packets that cannot be injected in the second Track may be routed back at reinjected at the Ingress of the first.

It is also possible to update a Track Leg by sending a Non-Storing Mode P-DAO to the Ingress with the same Segment ID, an incremented Segment Sequence, and the new complete list of hops in the NSM-VIO. Updating a live Leg means changing one or more of the intermediate loose hops, and involves laying out new Segments from and to the new loose hops before the NSM-VIO for the new Leg is issued.

Packets that are in flight over the old version of the Track Leg still follow the old source route path over the old Segments. After a reasonable time to enable the deprecated Segments to empty, the Root tears down those Segments as described in Section 6.5.

## 6.7. Encapsulating and Forwarding Along a Track

When injecting a packet in a Track, the Ingress router must encapsulate the packet using IP-in-IP to add the Source Routing Header with the final destination set to the Track Egress.

All properties of a Track operations are inherited from the main RPL Instance that is used to install the Track. For instance, the use of compression per [RFC8138] is determined by whether it is used in the RPL Main DODAG, e.g., by setting the "T" flag [RFC9035] in the RPL configuration option.

The Track Ingress that places a packet in a Track encapsulates it with an IP-in-IP header, a Routing Header, and an IPv6 Hop-by-Hop Option Header that contains the RPL Packet Information (RPI) as follows:

- \* In the uncompressed form the source of the packet is the address that this router uses as DODAGID for the Track, the destination is the first Via Address in the NSM-VIO, and the RH is a Source Routing Header (SRH) [RFC6554] that contains the list of the remaining Via Addresses terminating by the Track Egress.
- \* The preferred alternate in a network where 6LoWPAN Header Compression [RFC6282] is used is to leverage "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Paging Dispatch" [RFC8025] to compress the RPL artifacts as indicated in [RFC8138].

In that case, the source routed header is the exact copy of the (chain of) SRH-6LoRH found in the NSM-VIO, also terminating by the Track Egress. The RPI-6LoRH is appended next, followed by an IP-in-IP 6LoRH Header that indicates the Ingress router in the Encapsulator Address field, see as a similar case Figure 20 of [RFC9035].

To signal the Track in the packet, this specification leverages the RPL Forwarding model follows:

- \* In the data packets, the Track DODAGID and the TrackID MUST be respectively signaled as the IPv6 Source Address and the RPLInstanceID field of the RPI that MUST be placed in the outer chain of IPv6 Headers.

The RPI carries a local RPLInstanceID called the TrackID, which, in association with the DODAGID, indicates the Track along which the packet is forwarded.

The D flag in the RPLInstanceID MUST be set to 0 to indicate that the source address in the IPv6 header is set to the DODAGID, more in Section 6.3.

- \* This draft conforms to the principles of [RFC9008] with regards to packet forwarding and encapsulation along a Track, as follows:
  - With this draft, the Track is a RPL DODAG. From the perspective of that DODAG, the Track Ingress is the Root, the Track Egress is a RPL-Aware 6LR, and neighbors of the Track Egress that can be reached via the Track, but are external to it, are external destinations and treated as RPL-Unaware Leaves (RULs). The encapsulation rules in [RFC9008] apply.
  - If the Track Ingress is the originator of the packet and the Track Egress is the destination of the packet, there is no need for an encapsulation.
  - So the Track Ingress must encapsulate the traffic that it did not originate, and add an RPI.

A packet that is being routed over the RPL Instance associated to a first Non-Storing Mode Track MAY be placed (encapsulated) in a second Track to cover one loose hop of the first Track as discussed in more details Section 3.5.2.3. On the other hand, a Storing Mode Track must be strict and a packet that it placed in a Storing Mode Track MUST follow that Track till the Track Egress.

The forwarding of a packet along a track will fail if the Track continuity is broken, e.g.:

- \* In the case of a strict path along a Segment, if the next strict hop is not reachable, the packet is dropped.
- \* In the case of a loose source-routed path, when the loose next hop is not a neighbor, there must be a Segment of the same Track to that loose next hop. When that is the case the packet is forwarded to the next hop along that segment, or a common neighbor with the loose next hop, on which case the packet is forwarded to that neighbor, or another Track to the loose next hop for which this node or a neighbor is Ingress; in the last case, another encapsulation takes place and the process possibly recurses; otherwise the packet is dropped.

- \* When a Track Egress extracts a packet from a Track (decapsulates the packet), the destination of the inner packet must be either this node or a direct neighbor, or a Target of another Segment of the same Track for which this node is Ingress, otherwise the packet MUST be dropped.

In case of a failure forwarding a packet along a Segment, e.g., the next hop is unreachable, the node that discovers the fault MUST send an ICMPv6 Error message [RFC4443] to the Root, with a new Code "Error in P-Route" (See Section 11.15). The Root can then repair by updating the broken Segment and/or Tracks, and in the case of a broken Segment, remove the leftover sections of the segment using SM-VIOs with a lifetime of 0 indicating the section of one or more nodes being removed (See Section 6.6).

In case of a permanent forwarding error along a Source Route path, the node that fails to forward SHOULD send an ICMP error with a code "Error in Source Routing Header" back to the source of the packet, as described in section 11.2.2.3. of [RPL]. Upon this message, the encapsulating node SHOULD stop using the source route path for a reasonable period of time which duration depends on the deployment, and it SHOULD send an ICMP message with a Code "Error in P-Route" to the Root. Failure to follow these steps may result in packet loss and wasted resources along the source route path that is broken.

Either way, the ICMP message MUST be throttled in case of consecutive occurrences. It MUST be sourced at the ULA or a GUA that is used in this Track for the source node, so the Root can establish where the error happened.

The portion of the invoking packet that is sent back in the ICMP message SHOULD record at least up to the RH if one is present, and this hop of the RH SHOULD be consumed by this node so that the destination in the IPv6 header is the next hop that this node could not reach. If a 6LoWPAN Routing Header (6LoRH) [RFC8138] is used to carry the IPv6 routing information in the outer header then that whole 6LoRH information SHOULD be present in the ICMP message.

#### 6.8. Compression of the RPL Artifacts

When using [RFC8138] in the Main DODAG operated in Non-Storing Mode in a 6LoWPAN LLN, a typical packet that circulates in the Main DODAG is formatted as shown in Figure 20, representing the case where :

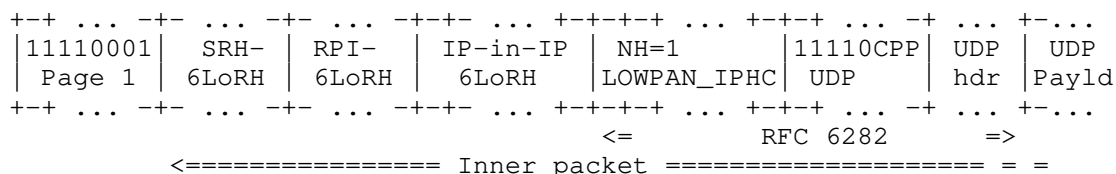


Figure 20: A Packet as Forwarded along the Main DODAG

Since there is no page switch between the encapsulated packet and the encapsulation, the first octet of the compressed packet that acts as page selector is actually removed at encapsulation, so the inner packet used in the descriptions below start with the SRH-6LoRH, and is verbatim the packet represented in Figure 20 from the second octet on.

When encapsulating that inner packet to place it in the Track, the first header that the Ingress appends at the head of the inner packet is an IP-in-IP 6LoRH Header; in that header, the encapsulator address, which maps to the IPv6 source address in the uncompressed form, contains a GUA or ULA IPv6 address of the Ingress node that serves as DODAG ID for the Track, expressed in the compressed form and using the DODAGID of the Main DODAG as compression reference. If the address is compressed to 2 bytes, the resulting value for the Length field shown in Figure 21 is 3, meaning that the SRH-6LoRH as a whole is 5-octets long.

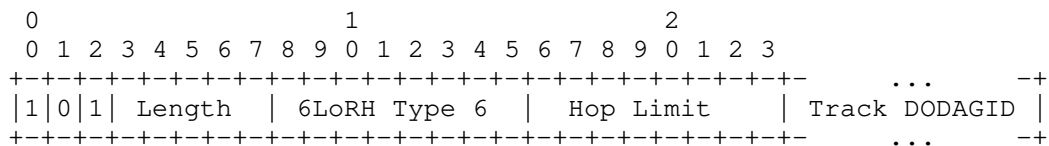


Figure 21: The IP-in-IP 6LoRH Header

At the head of the resulting sequence of bytes, the track Ingress then adds the RPI that carries the TrackID as RPLInstanceID as a P-RPI-6LoRH Header, as illustrated in Figure 12, using the TrackID as RPLInstanceID. Combined with the IP-in-IP 6LoRH Header, this allows to identify the Track without ambiguity.

The SRH-6LoRH is then added at the head of the resulting sequence of bytes as a verbatim copy of the content of the SR-VIO that signaled the selected Track Leg.

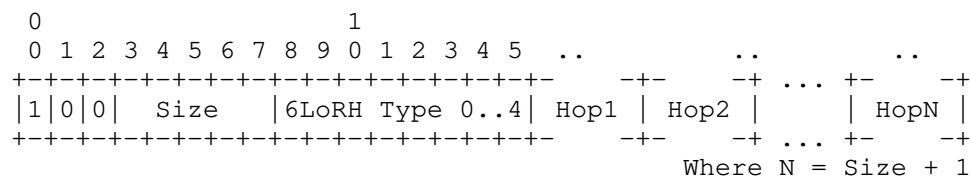
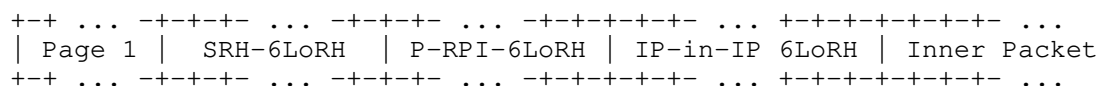


Figure 22: The SRH 6LoRH Header

The format of the resulting encapsulated packet in [RFC8138] compressed form is illustrated in Figure 23:



Signals : Loose Hops : TrackID : Track DODAGID :

Figure 23: A Packet as Forwarded along a Track

## 7. Lesser Constrained Variations

### 7.1. Storing Mode Main DODAG

This specification expects that the Main DODAG is operated in Non-Storing Mode. The reasons for that limitation are mostly related to LLN operations, power and spectrum conservation:

- \* In Non-Storing Mode The Root already possesses the DODAG topology, so the additional topological information is reduced to the siblings.
- \* The downwards routes are updated with unicast messages to the Root, which ensures that the Root can reach back to the LLN nodes after a repair faster than in the case of Storing Mode. Also the Root can control the use of the path diversity in the DODAG to reach to the LLN nodes. For both reasons, Non-Storing Mode provides better capabilities for the Root to maintain the P-Routes.
- \* When the Main DODAG is operated in Non-Storing Mode, P-Routes enable loose Source Routing, which is only an advantage in that mode. Storing Mode does not use Source Routing Headers, and does not derive the same benefits from this capability.

On the other hand, since RPL is a Layer-3 routing protocol, its applicability extends beyond LLNs to a generic IP network. RPL requires fewer resources than alternative IGPs like OSPF, ISIS,



EIGRP, BABEL or RIP at the expense of a route stretch vs. the shortest path routes to a destination that those protocols compute. P-Routes add the capability to install shortest and/or constrained routes to special destinations such as discussed in section A.9.4. of the ANIMA ACP [RFC8994].

In a powered and wired network, when enough memory to store the needed routes is available, the RPL Storing Mode proposes a better trade-off than the Non-Storing, as it reduces the route stretch and lowers the load on the Root. In that case, the control path between the Root and the LLN nodes is highly available compared to LLNs, and the nodes can be reached to maintain the P-Routes at most times.

This section specifies the additions that are needed to support Projected Routes when the Main DODAG is operated in Storing Mode. As long as the RPI can be processed adequately by the dataplane, the changes to this specification are limited to the DAO message. The Track structure, routes and forwarding operations remain the same. Since there is no capability negotiation, the expectation is that all the nodes in the network support this specification in the same fashion, or are configured the same way through management.

In Storing Mode, the Root misses the Child to Parent relationship that forms the Main DODAG, as well as the sibling information. To provide that knowledge the nodes in the network MUST send additional DAO messages that are unicast to the Root as Non-Storing DAO messages are.

In the DAO message, the originating router advertises a set of neighbor nodes using Sibling Information Options (SIO)s, regardless of the relative position in the DODAG of the advertised node vs. this router.

The DAO message MUST be formed as follows:

- \* The originating router is identified by the source address of the DAO. That address MUST be the one that this router registers to neighbor routers so the Root can correlate the DAOs from those routers when they advertise this router as their neighbor. The DAO contains one or more sequences of one Transit Information Option and one or more Sibling Information Options. There is no RPL Target Option so the Root is not confused into adding a Storing Mode route to the Target.

- \* The TIO is formed as in Storing Mode, and the Parent Address is not present. The Path Sequence and Path Lifetime fields are aligned with the values used in the Address Registration of the node(s) advertised in the SIO, as explained in Section 9.1. of [RFC9010]. Having similar values in all nodes allows to factorise the TIO for multiple SIOs as done with [RPL].
- \* The TIO is followed by one or more SIOs that provide an address (ULA or GUA) of the advertised neighbor node.

But the RPL routing information headers may not be supported on all type of routed network infrastructures, especially not in high-speed routers. When the RPI is not supported in the dataplane, there cannot be local RPL Instances and RPL can only operate as a single topology (the Main DODAG). The RPL Instance is that of the Main DODAG and the Ingress node that encapsulates is not the Root. The routes along the Tracks are alternate routes to those available along the Main DODAG. They MAY conflict with routes to children and MUST take precedence in the routing table. The Targets MUST be adjacent to the Track Egress to avoid loops that may form if the packet is reinjected in the Main DODAG.

#### 7.2. A Track as a Full DODAG

This specification builds parallel or crossing Track Legs as opposed to a more complex DODAG with interconnections at any place desirable. The reason for that limitation is related to constrained node operations, and capability to store large amount of topological information and compute complex paths:

- \* With this specification, the node in the LLN has no topological awareness, and does not need to maintain dynamic information about the link quality and availability.
- \* The Root has a complete topological information and statistical metrics that allow it or its PCE to perform a global optimization of all Tracks in its DODAG. Based on that information, the Root computes the Track Leg and predigest the source route paths.
- \* The node merely selects one of the proposed paths and applies the associated pre-computed routing header in the encapsulation. This alleviates both the complexity of computing a path and the compressed form of the routing header.

The RAW Architecture [RAW-ARCHI] actually expects the PSE at the Track Ingress to react to changes in the forwarding conditions along the Track, and reroute packets to maintain the required degree of reliability. To achieve this, the PSE need the full richness of a DODAG to form any path that could make meet the Service Level Objective (SLO).

This section specifies the additions that are needed to turn the Track into a full DODAG and enable the main Root to provide the necessary topological information to the Track Ingress. The expectation is that the metrics that the PSE uses are of an order other than that of the PCE, because of the difference of time scale between routing and forwarding, more in [RAW-ARCHI]. It follows that the PSE will learn the metrics it needs from an alternate source, e.g., OAM frames.

To pass the topological information to the Ingress, the Root uses a P-DAO messages that contains sequences of Target and Transit Information options that collectively represent the Track, expressed in the same fashion as in classical Non-Storing Mode. The difference is that the Root is the source as opposed to the destination, and can report information on many Targets, possibly the full Track, with one P-DAO.

Note that the Path Sequence and Lifetime in the TIO are selected by the Root, and that the Target/Transit information tuples in the P-DAO are not those received by the Root in the DAO messages about the said Targets. The Track may follow sibling routes and does not need to be congruent with the Main DODAG.

## 8. Profiles

THIS RFC provides a set of tools that may or may not be needed by an implementation depending on the type of application it serves. This sections described profiles that can be implemented separately and can be used to discriminate what an implementation can and cannot do. This section describes profiles that enable to implement only a portion of this specification to meet a particular use case.

Profiles 0 to 2 operate in the Main RPL Instance and do not require the support of local RPL Instances or the indication of the RPL Instance in the data plane. Profile 3 and above leverage Local RPL Instances to build arbitrary Tracks Rooted at the Track Ingress and using its namespace for TrackID.

Profiles 0 and 1 are REQUIRED by all implementations that may be used in LLNs; Profiles 1 leverages Storing Mode to reduce the size of the Source Route Header in the most common LLN deployments. Profile 2 is

RECOMMENDED in high speed / wired environment to enable traffic Engineering and network automation. All the other profile / environment combinations are OPTIONAL.

**Profile 0** Profile 0 is the Legacy support of [RPL] Non-Storing Mode, with default routing Northwards (up) and strict source routing Southwards (down the main DODAG). It provides the minimal common functionality that must be implemented as a prerequisite to all the Track-supporting profiles. The other Profiles extend Profile 0 with selected capabilities that this specification introduces on top.

**Profile 1 (Storing Mode P-Route Segments along the Main DODAG)** Profile 1 does not create new paths; compared to Profile 0, it combines Storing and Non-Storing Modes to balance the size of the Routing Header in the packet and the amount of state in the intermediate routers in a Non-Storing Mode RPL DODAG.

**Profile 2 (Non-Storing Mode P-Route Segments along the Main DODAG)** Profile 2 extends Profile 0 with Strict Source-Routing Non-Storing Mode P-Routes along the Main DODAG, which is the same as Profile 1 but using NSM VIOs as opposed to SM VIOs. Profile 2 provides the same capability to compress the SRH in packets down the Main DODAG as Profile 1, but it requires an encapsulation, in order to insert an additional SRH between the loose source routing hops. In that case, the Tracks MUST be installed as subTracks of the Main DODAG, the main RPL Instance MUST be used as TrackID, and the Ingress node that encapsulates is not the Root as it does not own the DODAGID.

**Profile 3** In order to form the best path possible, those Profiles require the support of Sibling Information Option to inform the Root of additional possible hops. Profile 3 extends Profile 1 with additional Storing Mode P-Routes that install segments that do not follow the Main DODAG. If the Segment Ingress (in the SM-VIO) is the same as the IPv6 Address of the Track Ingress (in the projected DAO base Object), the P-DAO creates an implicit Track between the Segment Ingress and the Segment Egress.

**Profile 4** Profile 4 extends Profile 2 with Strict Source-Routing Non-Storing Mode P-Routes to form East-West Tracks that are inside the Main DODAG but do not necessarily follow it. A Track is formed as one or more strict source source routed paths between the Root that is the Track Ingress, and the Track Egress that is the last node.

**Profile 5** Profile 5 Combines Profile 4 with Profile 1 and enables to

loose source routing between the Ingress and the Egress of the Track. As in Profile 1, Storing Mode P-Routes connect the dots in the loose source route.

Profile 6 Profile 6 Combines Profile 4 with Profile 2 and also enables to loose source routing between the Ingress and the Egress of the Track.

Profile 7 Profile 7 implements profile 5 in a Main DODAG that is operated in Storing Mode as presented in Section 7.1. As in Profile 1 and 2, the TrackID is the RPLInstanceID of the Main DODAG. Longest match rules decide whether a packet is sent along the Main DODAG or rerouted in a track.

Profile 8 Profile 8 is offered in preparation of the RAW work, and for use cases where an arbitrary node in the network can afford the same code complexity as the RPL Root in a traditional deployment. It offers a full DODAG visibility to the Track Ingress as specified in Section 7.2 in a Non-Storing Mode Main DODAG.

Profile 9 Profile 9 combines profiles 7 and 8, operating the Track as a full DODAG within a Storing Mode Main DODAG, using only the Main DODAG RPLInstanceID as TrackID.

## 9. Backwards Compatibility

This specification can operate in a mixed network where some nodes support it and some do not. There are restrictions, though. All nodes that need to process a P-DAO MUST support this specification. As discussed in Section 3.7.1, how the root knows whether the nodes capabilities and whether it support this specification is out of scope.

This specification defines the 'D' flag in the RPL DODAG Configuration Option (see Section 4.1.7) to signal that the RPL nodes can request the creation of Tracks. The requester may not know whether the Track can effectively be constructed, and whether enough nodes along the preferred paths support this specification. Therefore it makes sense to only set the 'D' flags in DIO when the conditions of success are in place, in particular when all the nodes that could be on path of tracks are upgraded.

## 10. Security Considerations

It is worth noting that with [RPL], every node in the LLN is RPL-aware and can inject any RPL-based attack in the network. This draft uses messages that are already present in RPL [RPL] with optional secured versions. The same secured versions may be used with this draft, and whatever security is deployed for a given network also applies to the flows in this draft.

The LLN nodes depend on the 6LBR and the RPL participants for their operation. A trust model is necessary to ensure that the right devices are acting in these roles, so as to avoid threats such as black-holing, (see [RFC7416] section 7). This trust model could be at a minimum based on a Layer-2 Secure joining and the Link-Layer security. This is a generic 6LoWPAN requirement, see Req5.1 in Appendix B.5 of [RFC8505].

In a general manner, the Security Considerations in [RPL], and [RFC7416] apply to this specification as well. The Link-Layer security is needed in particular to prevent Denial-Of-Service attacks whereby a rogue router creates a high churn in the RPL network by constantly injected forged P-DAO messages and using up all the available storage in the attacked routers.

With this specification, only the Root may generate P-DAO messages. PDR messages may only be sent to the Root. This specification expects that the communication with the Root is authenticated but does enforce which method is used.

Additionally, the trust model could include a role validation (e.g., using a role-based authorization) to ensure that the node that claims to be a RPL Root is entitled to do so. That trust should propagate from Egress to Ingress in the case of a Storing Mode P-DAO.

This specification suggests some validation of the VIO to prevent basic loops by avoiding that a node appears twice. But that is only a minimal protection. Arguably, an attacker that can inject P-DAOs can reroute any traffic and deplete critical resources such as spectrum and battery in the LLN rapidly.

## 11. IANA Considerations

### 11.1. RPL DODAG Configuration Option Flag

IANA is requested to assign a flag from the "DODAG Configuration Option Flags for MOP 0..6" [RFC9010] registry as follows:

Bit Number	Capability Description	Reference
0 (suggested)	Projected Routes Support (D)	THIS RFC

Table 21: New DODAG Configuration Option Flag

IANA is requested to add [THIS RFC] as a reference for MOP 7 in the RPL Mode of Operation registry.

#### 11.2. Elective 6LoWPAN Routing Header Type

THIS RFC updates the IANA registry titled "Elective 6LoWPAN Routing Header Type" that was created for [RFC8138] and assigns the following value:

Value	Description	Reference
8 (Suggested)	P-RPI-6LoRH	THIS RFC

Table 22: New Elective 6LoWPAN Routing Header Type

#### 11.3. Critical 6LoWPAN Routing Header Type

THIS RFC updates the IANA registry titled "Critical 6LoWPAN Routing Header Type" that was created for [RFC8138] and assigns the following value:

Value	Description	Reference
8 (Suggested)	P-RPI-6LoRH	THIS RFC

Table 23: New Critical 6LoWPAN Routing Header Type

#### 11.4. Subregistry For The RPL Option Flags

IANA is required to create a subregistry for the 8-bit RPL Option Flags field, as detailed in Figure 11, under the "Routing Protocol for Low Power and Lossy Networks (RPL)" registry. The bits are indexed from 0 (leftmost) to 7. Each bit is Tracked with the following qualities:

- \* Bit number (counting from bit 0 as the most significant bit)
- \* Indication When Set
- \* Reference

Registration procedure is "Standards Action" [RFC8126]. The initial allocation is as indicated in Table 27:

Bit number	Indication When Set	Reference
0	Down 'O'	[RFC6553]
1	Rank-Error (R)	[RFC6553]
2	Forwarding-Error (F)	[RFC6553]
3 (Suggested)	Projected-Route (P)	THIS RFC

Table 24: Initial PDR Flags

#### 11.5. RPL Control Codes

THIS RFC extends the IANA Subregistry created by RFC 6550 for RPL Control Codes as indicated in Table 25:

Code	Description	Reference
0x09 (Suggested)	Projected DAO Request (PDR)	THIS RFC
0x0A (Suggested)	PDR-ACK	THIS RFC

Table 25: New RPL Control Codes

#### 11.6. RPL Control Message Options

THIS RFC extends the IANA Subregistry created by RFC 6550 for RPL Control Message Options as indicated in Table 26:



Value	Meaning	Reference
0x0E (Suggested)	Stateful VIO (SM-VIO)	THIS RFC
0x0F (Suggested)	Source-Routed VIO (NSM-VIO)	THIS RFC
0x10 (Suggested)	Sibling Information option	THIS RFC

Table 26: RPL Control Message Options

### 11.7. SubRegistry for the Projected DAO Request Flags

IANA is required to create a registry for the 8-bit Projected DAO Request (PDR) Flags field. Each bit is Tracked with the following qualities:

- \* Bit number (counting from bit 0 as the most significant bit)
- \* Capability description
- \* Reference

Registration procedure is "Standards Action" [RFC8126]. The initial allocation is as indicated in Table 27:

Bit number	Capability description	Reference
0	PDR-ACK request (K)	THIS RFC
1	Requested path should be redundant (R)	THIS RFC

Table 27: Initial PDR Flags

### 11.8. SubRegistry for the PDR-ACK Flags

IANA is required to create an subregistry for the 8-bit PDR-ACK Flags field. Each bit is Tracked with the following qualities:

- \* Bit number (counting from bit 0 as the most significant bit)
- \* Capability description
- \* Reference

Registration procedure is "Standards Action" [RFC8126]. No bit is currently defined for the PDR-ACK Flags.

#### 11.9. Subregistry for the PDR-ACK Acceptance Status Values

IANA is requested to create a Subregistry for the PDR-ACK Acceptance Status values.

- \* Possible values are 6-bit unsigned integers (0..63).
- \* Registration procedure is "Standards Action" [RFC8126].
- \* Initial allocation is as indicated in Table 28:

Value	Meaning	Reference
0	Unqualified Acceptance	THIS RFC

Table 28: Acceptance values of the PDR-ACK Status

#### 11.10. Subregistry for the PDR-ACK Rejection Status Values

IANA is requested to create a Subregistry for the PDR-ACK Rejection Status values.

- \* Possible values are 6-bit unsigned integers (0..63).
- \* Registration procedure is "Standards Action" [RFC8126].
- \* Initial allocation is as indicated in Table 29:

Value	Meaning	Reference
0	Unqualified Rejection	THIS RFC
1	Transient Failure	THIS RFC

Table 29: Rejection values of the PDR-ACK Status

## 11.11. SubRegistry for the Via Information Options Flags

IANA is requested to create a Subregistry for the 5-bit Via Information Options (Via Information Option) Flags field. Each bit is Tracked with the following qualities:

- \* Bit number (counting from bit 0 as the most significant bit)
- \* Capability description
- \* Reference

Registration procedure is "Standards Action" [RFC8126]. No bit is currently defined for the Via Information Options (Via Information Option) Flags.

## 11.12. SubRegistry for the Sibling Information Option Flags

IANA is required to create a registry for the 5-bit Sibling Information Option (SIO) Flags field. Each bit is Tracked with the following qualities:

- \* Bit number (counting from bit 0 as the most significant bit)
- \* Capability description
- \* Reference

Registration procedure is "Standards Action" [RFC8126]. The initial allocation is as indicated in Table 30:

Bit number	Capability description	Reference
0 (Suggested)	"S" flag: Sibling in same DODAG as Self	THIS RFC

Table 30: Initial SIO Flags

## 11.13. Destination Advertisement Object Flag

THIS RFC modifies the "Destination Advertisement Object (DAO) Flags" registry initially created in Section 20.11 of [RPL] .

Section 4.1.1 also defines one new entry in the Registry as follows:

Bit Number	Capability Description	Reference
2 (Suggested)	Projected DAO (P)	THIS RFC

Table 31: New Destination Advertisement Object  
(DAO) Flag

#### 11.14. Destination Advertisement Object Acknowledgment Flag

THIS RFC modifies the "Destination Advertisement Object (DAO) Acknowledgment Flags" registry initially created in Section 20.12 of [RPL] .

Section 4.1.2 also defines one new entry in the Registry as follows:

Bit Number	Capability Description	Reference
1 (Suggested)	Projected DAO-ACK (P)	THIS RFC

Table 32: New Destination Advertisement Object  
Acknowledgment Flag

#### 11.15. New ICMPv6 Error Code

In some cases RPL will return an ICMPv6 error message when a message cannot be forwarded along a P-Route.

IANA has defined an ICMPv6 "Code" Fields Registry for ICMPv6 Message Types. ICMPv6 Message Type 1 describes "destination Unreachable" codes. This specification requires that a new code is allocated from the ICMPv6 Code Fields Registry for ICMPv6 Message Type 1, for "Error in P-Route", with a suggested code value of 8, to be confirmed by IANA.

#### 11.16. RPL Rejection Status values

This specification updates the Subregistry for the "RPL Rejection Status" values under the RPL registry, as follows:

Value	Meaning	Reference
2 (Suggested)	Out of Resources	THIS RFC
3 (Suggested)	Error in VIO	THIS RFC
4 (Suggested)	Predecessor Unreachable	THIS RFC
5 (Suggested)	Unreachable Target	THIS RFC
6..63	Unassigned	

Table 33: Rejection values of the RPL Status

## 12. Acknowledgments

The authors wish to acknowledge JP Vasseur, Remy Liubing, James Pylakutty, and Patrick Wetterwald for their contributions to the ideas developed here. Many thanks to Dominique Barthel and SVR Anand for their global contribution to 6TiSCH, RAW and this RFC, as well as text suggestions that were incorporated. Also special thanks Li Zhao and Toerless Eckert for their in-depth reviews, with many excellent suggestions that improved the readability and well as the content of the specification. Many thanks to Remous-Aris Koutsiamanis for his review during WGLC.

## 13. Normative References

### [INT-ARCHI]

Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <<https://www.rfc-editor.org/info/rfc1122>>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <<https://www.rfc-editor.org/info/rfc4443>>.

- [RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <<https://www.rfc-editor.org/info/rfc4655>>.
- [RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <<https://www.rfc-editor.org/info/rfc6282>>.
- [RPL] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <<https://www.rfc-editor.org/info/rfc6550>>.
- [RFC6553] Hui, J. and JP. Vasseur, "The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams", RFC 6553, DOI 10.17487/RFC6553, March 2012, <<https://www.rfc-editor.org/info/rfc6553>>.
- [RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, DOI 10.17487/RFC6554, March 2012, <<https://www.rfc-editor.org/info/rfc6554>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8138] Thubert, P., Ed., Bormann, C., Toutain, L., and R. Cragie, "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing Header", RFC 8138, DOI 10.17487/RFC8138, April 2017, <<https://www.rfc-editor.org/info/rfc8138>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <<https://www.rfc-editor.org/info/rfc8754>>.

- [RFC9008] Robles, M.I., Richardson, M., and P. Thubert, "Using RPI Option Type, Routing Header for Source Routes, and IPv6-in-IPv6 Encapsulation in the RPL Data Plane", RFC 9008, DOI 10.17487/RFC9008, April 2021, <<https://www.rfc-editor.org/info/rfc9008>>.

#### 14. Informative References

- [6LoWPAN] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <<https://www.rfc-editor.org/info/rfc4944>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<https://www.rfc-editor.org/info/rfc5440>>.
- [RFC6997] Goyal, M., Ed., Baccelli, E., Philipp, M., Brandt, A., and J. Martocci, "Reactive Discovery of Point-to-Point Routes in Low-Power and Lossy Networks", RFC 6997, DOI 10.17487/RFC6997, August 2013, <<https://www.rfc-editor.org/info/rfc6997>>.
- [RFC7102] Vasseur, JP., "Terms Used in Routing for Low-Power and Lossy Networks", RFC 7102, DOI 10.17487/RFC7102, January 2014, <<https://www.rfc-editor.org/info/rfc7102>>.
- [RFC7416] Tsao, T., Alexander, R., Dohler, M., Daza, V., Lozano, A., and M. Richardson, Ed., "A Security Threat Analysis for the Routing Protocol for Low-Power and Lossy Networks (RPLs)", RFC 7416, DOI 10.17487/RFC7416, January 2015, <<https://www.rfc-editor.org/info/rfc7416>>.
- [RFC8025] Thubert, P., Ed. and R. Cragie, "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Paging Dispatch", RFC 8025, DOI 10.17487/RFC8025, November 2016, <<https://www.rfc-editor.org/info/rfc8025>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.

- [RFC8505] Thubert, P., Ed., Nordmark, E., Chakrabarti, S., and C. Perkins, "Registration Extensions for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Neighbor Discovery", RFC 8505, DOI 10.17487/RFC8505, November 2018, <<https://www.rfc-editor.org/info/rfc8505>>.
- [RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <<https://www.rfc-editor.org/info/rfc8655>>.
- [RFC8930] Watteyne, T., Ed., Thubert, P., Ed., and C. Bormann, "On Forwarding 6LoWPAN Fragments over a Multi-Hop IPv6 Network", RFC 8930, DOI 10.17487/RFC8930, November 2020, <<https://www.rfc-editor.org/info/rfc8930>>.
- [RFC8931] Thubert, P., Ed., "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Selective Fragment Recovery", RFC 8931, DOI 10.17487/RFC8931, November 2020, <<https://www.rfc-editor.org/info/rfc8931>>.
- [RFC8994] Eckert, T., Ed., Behringer, M., Ed., and S. Bjarnason, "An Autonomic Control Plane (ACP)", RFC 8994, DOI 10.17487/RFC8994, May 2021, <<https://www.rfc-editor.org/info/rfc8994>>.
- [RFC9010] Thubert, P., Ed. and M. Richardson, "Routing for RPL (Routing Protocol for Low-Power and Lossy Networks) Leaves", RFC 9010, DOI 10.17487/RFC9010, April 2021, <<https://www.rfc-editor.org/info/rfc9010>>.
- [RFC9030] Thubert, P., Ed., "An Architecture for IPv6 over the Time-Slotted Channel Hopping Mode of IEEE 802.15.4 (6TiSCH)", RFC 9030, DOI 10.17487/RFC9030, May 2021, <<https://www.rfc-editor.org/info/rfc9030>>.
- [RFC9035] Thubert, P., Ed. and L. Zhao, "A Routing Protocol for Low-Power and Lossy Networks (RPL) Destination-Oriented Directed Acyclic Graph (DODAG) Configuration Option for the 6LoWPAN Routing Header", RFC 9035, DOI 10.17487/RFC9035, April 2021, <<https://www.rfc-editor.org/info/rfc9035>>.



## [RAW-ARCHI]

Thubert, P. and G. Z. Papadopoulos, "Reliable and Available Wireless Architecture", Work in Progress, Internet-Draft, draft-ietf-raw-architecture-04, 4 March 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-raw-architecture-04>>.

## [USE-CASES]

Bernardos, C. J., Papadopoulos, G. Z., Thubert, P., and F. Theoleyre, "RAW use-cases", Work in Progress, Internet-Draft, draft-ietf-raw-use-cases-05, 23 February 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-raw-use-cases-05>>.

## [I-D.kuehlewind-update-tag]

Kuehlewind, M. and S. Krishnan, "Definition of new tags for relations between RFCs", Work in Progress, Internet-Draft, draft-kuehlewind-update-tag-04, 12 July 2021, <<https://datatracker.ietf.org/doc/html/draft-kuehlewind-update-tag-04>>.

## [I-D.irtf-panrg-path-properties]

Enghardt, T. and C. Krähenbühl, "A Vocabulary of Path Properties", Work in Progress, Internet-Draft, draft-irtf-panrg-path-properties-05, 7 March 2022, <<https://datatracker.ietf.org/doc/html/draft-irtf-panrg-path-properties-05>>.

## [PCE]

IETF, "Path Computation Element", <<https://dataTracker.ietf.org/doc/charter-ietf-pce/>>.

## Authors' Addresses

Pascal Thubert (editor)  
Cisco Systems, Inc  
Building D  
45 Allee des Ormes - BP1200  
06254 Mougins - Sophia Antipolis  
France  
Phone: +33 497 23 26 34  
Email: [pthubert@cisco.com](mailto:pthubert@cisco.com)

Rahul Arvind Jadhav  
Huawei Tech  
Kundalahalli Village, Whitefield,  
Bangalore 560037  
Karnataka  
India  
Phone: +91-080-49160700  
Email: rahul.ietf@gmail.com

Michael C. Richardson  
Sandelman Software Works  
Email: mcr+ietf@sandelman.ca  
URI: <http://www.sandelman.ca/>

ROLL  
Internet-Draft  
Intended status: Standards Track  
Expires: September 30, 2018

R. Jadhav, Ed.  
Huawei  
P. Thubert  
Cisco  
R. Sahoo  
Z. Cao  
Huawei  
March 29, 2018

Efficient Route Invalidation  
draft-ietf-roll-efficient-npdao-03

Abstract

This document describes the problems associated with the use of No-Path DAO messaging in RPL and signaling changes to improve route invalidation efficiency.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 30, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
1.1. Requirements Language and Terminology . . . . .	3
1.2. Current No-Path DAO messaging . . . . .	4
1.3. Cases when No-Path DAO may be used . . . . .	4
1.4. Why No-Path DAO is important? . . . . .	5
2. Problems with current No-Path DAO messaging . . . . .	5
2.1. Lost NPDAO due to link break to the previous parent . . . . .	5
2.2. Invalidate routes to dependent nodes of the switching node . . . . .	6
2.3. Route downtime caused by asynchronous operation of NPDAO and DAO . . . . .	6
3. Requirements for the No-Path DAO Optimization . . . . .	6
3.1. Req#1: Tolerant to link failures to the previous parents . . . . .	6
3.2. Req#2: Dependent nodes route invalidation on parent switching . . . . .	6
3.3. Req#3: No impact on traffic while NPDAO operation in progress . . . . .	7
4. Proposed changes to RPL signaling . . . . .	7
4.1. Change in RPL route invalidation semantics . . . . .	7
4.2. DAO message format changes . . . . .	7
4.3. Destination Cleanup Object (DCO) . . . . .	8
4.3.1. Secure DCO . . . . .	10
4.3.2. DCO Options . . . . .	10
4.3.3. Path Sequence number in the DCO . . . . .	10
4.3.4. Destination Cleanup Option Acknowledgement (DCO-ACK) . . . . .	10
4.3.5. Secure DCO-ACK . . . . .	11
4.4. Other considerations . . . . .	11
4.4.1. Dependent Nodes invalidation . . . . .	11
4.4.2. NPDAO and DCO in the same network . . . . .	12
5. Acknowledgements . . . . .	12
6. IANA Considerations . . . . .	12
7. Security Considerations . . . . .	12
8. References . . . . .	12
8.1. Normative References . . . . .	13
8.2. Informative References . . . . .	13
Appendix A. Example DCO Messaging . . . . .	13
Authors' Addresses . . . . .	14

## 1. Introduction

RPL [RFC6550] specifies a proactive distance-vector based routing scheme. The specification has an optional messaging in the form of DAO messages using which the 6LBR can learn route towards any of the nodes. In storing mode, DAO messages would result in routing entries been created on all intermediate hops from the node's parent all the way towards the 6LBR.

RPL allows use of No-Path DAO (NPDAO) messaging to invalidate a routing path corresponding to the given target, thus releasing resources utilized on that path. A No-Path DAO is a DAO message with route lifetime of zero, originates at the target node and always flows upstream towards the 6LBR, signaling route invalidation for the given target. This document explains the problems associated with the current use of NPDAO messaging and also discusses the requirements for an optimized No-Path DAO messaging scheme. Further a new pro-active route invalidation message called as "Destination Cleanup Object (DCO)" is specified which fulfills all mentioned requirements of an optimized route invalidation messaging.

6TiSCH architecture [I-D.ietf-6tisch-architecture] leverages RPL and specifies use of non-storing and storing MOP for its routing operation. Thus an improvement in route invalidation will help optimize 6TiSCH based networks.

### 1.1. Requirements Language and Terminology

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

The document only caters to the RPL's storing mode of operation (MOP). The non-storing MOP does not require use of NPDAO for route invalidation since routing entries are not maintained on 6LRs.

Common Ancestor node: 6LR node which is the first common node on the old and new path for the child node.

NPDAO: No-Path DAO. A DAO message which has target with lifetime 0.

DCO: Destination Cleanup Object, A new RPL control message type defined by this draft.

Regular DAO: A DAO message with non-zero lifetime.

This document also uses terminology described in [RFC6550].

### 1.2. Current No-Path DAO messaging

RPL introduced No-Path DAO messaging in the storing mode so that the node switching its current parent can inform its parents and ancestors to invalidate the existing route. Subsequently parents or ancestors would release any resources (such as the routing entry) it maintains on behalf of target node. The NPDAO message always traverses the RPL tree in upward direction, originating at the target node itself.

For the rest of this document consider the following topology:

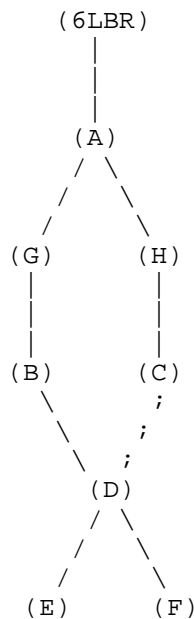


Figure 1: Sample topology

Node (D) is connected via preferred parent (B). (D) has an alternate path via (C) towards the BR. Node (A) is the common ancestor for (D) for paths through (B)-(G) and (C)-(H). When (D) switches from (B) to (C), [RFC6550] suggests sending No-Path DAO to (B) and regular DAO to (C).

### 1.3. Cases when No-Path DAO may be used

There are following cases in which a node switches its parent and may employ No-Path DAO messaging:

Case I: Current parent becomes unavailable because of transient or permanent link or parent node failure.

Case II: The node finds a better parent node i.e. the metrics of another parent is better than its current parent.

Case III: The node switches to a new parent whom it "thinks" has a better metric but does not in reality.

The usual steps of operation when the node switches the parent is that the node sends a No-Path DAO message via its current parent to invalidate its current route and subsequently it tries to establish a new routing path by sending a new DAO via its new parent.

#### 1.4. Why No-Path DAO is important?

Nodes in LLNs may be resource constrained. There is limited memory available and routing entry records are the one of the primary elements occupying dynamic memory in the nodes. Route invalidation helps 6LR nodes to decide which entries could be discarded to better achieve resource utilization in case of contention. Thus it becomes necessary to have efficient route invalidation mechanism. Also note that a single parent switch may result in a "sub-tree" switching from one parent to another. Thus the route invalidation needs to be done on behalf of the sub-tree and not the switching node alone. In the above example, when Node (D) switches parent, the route invalidation needs to be done for (D), (E) and (F). Thus without efficient route invalidation, a 6LR may have to hold a lot of unwanted route entries.

## 2. Problems with current No-Path DAO messaging

### 2.1. Lost NPDAO due to link break to the previous parent

When a node switches its parent, the NPDAO is to be sent via its previous parent and a regular DAO via its new parent. In cases where the node switches its parent because of transient or permanent parent link/node failure then the NPDAO message is bound to fail. RPL assumes communication link with the previous parent for No-Path DAO messaging.

RPL allows use of route lifetime to remove unwanted routes in case the routes could not be refreshed. But route lifetimes in case of LLNs could be substantially high and thus the route entries would be stuck for longer times.

## 2.2. Invalidate routes to dependent nodes of the switching node

No-path DAO is sent by the node who has switched the parent but it does not work for the dependent child nodes below it. The specification does not specify how route invalidation will work for sub-children, resulting in stale routing entries on behalf of the sub-children on the previous route. The only way for 6LR to invalidate the route entries for dependent nodes would be to use route lifetime expiry which could be substantially high for LLNs.

In the example topology, when Node (D) switches its parent, Node (D) generates an NPDAO on its behalf. Post switching, Node (D) transmits a DIO with incremented DTSN so that child nodes, node (E) and (F), generate DAOs to trigger route update on the new path for themselves. There is no NPDAO generated by these child nodes through the previous path resulting in stale entries on nodes (B) and (G) for nodes (E) and (F).

## 2.3. Route downtime caused by asynchronous operation of NPDAO and DAO

A switching node may generate both an NPDAO and DAO via two different paths at almost the same time. There is a possibility that an NPDAO generated may invalidate the previous route and the regular DAO sent via the new path gets lost on the way. This may result in route downtime thus impacting downward traffic for the switching node. In the example topology, consider Node (D) switches from parent (B) to (C) because the metrics of the path via (C) are better. Note that the previous path via (B) may still be available (albeit at relatively bad metrics). An NPDAO sent from previous route may invalidate the existing route whereas there is no way to determine whether the new DAO has successfully updated the route entries on the new path.

## 3. Requirements for the No-Path DAO Optimization

### 3.1. Req#1: Tolerant to link failures to the previous parents

When the switching node send the NPDAO message to the previous parent, it is normal that the link to the previous parent is prone to failure. Therefore, it is required that the NPDAO message MUST be tolerant to the link failure during the switching.

### 3.2. Req#2: Dependent nodes route invalidation on parent switching

While switching the parent node and sending NPDAO message, it is required that the routing entries to the dependent nodes of the switching node will be updated accordingly on the previous parents and other relevant upstream nodes.



### 3.3. Req#3: No impact on traffic while NPDAO operation in progress

While sending the NPDAO and DAO messages, it is possible that the NPDAO successfully invalidates the previous path, while the newly sent DAO gets lost (new path not set up successfully). This will result into downstream unreachability to the current switching node. Therefore, it is desirable that the NPDAO is synchronized with the DAO to avoid the risk of route downtime.

## 4. Proposed changes to RPL signaling

### 4.1. Change in RPL route invalidation semantics

As described in Section 1.2, the NPDAO originates at the node switching the parent and traverses upstream towards the root. In order to solve the problems as mentioned in Section 2, the draft adds new pro-active route invalidation message called as "Destination Cleanup Object" (DCO) that originates at a common ancestor node between the new and old path. The trigger for the common ancestor node to generate this DCO is the change in the next hop for the target on reception of an update message in the form of regular DAO for the target.

In the Figure 1, when node D decides to switch the path from B to C, it sends a regular DAO to node C with reachability information containing target as address of D and a incremented path sequence number. Node C will update the routing table based on the reachability information in DAO and in turn generate another DAO with the same reachability information and forward it to H. Node H also follows the same procedure as Node C and forwards it to node A. When node A receives the regular DAO, it finds that it already has a routing table entry on behalf of the target address of node D. It finds however that the next hop information for reaching node D has changed i.e. the node D has decided to change the paths. In this case, Node A which is the common ancestor node for node D along the two paths (previous and new), may generate a DCO which traverses downwards in the network. The document in the subsequent section will explain the message format changes to handle this downward flow of NPDAO.

### 4.2. DAO message format changes

Every RPL message is divided into base message fields and additional Options. The base fields apply to the message as a whole and options are appended to add message/use-case specific attributes. As an example, a DAO message may be attributed by one or more "RPL Target" options which specifies the reachability information for the given

targets. Similarly, a Transit Information option may be associated with a set of RPL Target options.

The draft proposes a change in DAO message to contain "Invalidate previous route" (I) bit. This I-bit which is carried in regular DAO message, signals the common ancestor node to generate a DCO on behalf of the target node. The I-bit is carried in the transit container option which augments the reachability information for a given set of RPL Target(s).

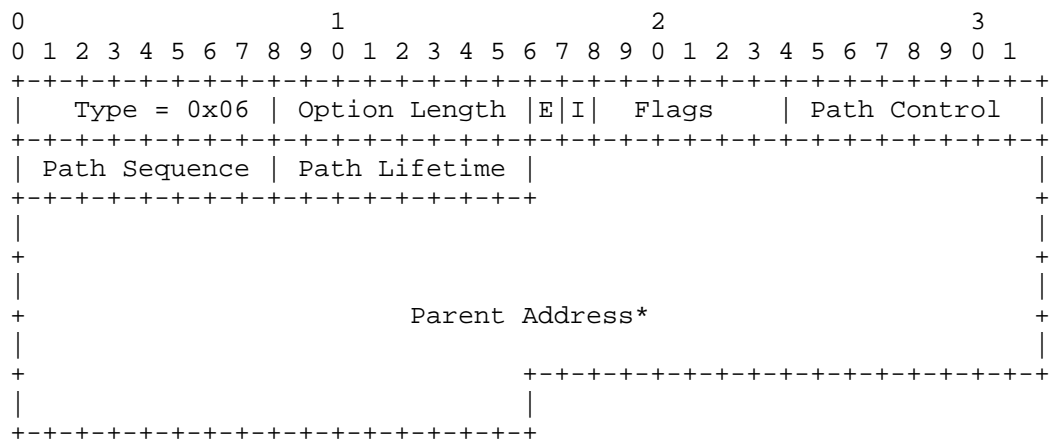


Figure 2: Updated Transit Information Option (New I flag added)

I (Invalidate previous route) bit: 1 bit flag. The 'I' flag is set by the target node to indicate that it wishes to invalidate the previous route by a common ancestor node between the two paths.

The common ancestor node SHOULD generate a DCO message in response to this I-bit when it sees that the routing adjacencies have changed for the target. I-bit governs the ownership of the DCO message in a way that the target node is still in control of its own route invalidation.

#### 4.3. Destination Cleanup Object (DCO)

A new ICMPv6 RPL control message type is defined by this specification called as "Destination Cleanup Object" (DCO), which is used for proactive cleanup of state and routing information held on behalf of the target node by 6LRs. The DCO message always traverses downstream and cleans up route information and other state information associated with the given target.

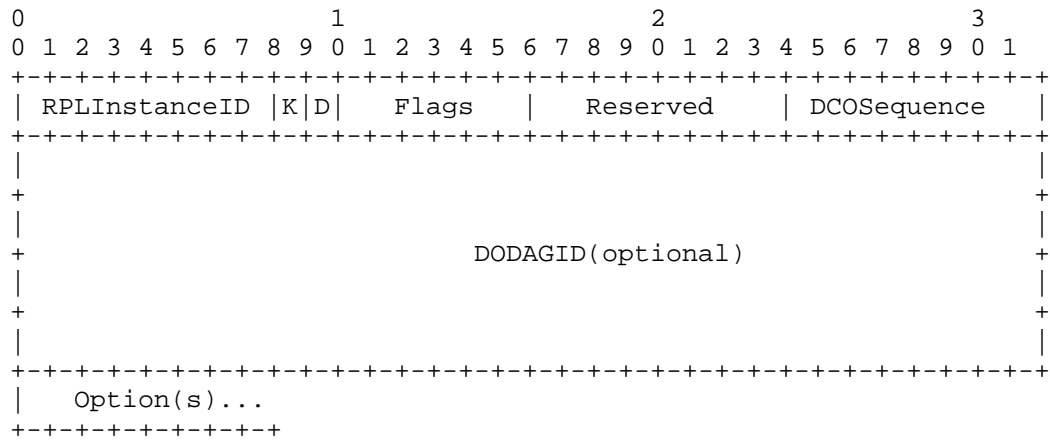


Figure 3: DCO base object

RPLInstanceID: 8-bit field indicating the topology instance associated with the DODAG, as learned from the DIO.

K: The 'K' flag indicates that the recipient is expected to send a DCO-ACK back.

D: The 'D' flag indicates that the DODAGID field is present. This flag MUST be set when a local RPLInstanceID is used.

Flags: The 6 bits remaining unused in the Flags field are reserved for flags. The field MUST be initialized to zero by the sender and MUST be ignored by the receiver.

Reserved: 8-bit unused field. The field MUST be initialized to zero by the sender and MUST be ignored by the receiver.

DCOSequence: Incremented at each unique DCO message from a node and echoed in the DCO-ACK message.

DODAGID (optional): 128-bit unsigned integer set by a DODAG root that uniquely identifies a DODAG. This field is only present when the 'D' flag is set. This field is typically only present when a local RPLInstanceID is in use, in order to identify the DODAGID that is associated with the RPLInstanceID. When a global RPLInstanceID is in use, this field need not be present. Unassigned bits of the DCO Base are reserved. They MUST be set to zero on transmission and MUST be ignored on reception.

#### 4.3.1. Secure DCO

A Secure DCO message follows the format in [RFC6550] figure 7, where the base message format is the DCO message shown in Figure 3.

#### 4.3.2. DCO Options

The DCO message MAY carry valid options. This specification allows for the DCO message to carry the following options:

```
0x00 Pad1
0x01 PadN
0x05 RPL Target
0x06 Transit Information
0x09 RPL Target Descriptor
```

The DCO carries a Target option and an associated Transit Information option with a lifetime of 0x00000000 to indicate a loss of reachability to that Target.

#### 4.3.3. Path Sequence number in the DCO

A DCO message may contain a Path Sequence in the transit information option to identify the freshness of the DCO message. The Path Sequence in the DCO MUST use the same Path Sequence number present in the regular DAO message when the DCO is generated in response to DAO message.

#### 4.3.4. Destination Cleanup Option Acknowledgement (DCO-ACK)

The DCO-ACK message may be sent as a unicast packet by a DCO recipient in response to a unicast DCO message.

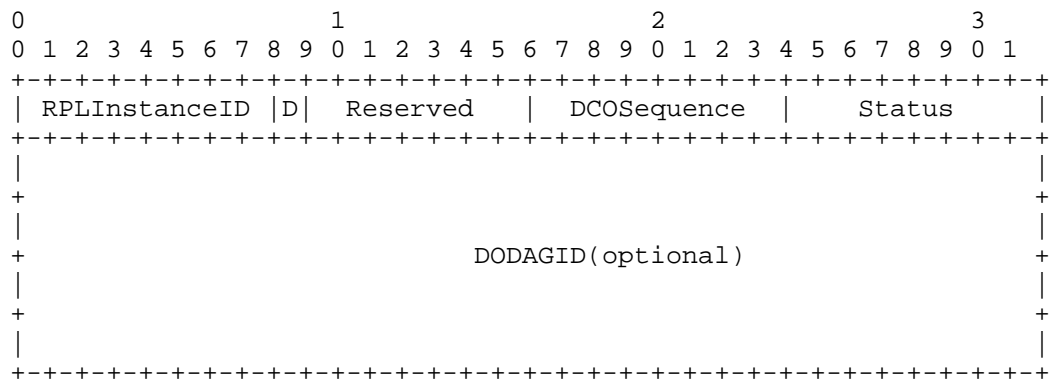


Figure 4: DCO-ACK base object

RPLInstanceID: 8-bit field indicating the topology instance associated with the DODAG, as learned from the DIO.

D: The 'D' flag indicates that the DODAGID field is present. This flag MUST be set when a local RPLInstanceID is used.

Reserved: 7-bit unused field. The field MUST be initialized to zero by the sender and MUST be ignored by the receiver.

DCOSequence: Incremented at each unique DCO message from a node and echoed in the DCO-ACK message.

Status: Indicates the completion. Status 0 is defined as unqualified acceptance in this specification. The remaining status values are reserved as rejection codes.

DODAGID (optional): 128-bit unsigned integer set by a DODAG root that uniquely identifies a DODAG. This field is only present when the 'D' flag is set. This field is typically only present when a local RPLInstanceID is in use, in order to identify the DODAGID that is associated with the RPLInstanceID. When a global RPLInstanceID is in use, this field need not be present. Unassigned bits of the DCO-Ack Base are reserved. They MUST be set to zero on transmission and MUST be ignored on reception.

#### 4.3.5. Secure DCO-ACK

A Secure DCO-ACK message follows the format in [RFC6550] figure 7, where the base message format is the DCO-ACK message shown in Figure 4.

#### 4.4. Other considerations

##### 4.4.1. Dependent Nodes invalidation

Current RPL [RFC6550] does not provide a mechanism for route invalidation for dependent nodes. This document allows the dependent nodes invalidation. Dependent nodes will generate their respective DAOs to update their paths, and the previous route invalidation for those nodes should work in the similar manner described for switching node. The dependent node may set the I-bit in the transit information container as part of regular DAO so as to request invalidation of previous route from the common ancestor node.

#### 4.4.2. NPDAO and DCO in the same network

Even with the changed semantics, the current NPDAO mechanism in [RFC6550] can still be used. There are certain scenarios where current NPDAO signalling may still be used, for example, when the route lifetime expiry of the target happens or when the node simply decides to gracefully terminate the RPL session on graceful node shutdown. Moreover a deployment can have a mix of nodes supporting the proposed DCO and the existing NPDAO mechanism.

### 5. Acknowledgements

We would like to thank Cenk Gundogan and Simon Duquennoy for their review and comments.

### 6. IANA Considerations

IANA is requested to allocate new ICMPv6 RPL control codes in RPL [RFC6550] for DCO and DCO-ACK messages.

Code	Description	Reference
0x04	Destination Cleanup Object	This document
0x05	Destination Cleanup Object Acknowledgement	This document
0x84	Secure Destination Cleanup Object	This document
0x85	Secure Destination Cleanup Object Acknowledgement	This document

IANA is requested to allocate bit 18 in the Transit Information Option defined in RPL [RFC6550] section 6.7.8 for Invalidate route 'I' flag.

### 7. Security Considerations

This document handles security considerations inline to base RPL. Secure versions of DCO and DCO-ACK are added similar to other RPL messages. For general RPL security considerations, see [RFC6550].

### 8. References

## 8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <<https://www.rfc-editor.org/info/rfc6550>>.

## 8.2. Informative References

- [I-D.ietf-6tisch-architecture] Thubert, P., "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4", draft-ietf-6tisch-architecture-13 (work in progress), November 2017.

## Appendix A. Example DCO Messaging

In Figure 1, node (D) switches its parent from (B) to (C). The sequence of actions is as follows:

1. Node D switches its parent from node B to node C
2. D sends a regular DAO(*tgt=D,pathseq=x+1,I\_flag=1*) in the updated path to C
3. C checks for routing entry on behalf of D, since it cannot find an entry on behalf of D it creates a new routing entry and forwards the reachability information of the target D to H in a DAO.
4. Similar to C, node H checks for routing entry on behalf of D, cannot find an entry and hence creates a new routing entry and forwards the reachability information of the target D to H in a DAO.
5. Node A receives the DAO, and checks for routing entry on behalf of D. It finds a routing entry but checks that the next hop for target D is now changed. Node A checks the *I\_flag* and generates DCO(*tgt=D,pathseq=pathseq(DAO)*) to previous next hop for target D which is G. Subsequently, A updates the routing entry and forwards the reachability information of target D upstream DAO(*tgt=D,pathseq=x+1,I\_flag=x*) (the *I\_flag* carries no significance henceforth).
6. Node G receives the DCO and invalidates routing entry of target D and forwards the (un)reachability information downstream to B.

7. Similarly, B processes the DCO by invalidating the routing entry of target D and forwards the (un)reachability information downstream to D.
8. D ignores the DCO since the target is itself.
9. The propagation of the DCO will stop at any node where the node does not have an routing information associated with the target. If the routing information is present and the pathseq associated is not older, then still the DCO is dropped.

#### Authors' Addresses

Rahul Arvind Jadhav (editor)  
Huawei  
Kundalahalli Village, Whitefield,  
Bangalore, Karnataka 560037  
India

Phone: +91-080-49160700  
Email: rahul.ietf@gmail.com

Pascal Thubert  
Cisco Systems, Inc  
Building D  
45 Allee des Ormes - BP1200  
MOUGINS - Sophia Antipolis 06254  
France

Phone: +33 497 23 26 34  
Email: pthubert@cisco.com

Rabi Narayan Sahoo  
Huawei  
Kundalahalli Village, Whitefield,  
Bangalore, Karnataka 560037  
India

Phone: +91-080-49160700  
Email: rabinarayans@huawei.com



Zhen Cao  
Huawei  
W Chang'an Ave  
Beijing 560037  
China

Email: [zhencao.ietf@gmail.com](mailto:zhencao.ietf@gmail.com)

ROLL  
Internet-Draft  
Intended status: Standards Track  
Expires: October 17, 2020

R. Jadhav, Ed.  
Huawei  
P. Thubert  
Cisco  
R. Sahoo  
Z. Cao  
Huawei  
April 15, 2020

Efficient Route Invalidation  
draft-ietf-roll-efficient-npdao-18

Abstract

This document explains the problems associated with the current use of NPDAO messaging and also discusses the requirements for an optimized route invalidation messaging scheme. Further a new proactive route invalidation message called as "Destination Cleanup Object" (DCO) is specified which fulfills requirements of an optimized route invalidation messaging.

Status of This Memo

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

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

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

This Internet-Draft will expire on October 17, 2020.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
1.1. Requirements Language and Terminology . . . . .	3
1.2. Current NPDAO messaging . . . . .	4
1.3. Why Is NPDAO Important? . . . . .	5
2. Problems with current NPDAO messaging . . . . .	6
2.1. Lost NPDAO due to link break to the previous parent . . . . .	6
2.2. Invalidate Routes of Dependent Nodes . . . . .	6
2.3. Possible route downtime caused by asynchronous operation of NPDAO and DAO . . . . .	6
3. Requirements for the NPDAO Optimization . . . . .	6
3.1. Req#1: Remove messaging dependency on link to the previous parent . . . . .	6
3.2. Req#2: Dependent nodes route invalidation on parent switching . . . . .	7
3.3. Req#3: Route invalidation should not impact data traffic . . . . .	7
4. Changes to RPL signaling . . . . .	7
4.1. Change in RPL route invalidation semantics . . . . .	7
4.2. Transit Information Option changes . . . . .	8
4.3. Destination Cleanup Object (DCO) . . . . .	9
4.3.1. Secure DCO . . . . .	10
4.3.2. DCO Options . . . . .	10
4.3.3. Path Sequence number in the DCO . . . . .	11
4.3.4. Destination Cleanup Option Acknowledgment (DCO-ACK) . . . . .	11
4.3.5. Secure DCO-ACK . . . . .	12
4.4. DCO Base Rules . . . . .	12
4.5. Unsolicited DCO . . . . .	13
4.6. Other considerations . . . . .	13
4.6.1. Dependent Nodes invalidation . . . . .	13
4.6.2. NPDAO and DCO in the same network . . . . .	14
4.6.3. Considerations for DCO retry . . . . .	14
4.6.4. DCO with multiple preferred parents . . . . .	15
5. Acknowledgments . . . . .	16
6. IANA Considerations . . . . .	16
6.1. New Registry for the Destination Cleanup Object (DCO) Flags . . . . .	16
6.2. New Registry for the Destination Cleanup Object Acknowledgment (DCO-ACK) Status field . . . . .	17
6.3. New Registry for the Destination Cleanup Object (DCO) Acknowledgment Flags . . . . .	17
7. Security Considerations . . . . .	18

8. Normative References . . . . .	19
Appendix A. Example Messaging . . . . .	20
A.1. Example DCO Messaging . . . . .	20
A.2. Example DCO Messaging with multiple preferred parents . .	21
Authors' Addresses . . . . .	22

## 1. Introduction

RPL [RFC6550] (Routing Protocol for Low power and lossy networks) specifies a proactive distance-vector based routing scheme. RPL has optional messaging in the form of DAO (Destination Advertisement Object) messages, which the 6LBR (6Lo Border Router) and 6LR (6Lo Router) can use to learn a route towards the downstream nodes. In storing mode, DAO messages would result in routing entries being created on all intermediate 6LRs from the node's parent all the way towards the 6LBR.

RPL allows the use of No-Path DAO (NPDAO) messaging to invalidate a routing path corresponding to the given target, thus releasing resources utilized on that path. A NPDAO is a DAO message with route lifetime of zero, originates at the target node and always flows upstream towards the 6LBR. This document explains the problems associated with the current use of NPDAO messaging and also discusses the requirements for an optimized route invalidation messaging scheme. Further a new proactive route invalidation message called as "Destination Cleanup Object" (DCO) is specified which fulfills requirements of an optimized route invalidation messaging.

The document only caters to the RPL's storing mode of operation (MOP). The non-storing MOP does not require use of NPDAO for route invalidation since routing entries are not maintained on 6LRs.

### 1.1. Requirements Language and 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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This specification requires readers to be familiar with all the terms and concepts that are discussed in "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks" [RFC6550].

Low Power and Lossy Networks (LLN):

Network in which both the routers and their interconnect are constrained. LLN routers typically operate with constraints on processing power, memory, and energy (battery power). Their

interconnects are characterized by high loss rates, low data rates, and instability.

**6LoWPAN Router (6LR):**

An intermediate router that is able to send and receive Router Advertisements (RAs) and Router Solicitations (RSs) as well as forward and route IPv6 packets.

**Directed Acyclic Graph (DAG):**

A directed graph having the property that all edges are oriented in such a way that no cycles exist.

**Destination-Oriented DAG (DODAG):**

A DAG rooted at a single destination, i.e., at a single DAG root with no outgoing edges.

**6LoWPAN Border Router (6LBR):**

A border router which is a DODAG root and is the edge node for traffic flowing in and out of the 6LoWPAN network.

**Destination Advertisement Object (DAO):**

DAO messaging allows downstream routes to the nodes to be established.

**DODAG Information Object (DIO):**

DIO messaging allows upstream routes to the 6LBR to be established. DIO messaging is initiated at the DAO root.

**Common Ancestor node**

6LR/6LBR node which is the first common node between two paths of a target node.

**No-Path DAO (NPDAO):**

A DAO message which has target with lifetime 0 used for the purpose of route invalidation.

**Destination Cleanup Object (DCO):**

A new RPL control message code defined by this document. DCO messaging improves proactive route invalidation in RPL.

**Regular DAO:**

A DAO message with non-zero lifetime. Routing adjacencies are created or updated based on this message.

**Target node:**

The node switching its parent whose routing adjacencies are updated (created/removed).

## 1.2. Current NPDAO messaging

RPL uses NPDAO messaging in the storing mode so that the node changing its routing adjacencies can invalidate the previous route. This is needed so that nodes along the previous path can release any resources (such as the routing entry) they maintain on behalf of target node.

For the rest of this document consider the following topology:

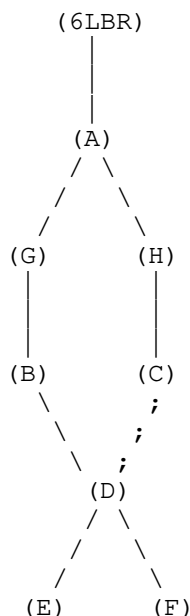


Figure 1: Sample topology

Node (D) is connected via preferred parent (B). (D) has an alternate path via (C) towards the 6LBR. Node (A) is the common ancestor for (D) for paths through (B)-(G) and (C)-(H). When (D) switches from (B) to (C), RPL allows sending NPDAO to (B) and regular DAO to (C).

### 1.3. Why Is NPDAO Important?

Nodes in LLNs may be resource constrained. There is limited memory available and routing entry records are one of the primary elements occupying dynamic memory in the nodes. Route invalidation helps 6LR nodes to decide which entries could be discarded to better optimize resource utilization. Thus it becomes necessary to have an efficient route invalidation mechanism. Also note that a single parent switch may result in a "sub-tree" switching from one parent to another. Thus the route invalidation needs to be done on behalf of the sub-tree and not the switching node alone. In the above example, when Node (D) switches parent, the route updates needs to be done for the routing tables entries of (C), (H), (A), (G), and (B) with destination (D), (E) and (F). Without efficient route invalidation, a 6LR may have to hold a lot of stale route entries.

## 2. Problems with current NPDAO messaging

### 2.1. Lost NPDAO due to link break to the previous parent

When a node switches its parent, the NPDAO is to be sent to its previous parent and a regular DAO to its new parent. In cases where the node switches its parent because of transient or permanent parent link/node failure then the NPDAO message is bound to fail.

### 2.2. Invalidate Routes of Dependent Nodes

RPL does not specify how route invalidation will work for dependent nodes rooted at the switching node, resulting in stale routing entries of the dependent nodes. The only way for 6LR to invalidate the route entries for dependent nodes would be to use route lifetime expiry which could be substantially high for LLNs.

In the example topology, when Node (D) switches its parent, Node (D) generates an NPDAO on its behalf. There is no NPDAO generated by the dependent child nodes (E) and (F), through the previous path via (D) to (B) and (G), resulting in stale entries on nodes (B) and (G) for nodes (E) and (F).

### 2.3. Possible route downtime caused by asynchronous operation of NPDAO and DAO

A switching node may generate both an NPDAO and DAO via two different paths at almost the same time. There is a possibility that an NPDAO generated may invalidate the previous route and the regular DAO sent via the new path gets lost on the way. This may result in route downtime impacting downward traffic for the switching node.

In the example topology, consider Node (D) switches from parent (B) to (C). An NPDAO sent via the previous route may invalidate the previous route whereas there is no way to determine whether the new DAO has successfully updated the route entries on the new path.

## 3. Requirements for the NPDAO Optimization

### 3.1. Req#1: Remove messaging dependency on link to the previous parent

When the switching node sends the NPDAO message to the previous parent, it is normal that the link to the previous parent is prone to failure (that's why the node decided to switch). Therefore, it is required that the route invalidation does not depend on the previous link which is prone to failure. The previous link referred here represents the link between the node and its previous parent (from whom the node is now disassociating).

### 3.2. Req#2: Dependent nodes route invalidation on parent switching

It should be possible to do route invalidation for dependent nodes rooted at the switching node.

### 3.3. Req#3: Route invalidation should not impact data traffic

While sending the NPDAO and DAO messages, it is possible that the NPDAO successfully invalidates the previous path, while the newly sent DAO gets lost (new path not set up successfully). This will result in downstream unreachability to the node switching paths. Therefore, it is desirable that the route invalidation is synchronized with the DAO to avoid the risk of route downtime.

## 4. Changes to RPL signaling

### 4.1. Change in RPL route invalidation semantics

As described in Section 1.2, the NPDAO originates at the node changing to a new parent and traverses upstream towards the root. In order to solve the problems as mentioned in Section 2, the document adds a new proactive route invalidation message called "Destination Cleanup Object" (DCO) that originates at a common ancestor node and flows downstream between the new and old path. The common ancestor node generates a DCO in response to the change in the next-hop on receiving a regular DAO with updated Path Sequence for the target.

The 6LRs in the path for DCO take action such as route invalidation based on the DCO information and subsequently send another DCO with the same information downstream to the next hop. This operation is similar to how the DAOs are handled on intermediate 6LRs in storing MOP in [RFC6550]. Just like DAO in storing MOP, the DCO is sent using link-local unicast source and destination IPv6 address. Unlike DAO, which always travels upstream, the DCO always travels downstream.

In Figure 1, when node D decides to switch the path from B to C, it sends a regular DAO to node C with reachability information containing the address of D as the target and an incremented Path Sequence. Node C will update the routing table based on the reachability information in the DAO and in turn generate another DAO with the same reachability information and forward it to H. Node H also follows the same procedure as Node C and forwards it to node A. When node A receives the regular DAO, it finds that it already has a routing table entry on behalf of the target address of node D. It finds however that the next hop information for reaching node D has changed i.e., node D has decided to change the paths. In this case, Node A which is the common ancestor node for node D along the two



paths (previous and new), should generate a DCO which traverses downwards in the network. Node A handles normal DAO forwarding to 6LBR as required by [RFC6550].

#### 4.2. Transit Information Option changes

Every RPL message is divided into base message fields and additional Options as described in Section 6 of [RFC6550]. The base fields apply to the message as a whole and options are appended to add message/use-case specific attributes. As an example, a DAO message may be attributed by one or more "RPL Target" options which specify the reachability information for the given targets. Similarly, a Transit Information option may be associated with a set of RPL Target options.

This document specifies a change in the Transit Information Option to contain the "Invalidate previous route" (I) flag. This 'I' flag signals the common ancestor node to generate a DCO on behalf of the target node with a RPL Status of 195 indicating that the address has moved. The 'I' flag is carried in the Transit Information Option which augments the reachability information for a given set of RPL Target(s). Transit Information Option with 'I' flag set should be carried in the DAO message when route invalidation is sought for the corresponding target(s).

Value 195 represents 'E' and 'A' bit in RPL Status to be set as per Figure 3 of [I-D.ietf-roll-unaware-leaves] with the lower 6 bits with value 3 indicating 'Moved' as per Table 1 of [RFC8505].

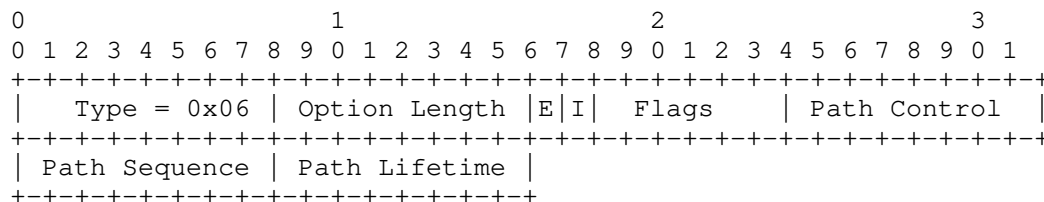


Figure 2: Updated Transit Information Option (New I flag added)

I (Invalidate previous route) flag: The 'I' flag is set by the target node to indicate to the common ancestor node that it wishes to invalidate any previous route between the two paths.

[RFC6550] allows the parent address to be sent in the Transit Information Option depending on the mode of operation. In case of storing mode of operation the field is usually not needed. In case of DCO, the parent address field MUST NOT be included.



D: The 'D' flag indicates that the DODAGID field is present. This flag MUST be set when a local RPLInstanceID is used.

Flags: The 6 bits remaining unused in the Flags field are reserved for future use. These bits MUST be initialized to zero by the sender and MUST be ignored by the receiver.

RPL Status: As defined in [RFC6550] and updated in [I-D.ietf-roll-unaware-leaves]. The root or common parent that generates a DCO is authoritative for setting the status information and the information is unchanged as propagated down the DODAG. This document does not specify a differentiated action based on the RPL status.

DCOSequence: 8-bit field incremented at each unique DCO message from a node and echoed in the DCO-ACK message. The initial DCOSequence can be chosen randomly by the node. Section 4.4 explains the handling of the DCOSequence.

DODAGID (optional): 128-bit unsigned integer set by a DODAG root that uniquely identifies a DODAG. This field MUST be present when the 'D' flag is set and MUST NOT be present if 'D' flag is not set. DODAGID is used when a local RPLInstanceID is in use, in order to identify the DODAGID that is associated with the RPLInstanceID.

#### 4.3.1. Secure DCO

A Secure DCO message follows the format in [RFC6550] Figure 7, where the base message format is the DCO message shown in Figure 3.

#### 4.3.2. DCO Options

The DCO message MUST carry at least one RPL Target and the Transit Information Option and MAY carry other valid options. This specification allows for the DCO message to carry the following options:

- 0x00 Pad1
- 0x01 PadN
- 0x05 RPL Target
- 0x06 Transit Information
- 0x09 RPL Target Descriptor

Section 6.7 of [RFC6550] defines all the above mentioned options. The DCO carries an RPL Target Option and an associated Transit Information Option with a lifetime of 0x00000000 to indicate a loss of reachability to that Target.

#### 4.3.3. Path Sequence number in the DCO

A DCO message may contain a Path Sequence in the Transit Information Option to identify the freshness of the DCO message. The Path Sequence in the DCO MUST use the same Path Sequence number present in the regular DAO message when the DCO is generated in response to a DAO message. Thus if a DCO is received by a 6LR and subsequently a DAO is received with an old sequence number, then the DAO MUST be ignored. When the DCO is generated in response to a DCO from upstream parent, the Path Sequence MUST be copied from the received DCO.

#### 4.3.4. Destination Cleanup Option Acknowledgment (DCO-ACK)

The DCO-ACK message SHOULD be sent as a unicast packet by a DCO recipient in response to a unicast DCO message with 'K' flag set. If 'K' flag is not set then the receiver of the DCO message MAY send a DCO-ACK, especially to report an error condition.

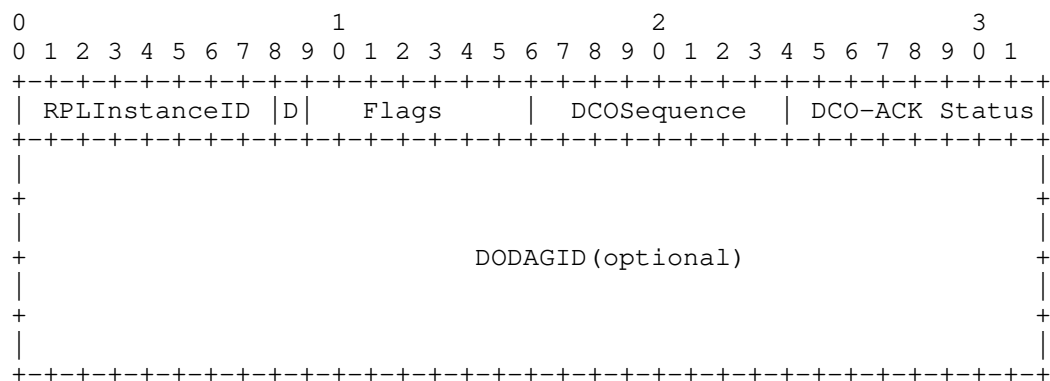


Figure 4: DCO-ACK base object

**RPLInstanceID:** 8-bit field indicating the topology instance associated with the DODAG, as learned from the DIO.

**D:** The 'D' flag indicates that the DODAGID field is present. This flag MUST be set when a local RPLInstanceID is used.

**Flags:** 7-bit unused field. The field MUST be initialized to zero by the sender and MUST be ignored by the receiver.

**DCOSequence:** 8-bit field. The DCOSequence in DCO-ACK is copied from the DCOSequence received in the DCO message.

DCO-ACK Status: Indicates the completion. A value of 0 is defined as unqualified acceptance in this specification. A value of 1 is defined as "No routing-entry for the Target found". The remaining status values are reserved as rejection codes.

DODAGID (optional): 128-bit unsigned integer set by a DODAG root that uniquely identifies a DODAG. This field MUST be present when the 'D' flag is set and MUST NOT be present when 'D' flag is not set. DODAGID is used when a local RPLInstanceID is in use, in order to identify the DODAGID that is associated with the RPLInstanceID.

#### 4.3.5. Secure DCO-ACK

A Secure DCO-ACK message follows the format in [RFC6550] Figure 7, where the base message format is the DCO-ACK message shown in Figure 4.

#### 4.4. DCO Base Rules

1. If a node sends a DCO message with newer or different information than the prior DCO message transmission, it MUST increment the DCOSequence field by at least one. A DCO message transmission that is identical to the prior DCO message transmission MAY increment the DCOSequence field. The DCOSequence counter follows the sequence counter operation as defined in Section 7.2 of [RFC6550].
2. The RPLInstanceID and DODAGID fields of a DCO message MUST be the same value as that of the DAO message in response to which the DCO is generated on the common ancestor node.
3. A node MAY set the 'K' flag in a unicast DCO message to solicit a unicast DCO-ACK in response in order to confirm the attempt.
4. A node receiving a unicast DCO message with the 'K' flag set SHOULD respond with a DCO-ACK. A node receiving a DCO message without the 'K' flag set MAY respond with a DCO-ACK, especially to report an error condition.
5. A node receiving a unicast DCO message MUST verify the stored Path Sequence in context to the given target. If the stored Path Sequence is more fresh, newer than the Path Sequence received in the DCO, then the DCO MUST be dropped.
6. A node that sets the 'K' flag in a unicast DCO message but does not receive DCO-ACK in response MAY reschedule the DCO message transmission for another attempt, up until an implementation specific number of retries.
7. A node receiving a unicast DCO message with its own address in the RPL Target Option MUST strip-off that Target Option. If this Target Option is the only one in the DCO message then the DCO message MUST be dropped.

The scope of DCOSequence values is unique to the node which generates it.

#### 4.5. Unsolicited DCO

A 6LR may generate an unsolicited DCO to unilaterally cleanup the path on behalf of the target entry. The 6LR has all the state information, namely, the Target address and the Path Sequence, required for generating DCO in its routing table. The conditions why 6LR may generate an unsolicited DCO are beyond the scope of this document but some possible reasons could be:

1. On route expiry of an entry, a 6LR may decide to graciously cleanup the entry by initiating DCO.
2. 6LR needs to entertain higher priority entries in case the routing table is full, thus resulting in eviction of an existing routing entry. In this case the eviction can be handled graciously using DCO.

Note that if the 6LR initiates a unilateral path cleanup using DCO and if it has the latest state for the target then the DCO would finally reach the target node. Thus the target node would be informed of its invalidation.

#### 4.6. Other considerations

##### 4.6.1. Dependent Nodes invalidation

Current RPL [RFC6550] does not provide a mechanism for route invalidation for dependent nodes. This document allows the dependent nodes invalidation. Dependent nodes will generate their respective DAOs to update their paths, and the previous route invalidation for those nodes should work in the similar manner described for switching node. The dependent node may set the 'I' flag in the Transit Information Option as part of regular DAO so as to request invalidation of previous route from the common ancestor node.

Dependent nodes do not have any indication regarding if any of their parents in turn have decided to switch their parent. Thus for route invalidation the dependent nodes may choose to always set the 'I' flag in all its DAO message's Transit Information Option. Note that setting the 'I' flag is not counterproductive even if there is no previous route to be invalidated.

#### 4.6.2. NPDAO and DCO in the same network

The current NPDAO mechanism in [RFC6550] can still be used in the same network where DCO is used. The NPDAO messaging can be used, for example, on route lifetime expiry of the target or when the node simply decides to gracefully terminate the RPL session on graceful node shutdown. Moreover, a deployment can have a mix of nodes supporting the DCO and the existing NPDAO mechanism. It is also possible that the same node supports both the NPDAO and DCO signaling for route invalidation.

Section 9.8 of [RFC6550] states, "When a node removes a node from its DAO parent set, it SHOULD send a No-Path DAO message to that removed DAO parent to invalidate the existing router". This document introduces an alternative and more optimized way of route invalidation but it also allows existing NPDAO messaging to work. Thus an implementation has two choices to make when a route invalidation is to be initiated:

1. Use NPDAO to invalidate the previous route and send regular DAO on the new path.
2. Send regular DAO on the new path with the 'I' flag set in the Transit Information Option such that the common ancestor node initiates the DCO message downstream to invalidate the previous route.

This document recommends using option 2 for reasons specified in Section 3 in this document.

This document assumes that all the 6LRs in the network support this specification. If there are 6LRs en-route DCO message path which do not support this document, then the route invalidation for corresponding targets may not work or may work partially i.e., only part of the path supporting DCO may be invalidated. Alternatively, a node could generate an NPDAO if it does not receive a DCO with itself as target within specified time limit. The specified time limit is deployment specific and depends upon the maximum depth of the network and per hop average latency. Note that sending NPDAO and DCO for the same operation would not result in unwanted side-effects because the acceptability of NPDAO or DCO depends upon the Path Sequence freshness.

#### 4.6.3. Considerations for DCO retry

A DCO message could be retried by a sender if it sets the 'K' flag and does not receive a DCO-ACK. The DCO retry time could be dependent on the maximum depth of the network and average per hop latency. This could range from 2 seconds to 120 seconds depending on

the deployment. In case the latency limits are not known, an implementation MUST NOT retry more than once in 3 seconds and MUST NOT retry more than 3 times.

The number of retries could also be set depending on how critical the route invalidation could be for the deployment and the link layer retry configuration. For networks supporting only MP2P and P2MP flows, such as in AMI and telemetry applications, the 6LRs may not be very keen to invalidate routes, unless they are highly memory-constrained. For home and building automation networks which may have substantial P2P traffic, the 6LRs might be keen to invalidate efficiently because it may additionally impact the forwarding efficiency.

#### 4.6.4. DCO with multiple preferred parents

[RFC6550] allows a node to select multiple preferred parents for route establishment. Section 9.2.1 of [RFC6550] specifies, "All DAOs generated at the same time for the same Target MUST be sent with the same Path Sequence in the Transit Information". Subsequently when route invalidation has to be initiated, RPL mentions use of NPDAO which can be initiated with an updated Path Sequence to all the parent nodes through which the route is to be invalidated.

With DCO, the Target node itself does not initiate the route invalidation and it is left to the common ancestor node. A common ancestor node when it discovers an updated DAO from a new next-hop, it initiates a DCO. With multiple preferred parents, this handling does not change. But in this case it is recommended that an implementation initiates a DCO after a time period (DelayDCO) such that the common ancestor node may receive updated DAOs from all possible next-hops. This will help to reduce DCO control overhead i.e., the common ancestor can wait for updated DAOs from all possible directions before initiating a DCO for route invalidation. After timeout, the DCO needs to be generated for all the next-hops for whom the route invalidation needs to be done.

This document recommends using a DelayDCO timer value of 1sec. This value is inspired by the default DelayDAO value of 1sec in [RFC6550]. Here the hypothesis is that the DAOs from all possible parent sets would be received on the common ancestor within this time period.

It is still possible that a DCO is generated before all the updated DAOs from all the paths are received. In this case, the ancestor node would start the invalidation procedure for paths from which the updated DAO is not received. The DCO generated in this case would start invalidating the segments along these paths on which the updated DAOs are not received. But once the DAO reaches these



segments, the routing state would be updated along these segments and should not lead to any inconsistent routing state.

Note that there is no requirement for synchronization between DCO and DAOs. The DelayDCO timer simply ensures that the DCO control overhead can be reduced and is only needed when the network contains nodes using multiple preferred parent.

## 5. Acknowledgments

Many thanks to Alvaro Retana, Cenk Gundogan, Simon Duquennoy, Georgios Papadopoulos, Peter Van Der Stok for their review and comments. Alvaro Retana helped shape this document's final version with critical review comments.

## 6. IANA Considerations

IANA is requested to allocate new codes for the DCO and DCO-ACK messages from the RPL Control Codes registry.

Code	Description	Reference
TBD1	Destination Cleanup Object	This document
TBD2	Destination Cleanup Object Acknowledgment	This document
TBD3	Secure Destination Cleanup Object	This document
TBD4	Secure Destination Cleanup Object Acknowledgment	This document

IANA is requested to allocate bit 1 from the Transit Information Option Flags registry for the 'I' flag (Section 4.2)

### 6.1. New Registry for the Destination Cleanup Object (DCO) Flags

IANA is requested to create a registry for the 8-bit Destination Cleanup Object (DCO) Flags field. This registry should be located in existing category of "Routing Protocol for Low Power and Lossy Networks (RPL)".

New bit numbers may be allocated only by an IETF Review. Each bit is tracked with the following qualities:

- o Bit number (counting from bit 0 as the most significant bit)
- o Capability description

- o Defining RFC

The following bits are currently defined:

Bit number	Description	Reference
0	DCO-ACK request (K)	This document
1	DODAGID field is present (D)	This document

#### DCO Base Flags

### 6.2. New Registry for the Destination Cleanup Object Acknowledgment (DCO-ACK) Status field

IANA is requested to create a registry for the 8-bit Destination Cleanup Object Acknowledgment (DCO-ACK) Status field. This registry should be located in existing category of "Routing Protocol for Low Power and Lossy Networks (RPL)".

New Status values may be allocated only by an IETF Review. Each value is tracked with the following qualities:

- o Status Code
- o Description
- o Defining RFC

The following values are currently defined:

Status Code	Description	Reference
0	Unqualified acceptance	This document
1	No routing-entry for the indicated Target found	This document

#### DCO-ACK Status Codes

### 6.3. New Registry for the Destination Cleanup Object (DCO) Acknowledgment Flags

IANA is requested to create a registry for the 8-bit Destination Cleanup Object (DCO) Acknowledgment Flags field. This registry

should be located in existing category of "Routing Protocol for Low Power and Lossy Networks (RPL)".

New bit numbers may be allocated only by an IETF Review. Each bit is tracked with the following qualities:

- o Bit number (counting from bit 0 as the most significant bit)
- o Capability description
- o Defining RFC

The following bits are currently defined:

Bit number	Description	Reference
0	DODAGID field is present (D)	This document

#### DCO-ACK Base Flags

## 7. Security Considerations

This document introduces the ability for a common ancestor node to invalidate a route on behalf of the target node. The common ancestor node could be directed to do so by the target node using the 'I' flag in DCO's Transit Information Option. However, the common ancestor node is in a position to unilaterally initiate the route invalidation since it possesses all the required state information, namely, the Target address and the corresponding Path Sequence. Thus a rogue common ancestor node could initiate such an invalidation and impact the traffic to the target node.

The DCO carries a RPL Status value, which is informative. New Status values may be created over time and a node will ignore an unknown Status value. This enables RPL Status field to be used as a cover channel. But the channel only works once since the message destroys its own medium, that is the existing route that it is removing.

This document also introduces an 'I' flag which is set by the target node and used by the ancestor node to initiate a DCO if the ancestor sees an update in the route adjacency. However, this flag could be spoofed by a malicious 6LR in the path and can cause invalidation of an existing active path. Note that invalidation will happen only if the other conditions such as Path Sequence condition is also met. Having said that, such a malicious 6LR may spoof a DAO on behalf of the (sub) child with the 'I' flag set and can cause route invalidation on behalf of the (sub) child node. Note that, using existing mechanisms offered by [RFC6550], a malicious 6LR might also

spoof a DAO with lifetime of zero or otherwise cause denial of service by dropping traffic entirely, so the new mechanism described in this document does not present a substantially increased risk of disruption.

This document assumes that the security mechanisms as defined in [RFC6550] are followed, which means that the common ancestor node and all the 6LRs are part of the RPL network because they have the required credentials. A non-secure RPL network needs to take into consideration the risks highlighted in this section as well as those highlighted in [RFC6550].

All RPL messages support a secure version of messages which allows integrity protection using either a MAC or a signature. Optionally, secured RPL messages also have encryption protection for confidentiality.

The document adds new messages (DCO, DCO-ACK) which are syntactically similar to existing RPL messages such as DAO, DAO-ACK. Secure versions of DCO and DCO-ACK are added similar to other RPL messages (such as DAO, DAO-ACK).

RPL supports three security modes as mentioned in Section 10.1 of [RFC6550]:

1. Unsecured: In this mode, it is expected that the RPL control messages are secured by other security mechanisms, such as link-layer security. In this mode, the RPL control messages, including DCO, DCO-ACK, do not have Security sections. Also note that unsecured mode does not imply that all messages are sent without any protection.
2. Preinstalled: In this mode, RPL uses secure messages. Thus secure versions of DCO, DCO-ACK MUST be used in this mode.
3. Authenticated: In this mode, RPL uses secure messages. Thus secure versions of DCO, DCO-ACK MUST be used in this mode.

## 8. Normative References

[I-D.ietf-roll-unaware-leaves]

Thubert, P. and M. Richardson, "Routing for RPL Leaves", draft-ietf-roll-unaware-leaves-14 (work in progress), April 2020.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

- [RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <<https://www.rfc-editor.org/info/rfc6550>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

## Appendix A. Example Messaging

### A.1. Example DCO Messaging

In Figure 1, node (D) switches its parent from (B) to (C). This example assumes that Node D has already established its own route via Node B-G-A-6LBR using pathseq=x. The example uses DAO and DCO messaging convention and specifies only the required parameters to explain the example namely, the parameter 'tgt', which stands for Target Option and value of this parameter specifies the address of the target node. The parameter 'pathseq', which specifies the Path Sequence value carried in the Transit Information Option. The parameter 'I\_flag' specifies the 'I' flag in the Transit Information Option. sequence of actions is as follows:

1. Node D switches its parent from node B to node C
2. D sends a regular DAO(tgt=D,pathseq=x+1,I\_flag=1) in the updated path to C
3. C checks for a routing entry on behalf of D, since it cannot find an entry on behalf of D it creates a new routing entry and forwards the reachability information of the target D to H in a DAO(tgt=D,pathseq=x+1,I\_flag=1).
4. Similar to C, node H checks for a routing entry on behalf of D, cannot find an entry and hence creates a new routing entry and forwards the reachability information of the target D to A in a DAO(tgt=D,pathseq=x+1,I\_flag=1).
5. Node A receives the DAO(tgt=D,pathseq=x+1,I\_flag=1), and checks for a routing entry on behalf of D. It finds a routing entry but checks that the next hop for target D is different (i.e., Node G). Node A checks the I\_flag and generates DCO(tgt=D,pathseq=x+1) to previous next hop for target D which is G. Subsequently, Node A updates the routing entry and forwards the reachability information of target D upstream DAO(tgt=D,pathseq=x+1,I\_flag=1).
6. Node G receives the DCO(tgt=D,pathseq=x+1). It checks if the received path sequence is later than the stored path sequence. If it is later, Node G invalidates the routing entry of target D

and forwards the (un)reachability information downstream to B in DCO(tgt=D,pathseq=x+1).

7. Similarly, B processes the DCO(tgt=D,pathseq=x+1) by invalidating the routing entry of target D and forwards the (un)reachability information downstream to D.
8. D ignores the DCO(tgt=D,pathseq=x+1) since the target is itself.
9. The propagation of the DCO will stop at any node where the node does not have an routing information associated with the target. If cached routing information is present and the cached Path Sequence is higher than the value in the DCO, then the DCO is dropped.

#### A.2. Example DCO Messaging with multiple preferred parents

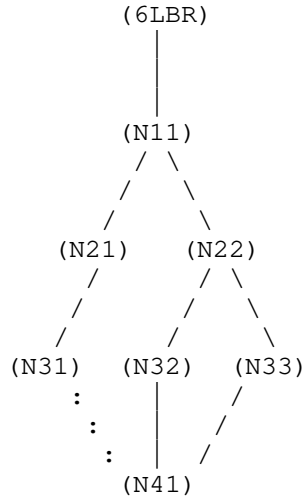


Figure 5: Sample topology 2

In Figure 5, node (N41) selects multiple preferred parents (N32) and (N33). The sequence of actions is as follows:

1. (N41) sends DAO(tgt=N41,PS=x,I\_flag=1) to (N32) and (N33). Here I\_flag refers to the Invalidation flag and PS refers to Path Sequence in Transit Information option.
2. (N32) sends DAO(tgt=N41,PS=x,I\_flag=1) to (N22). (N33) also sends DAO(tgt=N41,PS=x,I\_flag=1) to (N22). (N22) learns multiple routes for the same destination (N41) through multiple next-hops. (N22) may receive the DAOs from (N32) and (N33) in any order with the I\_flag set. The implementation should use the DelayDCO timer to wait to initiate the DCO. If (N22) receives an updated DAO from all the paths then the DCO need not

- be initiated in this case. Thus the route table at N22 should contain (Dst,NextHop,PS): { (N41,N32,x), (N41,N33,x) }.
3. (N22) sends DAO(tgt=N41,PS=x,I\_flag=1) to (N11).
  4. (N11) sends DAO(tgt=N41,PS=x,I\_flag=1) to (6LBR). Thus the complete path is established.
  5. (N41) decides to change preferred parent set from { N32, N33 } to { N31, N32 }.
  6. (N41) sends DAO(tgt=N41,PS=x+1,I\_flag=1) to (N32). (N41) sends DAO(tgt=N41,PS=x+1,I\_flag=1) to (N31).
  7. (N32) sends DAO(tgt=N41,PS=x+1,I\_flag=1) to (N22). (N22) has multiple routes to destination (N41). It sees that a new Path Sequence for Target=N41 is received and thus it waits for pre-determined time period (DelayDCO time period) to invalidate another route {(N41),(N33),x}. After time period, (N22) sends DCO(tgt=N41,PS=x+1) to (N33). Also (N22) sends the regular DAO(tgt=N41,PS=x+1,I\_flag=1) to (N11).
  8. (N33) receives DCO(tgt=N41,PS=x+1). The received Path Sequence is latest and thus it invalidates the entry associated with target (N41). (N33) then sends the DCO(tgt=N41,PS=x+1) to (N41). (N41) sees itself as the target and drops the DCO.
  9. From Step 6 above, (N31) receives the DAO(tgt=N41,PS=x+1,I\_flag=1). It creates a routing entry and sends the DAO(tgt=N41,PS=x+1,I\_flag=1) to (N21). Similarly (N21) receives the DAO and subsequently sends the DAO(tgt=N41,PS=x+1,I\_flag=1) to (N11).
  10. (N11) receives DAO(tgt=N41,PS=x+1,I\_flag=1) from (N21). It waits for DelayDCO timer since it has multiple routes to (N41). (N41) will receive DAO(tgt=N41,PS=x+1,I\_flag=1) from (N22) from Step 7 above. Thus (N11) has received regular DAO(tgt=N41,PS=x+1,I\_flag=1) from all paths and thus does not initiate DCO.
  11. (N11) forwards the DAO(tgt=N41,PS=x+1,I\_flag=1) to 6LBR and the full path is established.

#### Authors' Addresses

Rahul Arvind Jadhav (editor)  
Huawei  
Kundalahalli Village, Whitefield,  
Bangalore, Karnataka 560037  
India

Phone: +91-080-49160700  
Email: rahul.ietf@gmail.com

Pascal Thubert  
Cisco Systems, Inc  
Building D  
45 Allee des Ormes - BP1200  
MOUGINS - Sophia Antipolis 06254  
France

Phone: +33 497 23 26 34  
Email: pthubert@cisco.com

Rabi Narayan Sahoo  
Huawei  
Kundalahalli Village, Whitefield,  
Bangalore, Karnataka 560037  
India

Phone: +91-080-49160700  
Email: rabinarayans@huawei.com

Zhen Cao  
Huawei  
W Chang'an Ave  
Beijing  
P.R. China

Email: zhencao.ietf@gmail.com



roll  
Internet-Draft  
Intended status: Standards Track  
Expires: October 1, 2018

P. van der Stok, Ed.  
consultant  
March 30, 2018

A YANG model for Multicast Protocol for Low power and lossy Networks  
(MPL)  
draft-ietf-roll-mpl-yang-01

## Abstract

This document defines a YANG data model for management of Multicast Protocol for Low power and lossy Networks (MPL) implementations. The data model includes configuration data and state data.

## Note

Discussion and suggestions for improvement are requested, and should be sent to [roll@ietf.org](mailto:roll@ietf.org).

## Status of This Memo

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

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

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

This Internet-Draft will expire on October 1, 2018.

## Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

## Table of Contents

1. Introduction . . . . .	2
1.1. Terminology . . . . .	2
1.1.1. Tree Diagrams . . . . .	3
2. MPL model . . . . .	3
3. SID file generation . . . . .	5
4. yang-mpl modules . . . . .	8
4.1. yang-mpl-domain module . . . . .	8
4.2. yang-mpl-ops module . . . . .	11
4.3. yang-mpl-seeds module . . . . .	15
4.4. yang-mpl-statistics module . . . . .	19
5. IANA Considerations . . . . .	23
6. Acknowledgements . . . . .	23
7. Changelog . . . . .	23
8. References . . . . .	23
8.1. Normative References . . . . .	23
8.2. Informative References . . . . .	24
Author's Address . . . . .	24

## 1. Introduction

This document defines a YANG [RFC6020] data model for management of Multicast Protocol for Low power and lossy Networks (MPL) [RFC7731] implementations. The data model covers configuration of MPL parameters per interface. It also provides information about which Multicast addresses are operationally used, and the seeds of the forwarded packets.

### 1.1. Terminology

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

The following terms are defined in [RFC6241] and are not redefined here:

- o client
- o configuration data
- o server

- o state data

The following terms are defined in [RFC6020] and are not redefined here:

- o data model
- o data node

The terminology for describing YANG data models is found in [RFC6020].

Terms like message, domain, seed, I, k, c are defined in [RFC7731].

Mutiple copies of a message can be received or sent by a node.

#### 1.1.1. Tree Diagrams

A simplified graphical representation of the data model is used in the YANG modules specified in this document. The meaning of the symbols in these diagrams is as follows:

Brackets "[" and "]" enclose list keys.

Abbreviations before data node names: "rw" means configuration data (read-write) and "ro" state data (read-only).

Symbols after data node names: "?" means an optional node, "!" means a presence container, and "\*" denotes a list and leaf-list.

Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").

Ellipsis ("...") stands for contents of subtrees that are not shown.

## 2. MPL model

This document defines the YANG module "ietf-yang-mpl", which specifies a data model for MPL servers. The model is separated into four modules which can be loaded independently to accommodate the storage space to the wanted functionality. The model consists of the following parts: (1) a "mpl-domain" part that describes the MPL-domains and associated Multicast addresses and the interfaces on which the Multicast addresses are enabled, (2) a "mpl-op" part that describes the parameters settings per seed, (3) a "mpl-seeds" part that describes the MPL buffer contents and the Trickle timer values, and (4) a "mpl-statistics" part that describes the number of lost and

correctly forwarded messages. The data model, divided in four modules, has the following structure for MPL configuration per node:

```

module: ietf-yang-mpl-domain
  +--rw domain
    +--rw (single)?
      +--:(mpl-domain)
        |   +--rw mpl-domain
        |   |   +--rw domains* [domainID]
        |   |   |   +--rw domainID      uint16
        |   |   |   +--rw MClist*       inet:ipv6-address
        |   |   +--rw addresses* [MCaddress]
        |   |   |   +--rw MCaddress      inet:ipv6-address
        |   |   +--rw interfaces*      string
        |   +--:(mpl-single)
        |   |   +--rw mpl-single
        |   |   |   +--rw MCaddresses*   inet:ipv6-address
        |   |
        |   +--rw SE_LIFETIME?           uint16
        |   +--rw PROACTIVE_FORWARDING?   boolean
        |   +--rw SEED_SET_ENTRY_LIFETIME? uint64
        |   +--rw mpl-parameter* [domainID]
        |   |   +--rw domainID              uint16
        |   |   +--rw DATA_MESSAGE_IMIN?    uint16
        |   |   +--rw DATA_MESSAGE_IMAX?    uint16
        |   |   +--rw DATA_MESSAGE_K?       uint16
        |   |   +--rw DATA_MESSAGE_TIMER_EXPIRATIONS? uint16
        |   |   +--rw CONTROL_MESSAGE_IMIN?   uint16
        |   |   +--rw CONTROL_MESSAGE_IMAX?   uint16
        |   |   +--rw CONTROL_MESSAGE_K?      uint16
        |   |   +--rw CONTROL_MESSAGE_TIMER_EXPIRATIONS? uint16
        |
        +--ro mpl-seeds* [seedID domainID]
        |   +--ro seedID          uint64
        |   +--ro domainID        uint16
        |   +--ro local?          boolean
        |   +--ro generate-seqno?  uint8
        |   +--ro life-time?       uint64
        |   +--ro min-seqno?       uint8
        |   +--ro data-number?     uint8
        |   +--ro control-number?  uint8
        |   +--ro buffered-messages* [seqno]
        |   |   +--ro seqno      uint8
        |   |   +--ro I?         uint8

```

```

        +---ro c?          uint8
        +---ro e?          uint8
        +---ro t?          uint8

module: ietf-yang-mpl-statistics
  +---ro mpl-statistics* [seedID domainID]
    +---ro seedID          uint64
    +---ro domainID        uint16
    +---ro c-too-high?     uint64
    +---ro nr-forwarded?   uint64
    +---ro nr-of-messages-received? uint64
    +---ro nr-of-copies-received?  uint64
    +---ro nr-of-messages-forwarded? uint64
    +---ro nr-of-copies-forwarded?  uint64
    +---ro nr-of-refused?   uint64
    +---ro nr-of-missed?    uint64
    +---ro nr-of-notreceived? uint64
    +---ro nr-of-inconsistent-data? uint64
    +---ro nr-of-consistent-data?  uint64
    +---ro nr-of-consistent-control? uint64
    +---ro nr-of-inconsistent-control? uint64
    +---ro statistics-interval?  uint64
    +---x reset-statistics

```

### 3. SID file generation

SID are allocated to the identifiers specified in the four modules. Their values are:

SID	Assigned to
1004050	module ietf-yang-mpl-domain
1004051	data /ietf-yang-mpl-domain:domain
1004052	data /ietf-yang-mpl-domain:domain/mpl-domain
1004053	data /ietf-yang-mpl-domain:domain/mpl-domain/addresses
1004054	data /ietf-yang-mpl-domain:domain/mpl-domain /addresses/MCaddress
1004055	data /ietf-yang-mpl-domain:domain/mpl-domain /addresses/interfaces
1004056	data /ietf-yang-mpl-domain:domain/mpl-domain/domains
1004057	data /ietf-yang-mpl-domain:domain/mpl-domain /domains/MClist
1004058	data /ietf-yang-mpl-domain:domain/mpl-domain /domains/domainID
1004059	data /ietf-yang-mpl-domain:domain/mpl-single
1004060	data /ietf-yang-mpl-domain:domain/mpl-single

## /MCaddresses

File ietf-yang-mpl-domain@2018-03-29.sid created

Number of SIDs available : 50

Number of SIDs used : 11

SID	Assigned to
-----	-----
1004100	module ietf-yang-mpl-ops
1004101	data /ietf-yang-mpl-ops:mpl-ops
1004102	data /ietf-yang-mpl-ops:mpl-ops/PROACTIVE_FORWARDING
1004103	data /ietf-yang-mpl-ops:mpl-ops /SEED_SET_ENTRY_LIFETIME
1004104	data /ietf-yang-mpl-ops:mpl-ops/SE_LIFETIME
1004105	data /ietf-yang-mpl-ops:mpl-ops/mpl-parameter
1004106	data /ietf-yang-mpl-ops:mpl-ops/mpl-parameter /CONTROL_MESSAGE_IMAX
1004107	data /ietf-yang-mpl-ops:mpl-ops/mpl-parameter /CONTROL_MESSAGE_IMIN
1004108	data /ietf-yang-mpl-ops:mpl-ops/mpl-parameter /CONTROL_MESSAGE_K
1004109	data /ietf-yang-mpl-ops:mpl-ops/mpl-parameter /CONTROL_MESSAGE_TIMER_EXPIRATIONS
1004110	data /ietf-yang-mpl-ops:mpl-ops/mpl-parameter /DATA_MESSAGE_IMAX
1004111	data /ietf-yang-mpl-ops:mpl-ops/mpl-parameter /DATA_MESSAGE_IMIN
1004112	data /ietf-yang-mpl-ops:mpl-ops/mpl-parameter /DATA_MESSAGE_K
1004113	data /ietf-yang-mpl-ops:mpl-ops/mpl-parameter /DATA_MESSAGE_TIMER_EXPIRATIONS
1004114	data /ietf-yang-mpl-ops:mpl-ops/mpl-parameter/domainID

File ietf-yang-mpl-ops@2018-03-29.sid created

Number of SIDs available : 50

Number of SIDs used : 15

SID	Assigned to
-----	-----
1004150	module ietf-yang-mpl-seeds
1004151	data /ietf-yang-mpl-seeds:mpl-seeds
1004152	data /ietf-yang-mpl-seeds:mpl-seeds/buffered-messages
1004153	data /ietf-yang-mpl-seeds:mpl-seeds /buffered-messages/I
1004154	data /ietf-yang-mpl-seeds:mpl-seeds /buffered-messages/c
1004155	data /ietf-yang-mpl-seeds:mpl-seeds /buffered-messages/e

```

1004156  data /ietf-yang-mpl-seeds:mpl-seeds
          /buffered-messages/seqno
1004157  data /ietf-yang-mpl-seeds:mpl-seeds
          /buffered-messages/t
1004158  data /ietf-yang-mpl-seeds:mpl-seeds/control-number
1004159  data /ietf-yang-mpl-seeds:mpl-seeds/data-number
1004160  data /ietf-yang-mpl-seeds:mpl-seeds/domainID
1004161  data /ietf-yang-mpl-seeds:mpl-seeds/generate-seqno
1004162  data /ietf-yang-mpl-seeds:mpl-seeds/life-time
1004163  data /ietf-yang-mpl-seeds:mpl-seeds/local
1004164  data /ietf-yang-mpl-seeds:mpl-seeds/min-seqno
1004165  data /ietf-yang-mpl-seeds:mpl-seeds/seedID

```

File ietf-yang-mpl-seeds@2018-03-29.sid created

Number of SIDs available : 50

Number of SIDs used : 16

SID	Assigned to
1004200	module ietf-yang-mpl-statistics
1004201	data /ietf-yang-mpl-statistics:mpl-statistics
1004202	data /ietf-yang-mpl-statistics:mpl-statistics /c-too-high
1004203	data /ietf-yang-mpl-statistics:mpl-statistics /domainID
1004204	data /ietf-yang-mpl-statistics:mpl-statistics /nr-forwarded
1004205	data /ietf-yang-mpl-statistics:mpl-statistics /nr-of-consistent-control
1004206	data /ietf-yang-mpl-statistics:mpl-statistics /nr-of-consistent-data
1004207	data /ietf-yang-mpl-statistics:mpl-statistics /nr-of-copies-forwarded
1004208	data /ietf-yang-mpl-statistics:mpl-statistics /nr-of-copies-received
1004209	data /ietf-yang-mpl-statistics:mpl-statistics /nr-of-inconsistent-control
1004210	data /ietf-yang-mpl-statistics:mpl-statistics /nr-of-inconsistent-data
1004211	data /ietf-yang-mpl-statistics:mpl-statistics /nr-of-messages-forwarded
1004212	data /ietf-yang-mpl-statistics:mpl-statistics /nr-of-messages-received
1004213	data /ietf-yang-mpl-statistics:mpl-statistics /nr-of-missed
1004214	data /ietf-yang-mpl-statistics:mpl-statistics /nr-of-notreceived
1004215	data /ietf-yang-mpl-statistics:mpl-statistics

```
1004216    data /ietf-yang-mpl-statistics:mpl-statistics
                                                    /nr-of-refused
                                                    /reset-statistics
1004217    data /ietf-yang-mpl-statistics:mpl-statistics/seedID
1004218    data /ietf-yang-mpl-statistics:mpl-statistics
                                                    /statistics-interval
```

```
File ietf-yang-mpl-statistics@2018-03-29.sid created
Number of SIDs available : 50
Number of SIDs used : 19
```

#### 4. yang-mpl modules

This section describes four yang modules. The model is based on the MPL specification published in [RFC7731] and the specification of [RFC6206]. The identification of the interfaces follows the specification of ietf-interfaces of [RFC7223].

The data model allows to set values to the parameters of the MPL algorithm. This approach requires an active manager process to set the values without use of DHCP as described in: [RFC7774].

The names of the four modules are: yang-mpl-domain, yang-mpl-ops, yang-mpl-seeds, and yang-mpl-statistics, described in subsections with the same name.

##### 4.1. yang-mpl-domain module

This module describes (1) the MPL domains and the associated multicast addresses, and (2) the interfaces and the multicast addresses for which they are enabled.

The model features a choice such that for constrained devices with only one "single" interface and only one "single" domain, the model specifies a list of MC addresses for which the single interface is enabled.

```
<CODE BEGINS>file "ietf-yang-mpl-domain@2018-03-29.yang"
```

```
module ietf-yang-mpl-domain {
    yang-version 1.1;
    namespace
        "urn:ietf:params:xml:ns:yang:ietf-yang-mpl-domain";
```



```
prefix mpl;

import ietf-inet-types{
  prefix inet;
}

organization
  "IETF ROLL (Routing Over Low power and lossy networks)
  Working Group";

contact
  "WG Web:   http://tools.ietf.org/wg/roll/
  WG List:  mailto:roll@ietf.org

  WG Chair: Peter van der Stok
            mailto:consultancy@vanderstok.org

  WG Chair: Ines Robles
            mailto:maria.ines.robles@ericsson.com

  Editor:   Peter van der Stok
            mailto:consultancy@vanderstok.org";

description
  "This module contains information about the state of the MPL domain.

  Copyright (c) 2016 IETF Trust and the persons identified as
  authors of the code. All rights reserved.

  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject
  to the license terms contained in, the Simplified BSD License
  set forth in Section 4.c of the IETF Trust's Legal Provisions
  Relating to IETF Documents
  (http://trustee.ietf.org/license-info).

  This version of this YANG module is part of RFC XXXX; see
  the RFC itself for full legal notices.";

revision "2018-03-29" {
  description "Initial revision.";
  reference
    "I-D:draft-ietf-roll-mpl-yang: A YANG model for Multicast
    Protocol for Low power and lossy Networks (MPL)";
}

container domain {
  description
    "High level container containing the choice statement.";
```

```
choice single {
  description
    "A choice between single domain/interface and multiple
    domains and interfaces.";
  container mpl-domain {
    description
      "The entries describe the MPL domains, the associated
      Multicast addresses and interfaces.";

    list domains {
      key domainID;
      description
        "The entries describe a given domain identified with
        domainID and the associated Multicast addresses.";

      leaf domainID {
        type uint16;
        description
          "Entry uniquely identifies the domain in the
          forwarder.";
      }

      leaf-list Mclist{
        type inet:ipv6-address;
        description
          "List of associated IPv6 Addresses.";
      }
    } // domains list

    list addresses {
      key MCaddress;
      description
        "The entries describe the interfaces enabled
        with the specified MC address.";

      leaf MCaddress {
        type inet:ipv6-address;
        description
          "MC address belonging to a MPL domain.";
      }

      leaf-list interfaces {
        type string;
        description
          "List of names of interfaces enabled for this
          Multicast address. Interface name is defined in [RFC6206].";
      }
    } // addresses list
  }
}
```

```
    } // container mpl-domain
    container mpl-single {
        description
            "For small devices list of MC addresses for single
            interface and domain.";
        leaf-list MCaddresses{
            type inet:ipv6-address;
            description
                "list of MC addresses belonging to one single
                domain and interface.";
        }
    } // container mpl-simple
} // choice simple
} // container module
} //module ietf-yang-mpl-domain
```

<CODE ENDS>

#### 4.2. yang-mpl-ops module

This module models the operational aspects of MPL. Per domain MPL specifies four parameters I\_MAX, I\_MIN, K, and TIMER\_EXPIRATIONS for data and control messages. The value of the MPL intervals are expressed in TUNIT. The entry SE\_LIFETIME taken over from [RFC7774] fixes TUNIT to milliseconds. For very constrained devices with only one domain there can be only one instance of mpl-parameter list.

```
<CODE BEGINS>file "ietf-yang-mpl-ops@2018-03-29.yang"

module ietf-yang-mpl-ops {

    yang-version 1.1;

    namespace
        "urn:ietf:params:xml:ns:yang:ietf-yang-mpl-ops";

    prefix mpl;

    organization
        "IETF ROLL (Routing over Low power and lossy networks)
        Working Group";

    contact
        "WG Web:  http://tools.ietf.org/wg/roll/
        WG List:  mailto:roll@ietf.org"
```

WG Chair: Peter van der Stok  
mailto:consultancy@vanderstok.org

WG Chair: Ines Robles  
mailto:maria.ines.robles@ericsson.com

Editor: Peter van der Stok  
mailto:consultancy@vanderstok.org";

description

"This module contains information about the operation of the MPL protocol.

Copyright (c) 2016 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or

without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.";

```
revision "2018-03-29" {  
  description "Initial revision.";  
  reference  
    "I-D:draft-ietf-roll-mpl-yang: A YANG model for Multicast  
    Protocol for Low power and lossy Networks (MPL)";  
}
```

```
container mpl-ops {  
  description  
    "Parameter settings for each MPL server and for each  
    individual domain of the server.";
```

```
  leaf SE_LIFETIME {  
    type uint16;  
    description  
      "lifetime in milliseconds/(mpl timer units),  
      equivalent to SEED_SET_ENTRY_LIFETIME/TUNIT as  
      specified in RFC7774.";  
  }
```

```
leaf PROACTIVE_FORWARDING {
  type boolean;
  description
    "The boolean value indicates whether the MPL forwarder
    schedules MPL data message transmission after
    receiving them for the first time.";
}

leaf SEED_SET_ENTRY_LIFETIME {
  type uint64;
  description
    "The value indicates the minimum lifetime for an entry
    in the Seed set expressed in seconds. Default value
    is 30 minutes.";
}

list mpl-parameter{
  key domainID;
  description
    "Each domain has a set of MPL forwarding parameters
    which regulate the forwarding operation.";

  leaf domainID{
    type uint16;
    description
      "Each domainID must be present in
mpl-parameter list.";
  }

  leaf DATA_MESSAGE_IMIN{
    type uint16;
    description
      "The minimum Trickle timer interval, as defined in
      [RFC6206], for MPL Data Message transmissions.";
  }

  leaf DATA_MESSAGE_IMAX{
    type uint16;
    description
      "The maximum Trickle timer interval, as defined in
      [RFC6206], for MPL Data Message transmissions.";
  }

  leaf DATA_MESSAGE_K{
    type uint16;
    default 1;
    description
      "The redundancy constant, as defined in [RFC6206], for
```

```
        MPL Data Message transmissions.";
    }

    leaf DATA_MESSAGE_TIMER_EXPIRATIONS{
        type uint16;
        default 3;
        description
            "The number of Trickle timer expirations, as defined
             in [RFC7731], that occur
             before terminating the Trickle algorithm's
             retransmission of a given MPL Data Message.";
    }

    leaf CONTROL_MESSAGE_IMIN{
        type uint16;
        description
            "The minimum Trickle timer interval, as defined
             in [RFC6206], for MPL Control Message
             transmissions.";
    }

    leaf CONTROL_MESSAGE_IMAX{
        type uint16;
        description
            "The maximum Trickle timer interval, as defined
             in [RFC6206], for MPL Control Message
             transmissions.";
    }

    leaf CONTROL_MESSAGE_K{
        type uint16;
        default 1;
        description
            "The redundancy constant, as defined in [RFC6206],
             for MPL Control Message transmissions.";
    }

    leaf CONTROL_MESSAGE_TIMER_EXPIRATIONS{
        type uint16;
        default 10;
        description
            "The number of Trickle time expirations,
             as defined in [RFC7731], that occur
             before terminating the Trickle algorithm
             for MPL Control Message transmissions.";
    }
} // list MPL-parameter
```

```
    } // container MPL-ops  
  } // module ietf-yang-mpl-ops
```

```
<CODE ENDS>
```

#### 4.3. yang-mpl-seeds module

This module specifies the current values of the operation of the MPL forwarder. The values are acquired by the client and set by the server. The module specifies a set of message buffers, with a buffer per seed and domain. In constrained devices there will be only one domain, but probably multiple seeds.

The message buffer contains a set of messages where each message is uniquely identified by its sequence number and seed. The associated I, c, e, and t values indicate the progress of MPL with respect to this message, as specified in [RFC7731]. A forwarder sends and receives multiple copies of a message. When a forwarder has sent (received) a copy of a message, the forwarder has sent (received) that message.

For forwarders which are seeds, local has value true and seqno is the sequence number of the next message to send.

```
<CODE BEGINS>file "ietf-yang-mpl-seeds@2018-03-29.yang"
```

```
module ietf-yang-mpl-seeds {  
  
  yang-version 1.1;  
  
  namespace  
    "urn:ietf:params:xml:ns:yang:ietf-yang-mpl-seeds";  
  
  prefix mpl;  
  
  organization  
    "IETF ROLL (Routing over Low power and lossy networks)  
    Working Group";  
  
  contact  
    "WG Web:  http://tools.ietf.org/wg/roll/  
    WG List:  mailto:roll@ietf.org  
  
    WG Chair: Peter van der Stok  
              mailto:consultancy@vanderstok.org"
```

WG Chair: Ines Robles  
mailto:maria.ines.robles@ericsson.com

Editor: Peter van der Stok  
mailto:consultancy@vanderstok.org";

description

"This module contains information about the operation  
of the MPL protocol.

Copyright (c) 2016 IETF Trust and the persons identified as  
authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or

without modification, is permitted pursuant to, and subject  
to the license terms contained in, the Simplified BSD License  
set forth in Section 4.c of the IETF Trust's Legal Provisions  
Relating to IETF Documents  
(<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see  
the RFC itself for full legal notices.";

```
revision "2018-03-29" {  
  description "Initial revision.";  
  reference  
    "I-D:draft-ietf-roll-mpl-yang: A YANG model for Multicast  
    Protocol for Low power and lossy Networks (MPL)";  
}
```

```
list mpl-seeds{  
  key "seedID domainID";  
  config false;  
  description  
    "List describes all seeds that are active in the server.  
    Seed information contains the message buffer contents and the  
    operational values of I, c, sequence number and the life-times  
    per message.";
```

```
  leaf seedID{  
    type uint64;  
    description  
      "value uniquely identifies the MPL Seed within a MPL  
      domain.";  
  }
```



```
    leaf domainID{
      type uint16;
      description
        "together with seedID uniquely identifies buffer set.";
    }

    leaf local {
      type boolean;
      description
        "When local == TRUE, seed originated in this forwarder.
        WHEN local == false, seed originated in different
        forwarder.";
    }

    leaf generate-seqno {
      type uint8;
      description
        "Sequence number of next message to be generated by
this local seed.";
    }

    leaf life-time {
      type uint64;
      description
        " Minimum remaining lifetime of the seed entry in
        SE_LIFETIME units.";
    }

    leaf min-seqno{
      type uint8;
      description
        "Lower bound sequence number in the buffer of the seed.";
    }

    leaf data-number{
      type uint8;
      description
        "Number of currently buffered data messages.";
    }

    leaf control-number{
      type uint8;
      description
        "Number of currently buffered control messages.";
    }

    list buffered-messages{
      key seqno;
```

```
    description
        " status of trickle intervals of the buffered message
identified by seqno. and seed/domain";

    leaf seqno{
        type uint8;
        description
            "Sequence number of message.";
    }

    leaf I{
        type uint8;
        description
            "Current Trickle timer interval size in SE-LIFETIME
units.";
    }

    leaf c{
        type uint8;
        description
            "number of times that a copy of this message has been
            received in this interval.";
    }

    leaf e{
        type uint8;
        description
            "number of Trickle time expirations since last
            Trickle timer reset.";
    }

    leaf t{
        type uint8;
        description
            " Time expressed in SE-LIFETIME units
            that message will be (is) forwarded";
    }
} // list seed-timers
} // list MPL-seeds
} // module ietf-yang-mpl-seeds
```

<CODE ENDS>

#### 4.4. yang-mpl-statistics module

This module specifies the operation of the MPL forwarder expressed in number of messages and copies. The values are acquired by the client and set by the server. Statistics are specified per seed and domain. In constrained devices there will be only one domain, but probably multiple seeds.

The parameter *k* determines how many copies of a message can be forwarded. The counters *c-too-high*, *nr-forwarded*, and *nr-not-forwarded* give insight in the consequences of the current value of *k*.

The other counters give insight in the loss of messages caused by the medium or forwarding delays. The inconsistent/consistent counters indicate when consistent or inconsistent messages were received according to the definition of consistent in [RFC7731].

```
<CODE BEGINS>file "ietf-yang-mpl-statistics@2018-03-29.yang"
```

```
module ietf-yang-mpl-statistics {  
  
    yang-version 1.1;  
  
    namespace  
        "urn:ietf:params:xml:ns:yang:ietf-yang-mpl-statistics";  
  
    prefix mpl;  
  
    organization  
        "IETF ROLL (Routing over Low power and lossy networks)  
        Working Group";  
  
    contact  
        "WG Web:  http://tools.ietf.org/wg/roll/  
        WG List:  mailto:roll@ietf.org  
  
        WG Chair: Peter van der Stok  
                  mailto:consultancy@vanderstok.org  
  
        WG Chair: Ines Robles  
                  mailto:maria.ines.robles@ericsson.com  
  
        Editor:   Peter van der Stok  
                  mailto:consultancy@vanderstok.org";
```

```
description
  "This module contains information about the operation
  of the MPL protocol.

  Copyright (c) 2016 IETF Trust and the persons identified as
  authors of the code. All rights reserved.

  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject
  to the license terms contained in, the Simplified BSD License
  set forth in Section 4.c of the IETF Trust's Legal Provisions
  Relating to IETF Documents
  (http://trustee.ietf.org/license-info).

  This version of this YANG module is part of RFC XXXX; see
  the RFC itself for full legal notices."

revision "2018-03-29" {
  description "Initial revision.";
  reference
    "I-D:draft-ietf-roll-mpl-yang: A YANG model for Multicast
    Protocol for Low power and lossy Networks (MPL)";
}

list mpl-statistics{
  key "seedID domainID";

  config false;

  description
    "List describes performance statistics integrated over
    the messages identified by seed and domain identifiers. A
    forwarder can receive and forward multiple copies of a message
    uniquely identified by seqno, domain, and seed.";

  leaf seedID{
    type uint64;
    description
      "value uniquely identifies the MPL Seed within a MPL
      domain.";
  }

  leaf domainID{
    type uint16;
    description
```

```
    "together with seed-ID uniquely identifies buffer set.";
  }

  leaf c-too-high {
    type uint64;
    description
      "Number of times that a copy was not forwarded
       because c > k.";
  }

  leaf nr-forwarded {
    type uint64;
    description
      "number of times copies are forwarded, while c <= k.";
  }

  leaf nr-of-messages-received{
    type uint64;
    description
      "number of messages received,
       must be smaller than or equal to seqno.";
  }

  leaf nr-of-copies-received{
    type uint64;
    description
      "total number of message copies received.";
  }

  leaf nr-of-messages-forwarded{
    type uint64;
    description
      "number of forwarded messages, must be smaller than or
equal to nr-of-messages-received.";
  }

  leaf nr-of-copies-forwarded{
    type uint64;
    description
      "number of forwarded copies, can be larger than
number-of-copies-received.";
  }

  leaf nr-of-refused{
    type uint64;
    description
      "number of refused copies because seqno too small.";
  }
```

```
leaf nr-of-missed{
  type uint64;
  description
    "number of messages that were not received
     is equal to the number of empty message buffers
     seqno < min-seqno.";
}

leaf nr-of-notreceived{
  type uint64;
  description
    "number of messages that were not received
     according to control message.";
}

leaf nr-of-inconsistent-data{
  type uint64;
  description
    "number of inconsistent data messages.";
}

leaf nr-of-consistent-data{
  type uint64;
  description
    "number of consistent data messages.";
}

leaf nr-of-consistent-control{
  type uint64;
  description
    "number of consistent control messages.";
}

leaf nr-of-inconsistent-control{
  type uint64;
  description
    "number of inconsistent control messages.";
}

leaf statistics-interval{
  type uint64;
  description
    "Interval, expressed in seconds, during which
     the statistics are collected.";
}

action reset-statistics{
  description
```

```
        "set all statistics counters and
        statistics-interval to zero.";
    }

    } // list mpl statistics
} // module ietf-yang-mpl-statistics

<CODE ENDS>
```

## 5. IANA Considerations

This specification has no consequences for IANA.

## 6. Acknowledgements

Andy Bierman has commented on the use of YANG for mpl. YANG doctors pointed out a wrong use of config.

## 7. Changelog

Changes from version 00 to version 01

- o config false in "statistics" and "seeds" modules
- o separated into 4 modules
- o inserted choice in domain modules
- o more explanatory text
- o renamed some parameters
- o Introduced section per module
- o reset of statistics is added

## 8. References

### 8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, DOI 10.17487/RFC7223, May 2014, <<https://www.rfc-editor.org/info/rfc7223>>.
- [RFC7731] Hui, J. and R. Kelsey, "Multicast Protocol for Low-Power and Lossy Networks (MPL)", RFC 7731, DOI 10.17487/RFC7731, February 2016, <<https://www.rfc-editor.org/info/rfc7731>>.

## 8.2. Informative References

- [RFC6206] Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko, "The Trickle Algorithm", RFC 6206, DOI 10.17487/RFC6206, March 2011, <<https://www.rfc-editor.org/info/rfc6206>>.
- [RFC7774] Doi, Y. and M. Gillmore, "Multicast Protocol for Low-Power and Lossy Networks (MPL) Parameter Configuration Option for DHCPv6", RFC 7774, DOI 10.17487/RFC7774, March 2016, <<https://www.rfc-editor.org/info/rfc7774>>.

## Author's Address

Peter van der Stok (editor)  
consultant

Phone: +31-492474673 (Netherlands), +33-966015248 (France)  
Email: [consultancy@vanderstok.org](mailto:consultancy@vanderstok.org)  
URI: [www.vanderstok.org](http://www.vanderstok.org)



roll  
Internet-Draft  
Intended status: Standards Track  
Expires: January 16, 2019

P. van der Stok, Ed.  
consultant  
July 15, 2018

A YANG model for Multicast Protocol for Low power and lossy Networks  
(MPL)  
draft-ietf-roll-mpl-yang-02

## Abstract

This document defines a YANG data model for management of Multicast Protocol for Low power and lossy Networks (MPL) implementations. The data model includes configuration data and state data.

## Note

Discussion and suggestions for improvement are requested, and should be sent to [roll@ietf.org](mailto:roll@ietf.org).

## Status of This Memo

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

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

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

This Internet-Draft will expire on January 16, 2019.

## Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

## Table of Contents

1. Introduction . . . . .	2
1.1. Terminology . . . . .	2
1.1.1. Tree Diagrams . . . . .	3
2. MPL model . . . . .	3
2.1. NMDA considerations . . . . .	4
2.2. MPL-domain tree . . . . .	4
2.3. MPL-ops tree . . . . .	5
2.4. MPL-seeds tree . . . . .	5
2.5. MPL-statistics tree . . . . .	6
3. SID file generation . . . . .	7
3.1. mpl-domain . . . . .	7
3.2. mpl-ops . . . . .	7
3.3. mpl-seeds . . . . .	8
3.4. mpl-statistics . . . . .	9
4. yang-mpl modules . . . . .	11
4.1. yang-mpl-domain module . . . . .	11
4.2. yang-mpl-ops module . . . . .	14
4.3. yang-mpl-seeds module . . . . .	18
4.4. yang-mpl-statistics module . . . . .	22
5. IANA Considerations . . . . .	27
6. Acknowledgements . . . . .	28
7. Changelog . . . . .	28
8. References . . . . .	29
8.1. Normative References . . . . .	29
8.2. Informative References . . . . .	29
Author's Address . . . . .	30

## 1. Introduction

This document defines a YANG [RFC7950] data model for management of Multicast Protocol for Low power and lossy Networks (MPL) [RFC7731] implementations. The data model covers configuration of MPL parameters per interface. It also provides information about which Multicast addresses are operationally used, and the seeds of the forwarded packets.

### 1.1. Terminology

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

The following terms are defined in [RFC6241] and are not redefined here:

- o client
- o configuration data
- o server
- o state data

The following terms are defined in [RFC7950] and are not redefined here:

- o data model
- o data node

The terminology for describing YANG data models is found in [RFC7950].

Terms like message, domain, seed, I, k, c are defined in [RFC7731].

Terms like YANG Schema Item iDentifier (SID) and delta are defined in [I-D.ietf-core-sid].

Multiple copies of a message can be received or sent by a node.

#### 1.1.1. Tree Diagrams

YANG tree diagrams provide a concise representation of a YANG module, and SHOULD be included to help readers understand YANG module structure. Guidelines on tree diagrams can be found in Section 3 of [I-D.ietf-netmod-yang-tree-diagrams]. Tree diagrams used in this document follow the notation defined in [RFC8340].

## 2. MPL model

This document defines the YANG modules "ietf-yang-mpl-xxx", which specify a data model for MPL servers. The model is separated into four modules which can be loaded independently to accommodate the storage space to the wanted functionality. The model consists of the following parts: (1) "mpl-domain", (2) "mpl-op", (3) "mpl-seeds", and (4) "mpl-statistics". The four models are discussed below accompanied by their trees.

## 2.1. NMDA considerations

The Network Management Data Architecture (NMDA) is specified in [RFC8342]. The MPL protocol is designed for low-resource nodes, where the hardware is fixed once and for all. The configuration of the node in the MPL context is limited to enabling the interface(s) with Multicast addresses. The interface configuration of the YANG module replaces the multicast address assignment of section 2.6.2 of [RFC7390].

Consequently, The contents of the "candidate", "startup", "running", and "intended" datastores are identical. The conceptual datastore consists of an "operational" and an "intended" datastore. The "intended" store contains the nodes of the modules: ietf-yang-mpl-domain, and ietf-yang-mpl-ops. The "running" store contains the nodes of the modules: ietf-yang-mpl-seeds, and ietf-yang-mpl-statistics. The ietf-yang-mpl-domain is necessary when any MPL management is wanted. The two modules of the "running" store need the two modules of the "intended" store, and can be loaded dependent on the management wishes and resource constraints.

The modules ietf-yang-mpl-ops, ietf-yang-mpl-seeds, and ietf-yang-mpl-statistics "augment" ietf-yang-mpl-domain.

## 2.2. MPL-domain tree

The "mpl-domain" module describes the MPL-domains and associated Multicast addresses and the interfaces on which the Multicast addresses are enabled. The model allow for a short single MPL-domain configuration or a multi-domain configuration that needs more storage space.

```

module: ietf-yang-mpl-domain
  +--rw domain
    +--rw (multiple)?
      +--:(mpl-domain)
        |   +--rw mpl-domain
        |   |   +--rw domains* [domainID]
        |   |   |   +--rw domainID      uint16
        |   |   |   +--rw MClst*        inet:ipv6-address
        |   |   +--rw addresses* [MCaddress]
        |   |   |   +--rw MCaddress      inet:ipv6-address
        |   |   |   +--rw interfaces*   string
        |   +--:(mpl-single)
        |   |   +--rw mpl-single
        |   |   |   +--rw MCaddresses*   inet:ipv6-address

```

### 2.3. MPL-ops tree

The generator node of a MPL message is called a seed that emits on a multicast address of a domain. The "mpl-ops" module describes the operational parameters settings per domain. The parameters determine the dynamics of the message reception intervals per domain [RFC7731].

```
module: ietf-yang-mpl-ops
  augment /mpl:domain:
    +--rw mpl-ops
      +--rw PROACTIVE_FORWARDING?      boolean
      +--rw SEED_SET_ENTRY_LIFETIME?   uint64
      +--rw mpl-parameter* [domainID]
        +--rw domainID                  uint16
        +--rw DATA_MESSAGE_IMIN?      uint16
        +--rw DATA_MESSAGE_IMAX?      uint16
        +--rw DATA_MESSAGE_K?         uint16
        +--rw DATA_MESSAGE_TIMER_EXPIRATIONS? uint16
        +--rw CONTROL_MESSAGE_IMIN?    uint16
        +--rw CONTROL_MESSAGE_IMAX?    uint16
        +--rw CONTROL_MESSAGE_K?       uint16
        +--rw CONTROL_MESSAGE_TIMER_EXPIRATIONS? uint16
```

### 2.4. MPL-seeds tree

The "mpl-seeds" part describes the MPL buffer contents and the Trickle timer values associated with each seed and domain. Multiple seeds per domain may exist. The module exposes the state of the MPL buffer and key information about the messages in the MPL buffers at a given acquisition moment.

```
module: ietf-yang-mpl-seeds
augment /mpl:domain:
  +--rw SE_LIFETIME?      uint16
  +--ro mpl-seeds* [seedID domainID]
    +--ro seedID          uint64
    +--ro domainID        uint16
    +--ro local?          boolean
    +--ro generate-seqno?  uint8
    +--ro life-time?      uint64
    +--ro min-seqno?      uint8
    +--ro data-number?    uint8
    +--ro control-number? uint8
    +--ro buffered-messages* [seqno]
      +--ro seqno      uint8
      +--ro I?         uint8
      +--ro c?         uint8
      +--ro e?         uint8
      +--ro t?         uint8
```

## 2.5. MPL-statistics tree

The "mpl-statistics" module describes the number of lost and correctly forwarded messages and its copies.

```
module: ietf-yang-mpl-statistics
augment /mpl:domain:
  +--ro mpl-statistics* [seedID domainID]
    +--ro seedID          uint64
    +--ro domainID        uint16
    +--ro c-too-high?     uint64
    +--ro nr-forwarded?   uint64
    +--ro nr-of-messages-received? uint64
    +--ro nr-of-messages-forwarded? uint64
    +--ro nr-of-copies-received? uint64
    +--ro nr-of-copies-forwarded? uint64
    +--ro nr-of-refused-copies? uint64
    +--ro nr-of-missed-messages? uint64
    +--ro nr-of-notreceived-messages? uint64
    +--ro nr-of-inconsistent-data? uint64
    +--ro nr-of-consistent-data? uint64
    +--ro nr-of-consistent-control? uint64
    +--ro nr-of-inconsistent-control? uint64
    +--ro statistics-interval? uint64
    +---x reset-buffer-statistics
    +---x reset-all-statistics
```

### 3. SID file generation

YANG Schema Item iDentifiers (SID) are allocated to replace the relatively long YANG identifiers by the much shorter SIDs. Consequently, the payload size is considerably reduced. The assignment tool is <https://comi.space/>. SIDs are allocated to the identifiers specified in the four modules. Their values are:

#### 3.1. mpl-domain

Module : ietf-yang-mpl-domain  
Revision : 2018-07-06

SID	Namespace	YANG identifier
1004000	module	ietf-yang-mpl-domain
1004001	data	/ietf-yang-mpl-domain:domain
1004002	data	/ietf-yang-mpl-domain:domain/mpl-domain
1004003	data	/ietf-yang-mpl-domain:domain/mpl-domain /addresses
1004004	data	/ietf-yang-mpl-domain:domain/mpl-domain /addresses/MCaddress
1004005	data	/ietf-yang-mpl-domain:domain/mpl-domain /addresses/interfaces
1004006	data	/ietf-yang-mpl-domain:domain/mpl-domain /domains
1004007	data	/ietf-yang-mpl-domain:domain/mpl-domain /domains/MClist
1004008	data	/ietf-yang-mpl-domain:domain/mpl-domain /domains/domainID
1004009	data	/ietf-yang-mpl-domain:domain/mpl-single
1004010	data	/ietf-yang-mpl-domain:domain/mpl-single /MCaddresses

#### 3.2. mpl-ops

Module : ietf-yang-mpl-ops  
Revision : 2018-07-06

SID	Namespace	YANG identifier
1004050	module	ietf-yang-mpl-ops
1004051	data	/ietf-yang-mpl-domain:domain
		/ietf-yang-mpl-ops:mpl-ops
1004052	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-ops: mpl-ops/PROACTIVE_FORWARDING
1004053	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-ops: mpl-ops/SEED_SET_ENTRY_LIFETIME
1004054	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-ops: mpl-ops/mpl-parameter
1004055	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-ops: mpl-ops/mpl-parameter/CONTROL_MESSAGE_IMAX
1004056	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-ops: mpl-ops/mpl-parameter/CONTROL_MESSAGE_IMIN
1004057	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-ops: mpl-ops/mpl-parameter/CONTROL_MESSAGE_K
1004058	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-ops: mpl-ops/mpl-parameter/CONTROL_MESSAGE_TIMER_EXPIRATIONS
1004059	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-ops: mpl-ops/mpl-parameter/DATA_MESSAGE_IMAX
1004060	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-ops: mpl-ops/mpl-parameter/DATA_MESSAGE_IMIN
1004061	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-ops: mpl-ops/mpl-parameter/DATA_MESSAGE_K
1004062	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-ops: mpl-ops/mpl-parameter/DATA_MESSAGE_TIMER_EXPIRATIONS
1004063	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-ops: mpl-ops/mpl-parameter/domainID

### 3.3. mpl-seeds



Module : ietf-yang-mpl-seeds  
Revision : 2018-07-06

SID	Namespace	YANG identifier
1004100	module	ietf-yang-mpl-seeds
1004101	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-seeds:SE_LIFETIME
1004102	data	/ietf-yang-mpl-domain:domain /ietf-yang-mpl-seeds:mpl-seeds
1004103	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-seeds: mpl-seeds/buffered-messages
1004104	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-seeds: mpl-seeds/buffered-messages/I
1004105	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-seeds: mpl-seeds/buffered-messages/c
1004106	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-seeds: mpl-seeds/buffered-messages/e
1004107	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-seeds: mpl-seeds/buffered-messages/seqno
1004108	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-seeds: mpl-seeds/buffered-messages/t
1004109	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-seeds: mpl-seeds/control-number
1004110	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-seeds: mpl-seeds/data-number
1004111	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-seeds: mpl-seeds/domainID
1004112	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-seeds: mpl-seeds/generate-seqno
1004113	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-seeds: mpl-seeds/life-time
1004114	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-seeds: mpl-seeds/local
1004115	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-seeds: mpl-seeds/min-seqno
1004116	data	/ietf-yang-mpl-domain:domain/ietf-yang-mpl-seeds: mpl-seeds/seedID

### 3.4. mpl-statistics

Module : ietf-yang-mpl-statistics  
Revision : 2018-07-06

SID	Namespace	YANG identifier
1004150	module	ietf-yang-mpl-statistics
1004151	data	/ietf-yang-mpl-domain:domain/ ietf-yang-mpl-statistics:mpl-statistics
1004152	data	/ietf-yang-mpl-domain:domain/

```
    ietf-yang-mpl-statistics:mpl-statistics/c-too-high
1004153 data    /ietf-yang-mpl-domain:domain/
    ietf-yang-mpl-statistics:mpl-statistics/domainID
1004154 data    /ietf-yang-mpl-domain:domain/
    ietf-yang-mpl-statistics:mpl-statistics/nr-forwarded
1004155 data    /ietf-yang-mpl-domain:domain/
    ietf-yang-mpl-statistics:mpl-statistics
    /nr-of-consistent-control
1004156 data    /ietf-yang-mpl-domain:domain/
    ietf-yang-mpl-statistics:mpl-statistics
    /nr-of-consistent-data
1004157 data    /ietf-yang-mpl-domain:domain/
    ietf-yang-mpl-statistics:mpl-statistics
    /nr-of-copies-forwarded
1004158 data    /ietf-yang-mpl-domain:domain/
    ietf-yang-mpl-statistics:mpl-statistics
    /nr-of-copies-received
1004159 data    /ietf-yang-mpl-domain:domain/
    ietf-yang-mpl-statistics:mpl-statistics
    /nr-of-inconsistent-control
1004160 data    /ietf-yang-mpl-domain:domain/
    ietf-yang-mpl-statistics:mpl-statistics
    /nr-of-inconsistent-data
1004161 data    /ietf-yang-mpl-domain:domain/
    ietf-yang-mpl-statistics:mpl-statistics
    /nr-of-messages-forwarded
1004162 data    /ietf-yang-mpl-domain:domain/
    ietf-yang-mpl-statistics:mpl-statistics
    /nr-of-messages-received
1004163 data    /ietf-yang-mpl-domain:domain/
    ietf-yang-mpl-statistics:mpl-statistics
    /nr-of-missed-messages
1004164 data    /ietf-yang-mpl-domain:domain/
    ietf-yang-mpl-statistics:mpl-statistics
    /nr-of-notreceived-messages
1004165 data    /ietf-yang-mpl-domain:domain/
    ietf-yang-mpl-statistics:mpl-statistics
    /nr-of-refused-copies
1004166 data    /ietf-yang-mpl-domain:domain/
    ietf-yang-mpl-statistics:mpl-statistics
    /seedID
1004167 data    /ietf-yang-mpl-domain:domain/
    ietf-yang-mpl-statistics:mpl-statistics
    /statistics-interval
```

#### 4. yang-mpl modules

This section describes the four yang modules. The modules are based on the MPL specification published in [RFC7731] and the specification of [RFC6206]. The identification of the interfaces follows the specification of ietf-interfaces of [RFC8343].

The data model allows to set values to the parameters of the MPL algorithm. This approach requires an active manager process to set the values without use of DHCP as described in: [RFC7774].

The names of the four modules are: yang-mpl-domain, yang-mpl-ops, yang-mpl-seeds, and yang-mpl-statistics, described in subsections with the same name.

##### 4.1. yang-mpl-domain module

This module describes (1) the MPL domains and the associated multicast addresses, and (2) the interfaces and the multicast addresses for which they are enabled.

The model features a choice such that:

the model specifies for constrained devices with only one "single" interface and only one "single" domain, a list of MC addresses for which the single interface is enabled.

the model specifies for larger devices "multiple" interface and "multiple" domains, a list of MC addresses for which one or more interfaces are enabled.

```
<CODE BEGINS>file "ietf-yang-mpl-domain@2018-07-06.yang"
```

```
module ietf-yang-mpl-domain {  
    yang-version 1.1;  
  
    namespace  
        "urn:ietf:params:xml:ns:yang:ietf-yang-mpl-domain";  
  
    prefix mpl;  
  
    import ietf-inet-types{  
        prefix inet;  
    }  
}
```

## organization

"IETF ROLL (Routing Over Low power and lossy networks)  
Working Group";

## contact

"WG Web: <http://tools.ietf.org/wg/roll/>  
WG List: <mailto:roll@ietf.org>

WG Chair: Peter van der Stok  
<mailto:consultancy@vanderstok.org>

WG Chair: Ines Robles  
<mailto:maria.ines.robles@ericsson.com>

Editor: Peter van der Stok  
<mailto:consultancy@vanderstok.org>";

## description

"This module contains information about the state of the MPL domain.

Copyright (c) 2018 IETF Trust and the persons identified as  
authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or  
without modification, is permitted pursuant to, and subject  
to the license terms contained in, the Simplified BSD License  
set forth in Section 4.c of the IETF Trust's Legal Provisions  
Relating to IETF Documents  
(<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see  
the RFC itself for full legal notices.";

```
revision "2018-07-06" {  
  description "revision 3";  
  reference  
    "I-D:draft-ietf-roll-mpl-yang: A YANG model for Multicast  
    Protocol for Low power and lossy Networks (MPL)";  
}  
container domain {  
  description  
    "High level container containing the choice statement  
    between single domain/interface and multiple  
    domains and interfaces.";  
  
  choice multiple {  
    description  
      "A choice for large devices with multiple domains
```

```
    and interfaces.";
  container mpl-domain {
    description
      "The entries describe the MPL domains, the associated
      Multicast addresses and interfaces.";

    list domains {
      key domainID;
      description
        "The entries describe a given domain identified with
        domainID and the associated Multicast addresses.";

      leaf domainID {
        type uint16;
        description
          "Entry uniquely identifies the domain in the
          forwarder.";
      }

      leaf-list Mclist{
        type inet:ipv6-address;
        description
          "List of associated IPv6 Addresses.";
      }
    } // domains list

    list addresses {
      key MCaddress;
      description
        "The entries describe the interfaces enabled
        with the specified MC address.";

      leaf MCaddress {
        type inet:ipv6-address;
        description
          "MC address belonging to a MPL domain.";
      }

      leaf-list interfaces {
        type string;
        description
          "List of names of interfaces enabled for this
          Multicast address. Interface name is defined
          in Appendix A of [RFC8343].";
      }
    } // addresses list
  } // container mpl-domain
  container mpl-single {
```

```

    description
        "A choice for constrained devices with a list of
        MC addresses for single interface and domain.";
    leaf-list MCaddresses{
        type inet:ipv6-address;
        description
            "list of MC addresses belonging to one single
            domain and interface.";
    }
} // container mpl-simple
} // choice simple
} // container module
} //module ietf-yang-mpl-domain

```

<CODE ENDS>

#### 4.2. yang-mpl-ops module

This module models the operational aspects of MPL. Per domain MPL specifies four parameters I\_MAX, I\_MIN, K, and TIMER\_EXPIRATIONS for data and control messages. The value of the MPL intervals are expressed in TUNIT. The entry SE\_LIFETIME taken over from [RFC7774] fixes TUNIT to milliseconds. For very constrained devices with only one domain there can be only one instance of mpl-parameter list. The module augments the ietf-yang-mpl-domain module.

```

<CODE BEGINS>file "ietf-yang-mpl-ops@2018-07-06.yang"

module ietf-yang-mpl-ops {

    yang-version 1.1;

    namespace
        "urn:ietf:params:xml:ns:yang:ietf-yang-mpl-ops";
    prefix "mplo";

    import ietf-yang-mpl-domain{
        prefix "mpl";
    }

    organization
        "IETF ROLL (Routing over Low power and lossy networks)
        Working Group";

    contact

```

"WG Web: <http://tools.ietf.org/wg/roll/>  
WG List: <mailto:roll@ietf.org>  
  
WG Chair: Peter van der Stok  
<mailto:consultancy@vanderstok.org>  
  
WG Chair: Ines Robles  
<mailto:maria.ines.robles@ericsson.com>  
  
Editor: Peter van der Stok  
<mailto:consultancy@vanderstok.org>";

description

"This module contains information about the operation of the MPL protocol.

Copyright (c) 2018 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices."

```
revision "2018-07-06" {
  description "revision 3";
  reference
    "I-D:draft-ietf-roll-mpl-yang: A YANG model for Multicast
    Protocol for Low power and lossy Networks (MPL)";
}

augment "/mpl:domain" {
  description
    "additional MPL server settings to MPL domains";

  container mpl-ops {
    description
      "Parameter settings for each MPL server and for each
      individual domain of the server.";

    leaf PROACTIVE_FORWARDING {
      type boolean;
    }
  }
}
```

```
    description
      "The boolean value indicates whether the MPL forwarder
       schedules MPL data message transmission after
       receiving them for the first time.
       Specified in section 5.4 of [RFC7731]";
  }

  leaf SEED_SET_ENTRY_LIFETIME {
    type uint64;
    units "seconds";
    default 1800;
    description
      "The value indicates the minimum lifetime for
       an entry in the Seed set expressed in seconds.
       Default value is 30 minutes.
       Specified in section 5.4 of [RFC7731]";
  }

  list mpl-parameter{
    key domainID;
    description
      "Each domain has a set of MPL forwarding parameters
       which regulate the forwarding operation.";

    leaf domainID{
      type uint16;
      description
        "Each domainID must be present in
         mpl-parameter list.";
    }

    leaf DATA_MESSAGE_IMIN{
      type uint16;
      description
        "The minimum Trickle timer interval
         for MPL Data Message transmissions.
         mpl-ops/mpl-parameter/DATA_MESSAGE_IMIN <=
         mpl-seeds/buffered-messages/I <=
         mpl-ops/mpl-parameter/DATA_MESSAGE_IMAX
         Specified in section 5.4 of [RFC7731]";
    }

    leaf DATA_MESSAGE_IMAX{
      type uint16;
      description
        "The maximum Trickle timer interval
         for MPL Data Message transmissions.
         mpl-ops/mpl-parameter/DATA_MESSAGE_IMIN <=
```



```
        mpl-seeds/buffered-messages/I <=
        mpl-ops/mpl-parameter/DATA_MESSAGE_IMAX
        Specified in section 5.4 of [RFC7731]";
    }

    leaf DATA_MESSAGE_K{
        type uint16;
        default 1;
        description
            "The redundancy constant
             for MPL Data Message transmissions.
             Specified in section 5.4 of [RFC7731]";
    }

    leaf DATA_MESSAGE_TIMER_EXPIRATIONS{
        type uint16;
        default 3;
        description
            "The number of Trickle timer expirations, as
             that occur before terminating the Trickle
             algorithm's retransmission of a given
             MPL Data Message.
             Specified in section 5.4 of [RFC7731]";
    }

    leaf CONTROL_MESSAGE_IMIN{
        type uint16;
        description
            "The minimum Trickle timer interval
             for MPL Control Message transmissions.
             Specified in section 5.4 of [RFC7731]";
    }

    leaf CONTROL_MESSAGE_IMAX{
        type uint16;
        description
            "The maximum Trickle timer interval
             for MPL Control Message transmissions.
             Specified in section 5.4 of [RFC7731]";
    }

    leaf CONTROL_MESSAGE_K{
        type uint16;
        default 1;
        description
            "The redundancy constant
             for MPL Control Message transmissions.
             Specified in section 5.4 of [RFC7731]";
    }
```

```

    }

    leaf CONTROL_MESSAGE_TIMER_EXPIRATIONS{
        type uint16;
        default 10;
        description
            "The number of Trickle time expirations,
             that occur before terminating the Trickle
             algorithm for MPL Control Message
             transmissions.
             Specified in section 5.4 of [RFC7731]";
    }

    } // list MPL-parameter
  } // container MPL-ops
} // augment ietf-yang-mpl-domain
} // module ietf-yang-mpl-ops

<CODE ENDS>

```

#### 4.3. yang-mpl-seeds module

This module specifies the current values of the operation of the MPL forwarder. The values are acquired by the client and set by the server. The module specifies a set of message buffers, with a buffer per seed and domain. In constrained devices there will be only one domain, but probably multiple seeds.

The message buffer contains a set of messages where each message is uniquely identified by its sequence number and seed. The associated I, c, e, and t values indicate the progress of MPL with respect to this message, as specified in [RFC7731]. A forwarder sends and receives multiple copies of a message. When a forwarder has sent (received) a copy of a message, the forwarder has sent (received) that message.

For forwarders which are seeds, local has value true and seqno is the sequence number of the next message to send.

The module augments the ietf-yang-mpl-domain module.

```

<CODE BEGINS>file "ietf-yang-mpl-seeds@2018-07-06.yang"

module ietf-yang-mpl-seeds {

    yang-version 1.1;

```

```
namespace
  "urn:ietf:params:xml:ns:yang:ietf-yang-mpl-seeds";

prefix "mplse";

import ietf-yang-mpl-domain{
  prefix "mpl";
}

organization
  "IETF ROLL (Routing over Low power and lossy networks)
  Working Group";

contact
  "WG Web:   http://tools.ietf.org/wg/roll/
  WG List:   mailto:roll@ietf.org

  WG Chair: Peter van der Stok
             mailto:consultancy@vanderstok.org

  WG Chair: Ines Robles
             mailto:maria.ines.robles@ericsson.com

  Editor:    Peter van der Stok
             mailto:consultancy@vanderstok.org";

description
  "This module contains information about the operation
  of the MPL protocol.

  Copyright (c) 2018 IETF Trust and the persons identified as
  authors of the code. All rights reserved.

  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject
  to the license terms contained in, the Simplified BSD License
  set forth in Section 4.c of the IETF Trust's Legal Provisions
  Relating to IETF Documents
  (http://trustee.ietf.org/license-info).

  This version of this YANG module is part of RFC XXXX; see
  the RFC itself for full legal notices.";

revision "2018-07-06" {
  description "revision 3";
  reference
    "I-D:draft-ietf-roll-mpl-yang: A YANG model for Multicast
```

```
Protocol for Low power and lossy Networks (MPL)";
}

augment "/mpl:domain" {
    description "additional message buffer status to MPL
domains";

    leaf SE_LIFETIME {
        type uint16;
        units "milliseconds/timer units";
        description
            "Conversion from clock ticks to milliseconds,
            equivalent to SEED_SET_ENTRY_LIFETIME/TUNIT as
            specified in [RFC7774].";
    }

    list mpl-seeds{
        key "seedID domainID";
        config false;
        description
            "List describes all seeds that are active in the
            server. Seed information contains the message buffer
            contents and the operational values of I, c, sequence
            number and the life-times per message.";

        leaf seedID{
            type uint64;
            description
                "value uniquely identifies the MPL Seed within a MPL
                domain.";
        }

        leaf domainID{
            type uint16;
            description
                "together with seedID uniquely identifies buffer
                set.";
        }
    }

    leaf local {
        type boolean;
        description
            "When local == true, seed originated in this
            forwarder.
            When local == false, seed originated in different
            forwarder.";
    }
}
```

```
leaf generate-seqno {
    type uint8;
    description
        "Sequence number of next message to be generated by
        this local seed.";
}

leaf life-time {
    type uint64;
    units    "milliseconds";
    description
        "Minimum remaining lifetime of the seed entry in
        milliseconds.";
}

leaf min-seqno{
    type uint8;
    description
        "Lower bound sequence number in the buffer of the
        seed.";
}

leaf data-number{
    type uint8;
    description
        "Number of currently buffered data messages.";
}

leaf control-number{
    type uint8;
    description
        "Number of currently buffered control messages.";
}

list buffered-messages{
    key seqno;
    description
        "status of trickle intervals of the buffered message
        identified by seqno. and seed/domain";

    leaf seqno{
        type uint8;
        description
            "Sequence number of message.";
    }

    leaf I{
        type uint8;
    }
}
```

```
        units "milliseconds";
        description
            "Current Trickle timer interval size in
            milliseconds.
            mpl-ops/mpl-parameter/DATA_MESSAGE_IMIN <=
            mpl-seeds/buffered-messages/I <=
            mpl-ops/mpl-parameter/DATA_MESSAGE_IMAX";
    }

    leaf c{
        type uint8;
        description
            "number of times that a copy of this message has
            been received in this interval.";
    }

    leaf e{
        type uint8;
        description
            "number of Trickle time expirations since last
            Trickle timer reset.";
    }

    leaf t{
        type uint8;
        units "milliseconds";
        description
            "Time expressed in milliseconds
            that message will be (is) forwarded.
            mpl-seeds/buffered-messages/t <
            mpl-seeds/buffered-messages/I";
    }

    } // list seed-timers
    } // list MPL-seeds
    } // augment
} // module ietf-yang-mpl-seeds
```

<CODE ENDS>

#### 4.4. yang-mpl-statistics module

This module specifies the operation of the MPL forwarder expressed in number of messages and copies. The values are acquired by the client and set by the server. Statistics are specified per seed and domain.

In constrained devices there will be only one domain, but probably multiple seeds.

The parameter *k* determines how many copies of a message can be forwarded. The counters *c-too-high*, *nr-forwarded*, and *nr-not-forwarded* give insight in the consequences of the current value of *k*.

The other counters give insight in the loss of messages caused by the medium or forwarding delays. The inconsistent/consistent counters indicate when consistent or inconsistent messages were received according to the definition of consistent in [RFC7731].

The module augments the `ietf-yang-mpl-domain` module.

```
<CODE BEGINS>file "ietf-yang-mpl-statistics@2018-07-06.yang"
```

```
module ietf-yang-mpl-statistics {  
    yang-version 1.1;  
  
    namespace  
        "urn:ietf:params:xml:ns:yang:ietf-yang-mpl-statistics";  
  
    prefix "mplst";  
  
    import ietf-yang-mpl-domain {  
        prefix "mpl";  
    }  
  
    organization  
        "IETF ROLL (Routing over Low power and lossy networks)  
        Working Group";  
  
    contact  
        "WG Web:  http://tools.ietf.org/wg/roll/  
        WG List:  mailto:roll@ietf.org  
  
        WG Chair: Peter van der Stok  
                  mailto:consultancy@vanderstok.org  
  
        WG Chair: Ines Robles  
                  mailto:maria.ines.robles@ericsson.com  
  
        Editor:   Peter van der Stok  
                  mailto:consultancy@vanderstok.org";
```

```
description
  "This module contains information about the operation
  of the MPL protocol.

  Copyright (c) 2018 IETF Trust and the persons identified as
  authors of the code. All rights reserved.

  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject
  to the license terms contained in, the Simplified BSD License
  set forth in Section 4.c of the IETF Trust's Legal Provisions
  Relating to IETF Documents
  (http://trustee.ietf.org/license-info).

  This version of this YANG module is part of RFC XXXX; see
  the RFC itself for full legal notices."

revision "2018-07-06" {
  description "revision 3";
  reference
    "I-D:draft-ietf-roll-mpl-yang: A YANG model for Multicast
    Protocol for Low power and lossy Networks (MPL)";
}

augment "/mpl:domain" {
  description "additional MPL message statistics to MPL
  domains";

  list mpl-statistics{
    key "seedID domainID";

    config false;

    description
      "List describes performance statistics integrated over
      the messages identified by seed and domain identifiers.
      A forwarder can receive and forward multiple copies of
      a message uniquely identified by seqno, domain, and
      seed.";

    leaf seedID{
      type uint64;
      description
        "value uniquely identifies the MPL Seed within a MPL
        domain.";
    }
  }
}
```



```
leaf domainID{
    type uint16;
    description
        "together with seed-ID uniquely identifies buffer
        set.";
}

leaf c-too-high {
    type uint64;
    description
        "Number of times that a copy was not forwarded
        because c > k.";
}

leaf nr-forwarded {
    type uint64;
    description
        "number of times copies are forwarded,
        while c <= k.";
}

leaf nr-of-messages-received{
    type uint64;
    description
        "number of messages received,
        must be smaller than or equal to seqno.";
}

    leaf nr-of-messages-forwarded{
        type uint64;
        description
            "number of forwarded messages, must be smaller
            than or equal to nr-of-messages-received.";
    }

leaf nr-of-copies-received{
    type uint64;
    description
        "total number of message copies received.";
}

leaf nr-of-copies-forwarded{
    type uint64;
    description
        "number of forwarded copies, can be larger than
        number-of-copies-received.";
}
```

```
leaf nr-of-refused-copies{
  type uint64;
  description
    "number of refused copies because seqno too small.";
}

leaf nr-of-missed-messages{
  type uint64;
  description
    "number of messages that were not received
    because seqno < min-seqno.";
}

leaf nr-of-notreceived-messages{
  type uint64;
  description
    "number of messages that were not received
    according to control message.";
}

leaf nr-of-inconsistent-data{
  type uint64;
  description
    "number of inconsistent data messages.";
}

leaf nr-of-consistent-data{
  type uint64;
  description
    "number of consistent data messages.";
}

leaf nr-of-consistent-control{
  type uint64;
  description
    "number of consistent control messages.";
}

leaf nr-of-inconsistent-control{
  type uint64;
  description
    "number of inconsistent control messages.";
}

leaf statistics-interval{
  type uint64;
  description
    "Interval, expressed in seconds, during which
```

```
        the statistics are collected.";
    }

    action reset-buffer-statistics{
        description
            "set all statistics counters and
             statistics-interval of buffer[seedID domainID]
             to zero.";
    }

} // list mpl statistics

action reset-all-statistics{
    description
        "set all statistics counters and
         statistics-interval of all buffers to zero.";
}

} // augment
} // module ietf-yang-mpl-statistics

<CODE ENDS>
```

## 5. IANA Considerations

Registration of four YANG modules and corresponding SID files in the "YANG module assignment" registry is required.

module: ietf-yang-mpl-domain

.yang file: ietf-yang-mpl-domain@2018-07-06.yang

.sid file: ietf-yang-mpl-domain@2018-07-06.sid

module: ietf-yang-mpl-ops

.yang file: ietf-yang-mpl-ops@2018-07-06.yang

.sid file: ietf-yang-mpl-ops@2018-07-06.sid

module: ietf-yang-mpl-seeds

.yang file: ietf-yang-mpl-seeds@2018-07-06.yang  
.sid file: ietf-yang-mpl-seeds@2018-07-06.sid  
module: ietf-yang-mpl-statistics

.yang file: ietf-yang-mpl-statistics@2018-07-06.yang  
.sid file: ietf-yang-mpl-statistics@2018-07-06.sid

## 6. Acknowledgements

Andy Bierman has commented on the use of YANG for mpl. Many thanks to Radi Krejci for yang review.

## 7. Changelog

Changes from version 01 to version 02

- o Added NMDA section.
- o added module explanation in model section
- o IANA considerations added

Changes from version 00 to version 01

- o config false in "statistics" and "seeds" modules
- o separated into 4 modules
- o inserted choice in domain modules
- o more explanatory text
- o renamed some parameters
- o Introduced section per module
- o reset of statistics is added

Version ietf-00 copied from version vanderstok-02

## 8. References

### 8.1. Normative References

- [I-D.ietf-core-sid] Veillette, M. and A. Pelov, "YANG Schema Item iDentifier (SID)", draft-ietf-core-sid-04 (work in progress), June 2018.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC7731] Hui, J. and R. Kelsey, "Multicast Protocol for Low-Power and Lossy Networks (MPL)", RFC 7731, DOI 10.17487/RFC7731, February 2016, <<https://www.rfc-editor.org/info/rfc7731>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.

### 8.2. Informative References

- [I-D.ietf-netmod-yang-tree-diagrams] Bjorklund, M. and L. Berger, "YANG Tree Diagrams", draft-ietf-netmod-yang-tree-diagrams-06 (work in progress), February 2018.
- [RFC6206] Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko, "The Trickle Algorithm", RFC 6206, DOI 10.17487/RFC6206, March 2011, <<https://www.rfc-editor.org/info/rfc6206>>.
- [RFC7390] Rahman, A., Ed. and E. Dijk, Ed., "Group Communication for the Constrained Application Protocol (CoAP)", RFC 7390, DOI 10.17487/RFC7390, October 2014, <<https://www.rfc-editor.org/info/rfc7390>>.

- [RFC7774] Doi, Y. and M. Gillmore, "Multicast Protocol for Low-Power and Lossy Networks (MPL) Parameter Configuration Option for DHCPv6", RFC 7774, DOI 10.17487/RFC7774, March 2016, <<https://www.rfc-editor.org/info/rfc7774>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.

## Author's Address

Peter van der Stok (editor)  
consultant

Phone: +31-492474673 (Netherlands), +33-966015248 (France)  
Email: [consultancy@vanderstok.org](mailto:consultancy@vanderstok.org)  
URI: [www.vanderstok.org](http://www.vanderstok.org)

ROLL  
Internet-Draft  
Intended status: Standards Track  
Expires: September 22, 2018

C. Ji, Ed.  
R. Koutsiamanis  
G. Papadopoulos  
IMT Atlantique  
D. Dujovne  
Universidad Diego Portales  
N. Montavont  
IMT Atlantique  
March 21, 2018

Traffic-aware Objective Function  
draft-ji-roll-traffic-aware-objective-function-01

Abstract

This document proposes a packet transmission rate metric for parent selection. This metric represents the amount of traffic that the node is transmitting to the current parent node. This document also proposes an Objective Function (OF) using the packet transmission rate metric for parent selection in order to balance the amount of traffic between nodes.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 22, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	3
3. DODAG construction in RPL . . . . .	3
4. Load distribution problem in RPL . . . . .	3
5. TAOF description . . . . .	5
6. DIO Metric Container Type extension . . . . .	6
7. Security Considerations . . . . .	8
8. IANA Considerations . . . . .	8
9. Informative references . . . . .	8
Authors' Addresses . . . . .	9

## 1. Introduction

RPL [RFC6550] is an IPv6 Routing protocol for LLNs. It uses Objective Functions (OF) to construct the Destination Oriented Directed Acyclic Graph (DODAG) containing the nodes of the network. The existing OFs defined are OF Zero (OF0) [RFC6552] and Minimum Rank with Hysteresis OF (MRHOF) [RFC6719]. These OFs specify how nodes in a DODAG select their preferred parent using different metrics.

The metrics can be separated into two different types, link metrics (e.g. ETX) and node metrics (e.g. energy). Experimental results [I-D.qasem-roll-rpl-load-balancing] conclude that using the current OFs leads to an unbalanced network within which some of the nodes are overloaded. In this case, a node is overloaded in the sense that it forwards much more packets than it otherwise would if the network were balanced. This problem has consequences for the lifetime of the network because overloaded nodes tend to drain quicker than others, a problem which becomes even more significant when the overloaded nodes are near the DODAG root [I-D.qasem-roll-rpl-load-balancing].

This problem is still an open issue and this draft proposes a new way of parent selection as an attempt towards a solution. This draft proposes a new OF that considers the packet transmission rate as a representation of traffic each node faces and use this information to balance the amount of traffic between nodes.

In brief, each node tracks its packet transmission rate and appends this information to DIO messages it sends as a DAG Metric Container



option. When the DIO message is received by child nodes or potential child nodes, the packet transmission rate information is stored and used to influence the result when RPL parent selection is performed.

## 2. Terminology

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

## 3. DODAG construction in RPL

RPL uses OFs to construct a DODAG. OFs define the way the nodes select their preferred parent and how they compute the new rank. A node's rank is always larger than its parent's rank because the calculation of rank is based on an increment to the parent's rank. This increment differs for each OF but all include the MinHopRankIncrease which is the minimum increase in rank between a node and a node's parent and a step. Different OFs use different metrics or constraints to select the preferred parent and to define the step, depending on application requirements. Nodes obtain these values from DODAG Information Object (DIO) control messages sent by their neighbor nodes.

The construction of a DODAG starts when the root node sends DIO messages to its neighbors. After receiving the DIO, these neighbor nodes select the root as their preferred parent if they wish to join the DODAG. In order to announce that they joined the DODAG as its child node, they send a Destination Advertisement Object (DAO) to their preferred parent - the DODAG root. After joining the DODAG, these nodes send their own DIO messages with the new computed rank to their neighbors. This procedure repeats for every node which joins the DODAG.

## 4. Load distribution problem in RPL

Numerous experiments using existing OFs have been conducted and according to results, RPL faces a load distribution problem in large LLNs. With RPL using existing OFs, such as MRHOF, an unbalanced network is formed with some of the nodes overloaded and other nodes at rest. This problem is severe for network performance because overloaded nodes will use up their available energy faster than other nodes. This is exacerbated for nodes near the root (within 1 hop distance) or nodes which are the only parent candidate for some other nodes. Additionally, when the overloaded node shuts down, a big part of the network will become disconnected and will have to be transferred to another parent. There is a high probability that the children nodes will also select the same new node as their parent,

leading to another overloaded node. Also, when a node has selected its parent, it will change only when the parent node is not reachable (due to battery depletion or packet losses).

The existing OFs usually use a single metric to compare parent candidates, for example, as described in [RFC6719] the default metric used in MRHOF is ETX [RFC6551], which represents the number of transmissions a node expects to make to a destination in order to successfully deliver one packet. The result from using a single metric is that nodes prefer to select the same node as their parent, which according to [I-D.qasem-roll-rpl-load-balancing] leads to an unbalanced network with overloaded nodes (node load is indicated by a node's child count). But the child count does not accurately indicate the load because among these child nodes, some of them may have higher traffic load and others may have lower.

The network traffic can be quantified by tracking the packets a node generates/sends/receives and the amount of energy it consumes. Energy consumption is strongly correlated to the amount of network traffic handled by a node since the energy consumption for the operation of the radio is the primary energy consumer in typical nodes. However, directly measuring the packet transmission rate is both more accurate and also works when nodes have atypical energy consumption profiles (e.g. increased node processing or high energy consumption sensors).

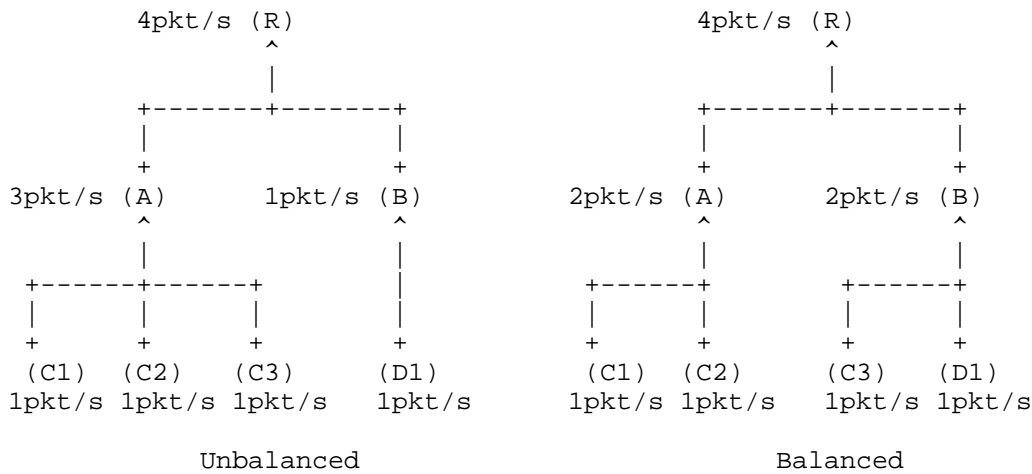


Figure 1: Packet Transmission Rates of nodes with the same requirements

As a first simple example, an unbalanced network with nodes which all have the same packet transmission rates is shown in Figure 1. Its

transformation into a balanced equivalent network is shown on the right.

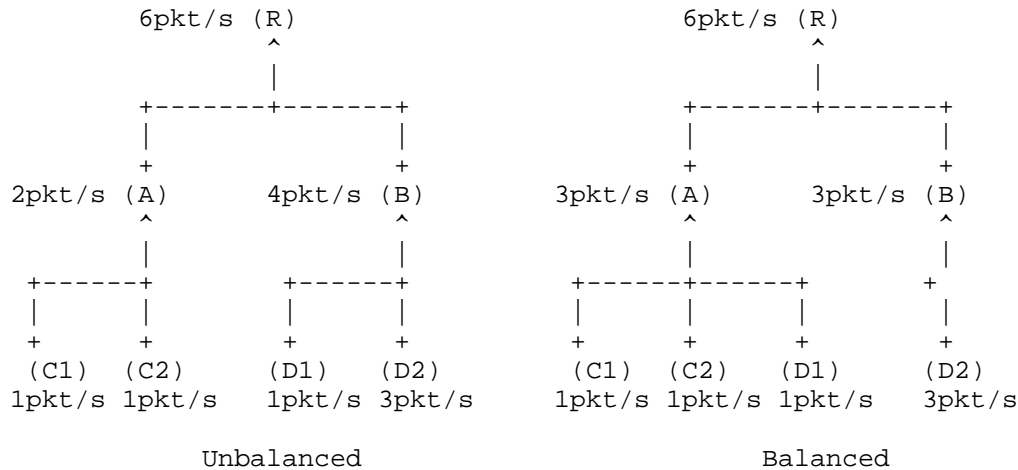


Figure 2: Packet Transmission Rates of nodes with different requirements

As a second simple example, an unbalanced network with nodes which have different packet transmission rates is shown in Figure 2. Its transformation into a balanced equivalent network is shown on the right.

## 5. TAOF description

In this specification, a metric is proposed to be used in the parent selection mechanism, the Packet Transmission Rate (PTR) which represents the number of packets each node transmitted (sent or forwarded) during a certain time period. As mentioned below, the number of transmitted packets can directly show the amount of traffic each node is facing. This information is added in DIO messages and is broadcast to every neighbor.

At first, each node MUST identify from their neighbor set which nodes are acceptable to be selected as a parent. For this purpose, the metric ETX is used as a filter to filter out parent candidates with low link quality with a preference for nodes with link quality below a given threshold. The ETX threshold SHOULD be different depending on application requirements. The suggested value for the relevant threshold MAX\_PATH\_COST from MRHOF [RFC6719] is 32768, which means the specific path has expected transmission counts greater than 256.

For the packet transmission rate, each node maintains in a variable a counter which will increment by 1 every time a data packet is transmitted by the node. When the ETX value is used as a filter, nodes with bad link quality will not be included in the parent set. This ensures that undue retransmissions caused by bad link will be avoided. In any case, the node chooses the parent candidate with the least packet transmission rate.

This proposal is expected to increase the frequency of parent change because the packet transmission rate is more likely to be different between DIO messages, even for DIO messages from the same node. There are multiple ways to minimize the frequency of unnecessary parent changes:

- a. Use the packet transmission rate in combination with another metric (e.g. child count, hop counts).
- b. Use a threshold when comparing the packet transmission rate, similar to the approach in MRHOF [RFC6719]. Switch parents when the difference of packet transmission rate between the original parent and the alternative parent is above a threshold. This threshold depends on different factors (e.g. network size, average traffic load) and SHOULD be defined differently for each use case.

#### 6. DIO Metric Container Type extension

```

0                                     1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-----+-----+-----+-----+
| Packet Transmission Rate (PTR) |
+-----+-----+-----+-----+

```

Figure 3: DAG metric container type format.

A DIO message carries fields as described in RFC6550 [RFC6550] and the available options for the DAG metric container are described in RFC6551 [RFC6551]. In this specification, a metric container option is proposed and the detailed format is shown in Figure 3. The information carried is the PTR, represented as a 2 byte unsigned integer.

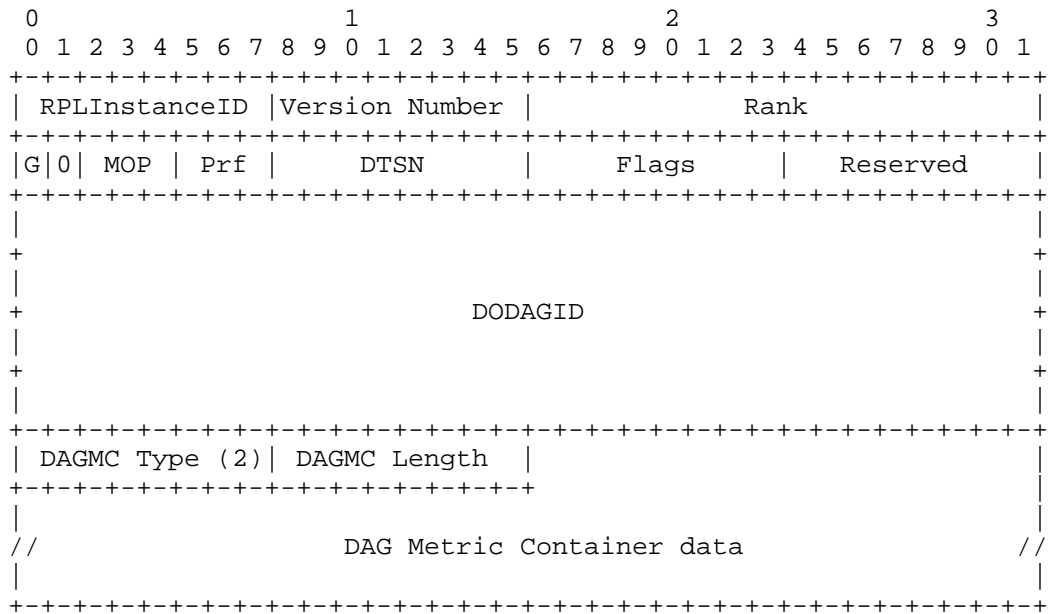


Figure 4: Example DIO Message with a DAG Metric Container option

The structure of the DIO Control Message when a DAG Metric Container option is included is shown in Figure 4. The DAG Metric Container option type (DAGMC Type in Figure 4) has the value 0x02 as per the IANA registry for the RPL Control Message Options, and is defined in [RFC6550]. The DAG Metric Container option length (DAGMC Length in Figure 4) expresses the the DAG Metric Container length in bytes. DAG Metric Container data holds the actual data and is shown further expanded in Figure 5.

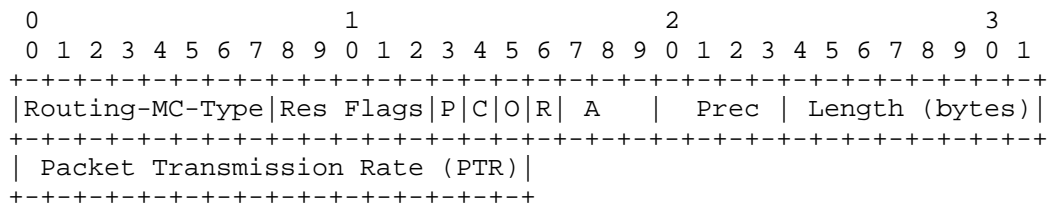


Figure 5: DAG Metric Container (MC) data with Packet Transmission Rate (PTR) object body

An example DAG Metric Container containing the proposed Metric Container object is shown in Figure 5. The explicit definition of the fields is:

Routing-MC-Type: TBD1. The type of the proposed DAGMC extension. To be assigned by IANA.

Packet Transmission Rate (PTR): The packet transmission rate, represented as a 2 byte unsigned integer.

## 7. Security Considerations

The structure of the DIO control message is extended, within the predefined DIO options. Therefore, the security mechanisms defined in RPL [RFC6550] apply to this proposed extension.

## 8. IANA Considerations

This proposal requests the allocation of a new value TBD1 for the metric type "PTR" in the Routing-MC-Type field in the DAG MC from IANA.

## 9. Informative references

- [I-D.qasem-roll-rpl-load-balancing]  
Qasem, M., Al-Dubai, A., Romdhani, I., Ghaleb, B., Hou, J., and R. Jadhav, "Load Balancing Objective Function in RPL", draft-qasem-roll-rpl-load-balancing-02 (work in progress), October 2017.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <<https://www.rfc-editor.org/info/rfc6550>>.
- [RFC6551] Vasseur, JP., Ed., Kim, M., Ed., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks", RFC 6551, DOI 10.17487/RFC6551, March 2012, <<https://www.rfc-editor.org/info/rfc6551>>.

- [RFC6552] Thubert, P., Ed., "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6552, DOI 10.17487/RFC6552, March 2012, <<https://www.rfc-editor.org/info/rfc6552>>.
- [RFC6719] Gnawali, O. and P. Levis, "The Minimum Rank with Hysteresis Objective Function", RFC 6719, DOI 10.17487/RFC6719, September 2012, <<https://www.rfc-editor.org/info/rfc6719>>.

## Authors' Addresses

Chenyang Ji (editor)  
IMT Atlantique  
Office D00 - 116A  
2 Rue de la Chataigneraie  
Cesson-Sevigne - Rennes 35510  
FRANCE

Email: [chenyang.ji@imt-atlantique.net](mailto:chenyang.ji@imt-atlantique.net)

Remous-Aris Koutsiamanis  
IMT Atlantique  
Office B00 - 126A  
2 Rue de la Chataigneraie  
Cesson-Sevigne - Rennes 35510  
FRANCE

Phone: +33 299 12 70 49  
Email: [aris@ariskou.com](mailto:aris@ariskou.com)

Georgios Papadopoulos  
IMT Atlantique  
Office B00 - 114A  
2 Rue de la Chataigneraie  
Cesson-Sevigne - Rennes 35510  
FRANCE

Phone: +33 299 12 70 04  
Email: [georgios.papadopoulos@imt-atlantique.fr](mailto:georgios.papadopoulos@imt-atlantique.fr)

Diego Dujovne  
Universidad Diego Portales  
Escuela de Informatica y Telecomunicaciones, Av. Ejercito 441  
Santiago, Region Metropolitana  
Chile

Phone: +56 (2) 676-8121  
Email: [diego.dujovne@mail.udp.cl](mailto:diego.dujovne@mail.udp.cl)

Nicolas Montavont  
IMT Atlantique  
Office B00 - 106A  
2 Rue de la Chataigneraie  
Cesson-Sevigne - Rennes 35510  
FRANCE

Phone: +33 299 12 70 23  
Email: [nicolas.montavont@imt-atlantique.fr](mailto:nicolas.montavont@imt-atlantique.fr)



ROLL  
Internet-Draft  
Intended status: Standards Track  
Expires: April 22, 2019

C. Ji, Ed.  
Alexander TEI of Thessaloniki  
R. Koutsiamanis  
G. Papadopoulos  
IMT Atlantique  
D. Dujovne  
Universidad Diego Portales  
N. Montavont  
IMT Atlantique  
October 19, 2018

Traffic-aware Objective Function  
draft-ji-roll-traffic-aware-objective-function-03

Abstract

This document proposes a remaining throughput metric for parent and DODAG selection. This metric represents the amount of remaining traffic handling capacity that the node has. This document also proposes an Objective Function (OF) which uses the proposed metric for parent and DODAG selection to balance the amount of traffic between nodes and DODAGs.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://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 April 22, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	3
3. DODAG construction in RPL . . . . .	3
4. Load distribution problem in RPL . . . . .	3
4.1. Parent selection problem . . . . .	4
4.2. DODAG selection problem . . . . .	6
5. TAOF description . . . . .	8
6. DIO Metric Container Type extension . . . . .	9
7. Enrollment . . . . .	11
8. Security Considerations . . . . .	12
9. IANA Considerations . . . . .	12
10. Informative references . . . . .	12
Authors' Addresses . . . . .	13

## 1. Introduction

RPL [RFC6550] is an IPv6 Routing protocol for LLNs. It uses Objective Functions (OF) to construct the Destination Oriented Directed Acyclic Graph (DODAG) containing the nodes of the network. The existing OFs defined are OF Zero (OF0) [RFC6552] and Minimum Rank with Hysteresis OF (MRHOF) [RFC6719]. These OFs specify how nodes in a DODAG select their preferred parent using different metrics.

The metrics can be separated into two different types, link metrics (e.g. ETX) and node metrics (e.g. energy). Experimental results [I-D.qasem-roll-rpl-load-balancing] conclude that using the current OFs leads to an unbalanced network within which some nodes are overloaded. Here, a node is overloaded in the sense that it forwards many more packets than it otherwise would if the network were balanced. This problem has consequences for the lifetime of the network because overloaded nodes drain quicker than others, a problem which becomes even more significant when the overloaded nodes are near the DODAG root [I-D.qasem-roll-rpl-load-balancing].

Similarly, one DODAG might be overloaded in the same sense compared to another DODAG, and this will lead to the same consequences for the whole DODAG as for a specific node.

This problem is still an open issue. This draft proposes a new way of parent and DODAG selection as an attempt towards a solution. This draft proposes a new OF that considers the remaining throughput as a representation of the remaining traffic handling capacity each node possesses and which uses this information to balance the amount of traffic between nodes and DODAGs.

In brief, each node tracks its remaining throughput and appends this information as a DAG Metric Container option to DIO messages it sends. When the DIO message is received by child nodes or potential child nodes, the remaining throughput information is stored and used to influence the result when RPL parent or DODAG selection is performed.

## 2. Terminology

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

## 3. DODAG construction in RPL

RPL uses OFs to construct a DODAG. OFs define the way the nodes select their preferred parent and DODAG and how they compute the new rank. A node's rank is always larger than its parent's rank because the calculation of rank is based on an increment to the parent's rank. This increment differs for each OF but all OFs include the MinHopRankIncrease, which is the minimum increase in rank between a node and a node's parent and a step. Different OFs use different metrics or constraints to select the preferred parent and DODAG and to define the step, depending on application requirements. Nodes obtain these values from DODAG Information Object (DIO) control messages sent by their neighbor nodes.

The construction of a DODAG starts when the root node sends DIO messages to its neighbors. After receiving the DIO, these neighbor nodes select the root as their preferred parent if they wish to join the DODAG. To announce that they joined the DODAG as its child node, they send a Destination Advertisement Object (DAO) to their preferred parent - the DODAG root. After joining the DODAG, these nodes send their own DIO messages with the new computed rank to their neighbors. This procedure repeats for every node which joins the DODAG.

## 4. Load distribution problem in RPL

According to the experiments conducted using existing OFs RPL faces a load distribution problem in large LLNs. With RPL using existing OFs, such as MRHOF, an unbalanced network is formed with some nodes

overloaded and other nodes at rest. This problem is severe for network performance because overloaded nodes will use up their available energy faster than other nodes. This is exacerbated for nodes near the root (within 1 hop distance) or nodes which are the only parent candidate for other nodes. Additionally, when the overloaded node shuts down, a big part of the network will become disconnected and will have to be transferred to another parent or DODAG. There is a high probability that the children nodes will also select the same new node as their parent or the same DODAG, leading to another overloaded node/DODAG. Also, when a node has selected its parent, it will change only when the parent node is not reachable (due to battery depletion or packet losses).

The existing OFs usually use a single metric to compare parent candidates, for example, as described in [RFC6719] the default metric used in MRHOF is ETX [RFC6551], which represents the number of transmissions a node expects to make to a destination to successfully deliver one packet. The result from using a single metric is that nodes prefer to select the same node as their parent, which according to [I-D.qasem-roll-rpl-load-balancing] leads to an unbalanced network with overloaded nodes (node load is indicated by a node's child count). But the child count does not accurately indicate the load because among the child nodes some may have higher traffic load and others may have lower.

The network traffic can be quantified by tracking the packets a node generates/sends/receives and the amount of energy it consumes. Energy consumption is strongly correlated to the amount of network traffic handled by a node since the energy consumption for the operation of the radio is the primary energy consumer in typical nodes. However, directly measuring the remaining throughput is both more accurate and also works when nodes have atypical energy consumption profiles (e.g. increased node processing or high energy consumption sensors).

Calculating the remaining throughput then requires knowledge of the total throughput supported by a node and subtraction of the used throughput.

#### 4.1. Parent selection problem

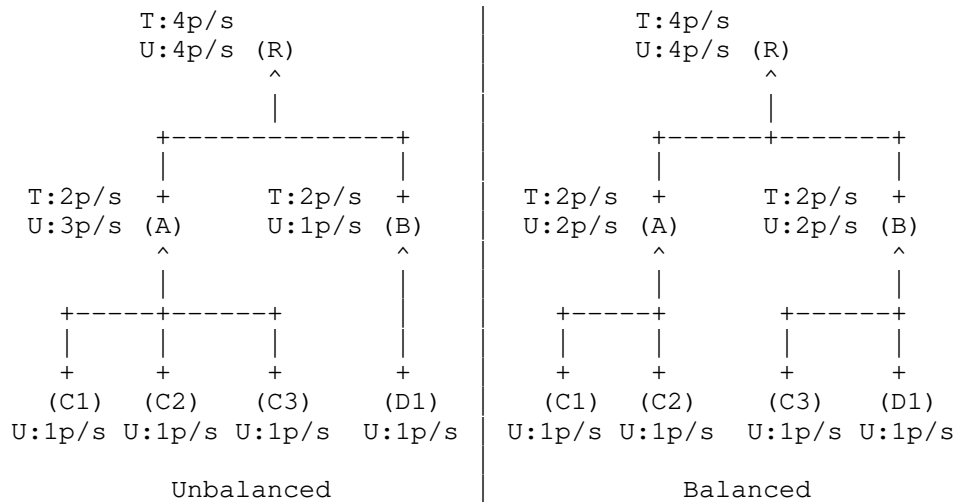


Figure 1: Use of Remaining Throughput with nodes with the same requirements

As a first simple example, an unbalanced network with nodes which all use the same throughput ("U:") is shown in Figure 1. Nodes A and B have the same total throughput ("T:"), but node A is overloaded due to trying to handle more than its ability while node B has a spare throughput of  $2-1=1p/s$ . Its transformation into a balanced network is shown on the right and it involves a node (C3) switching parents from A to B so that the capacity of its parent is no longer exceeded.

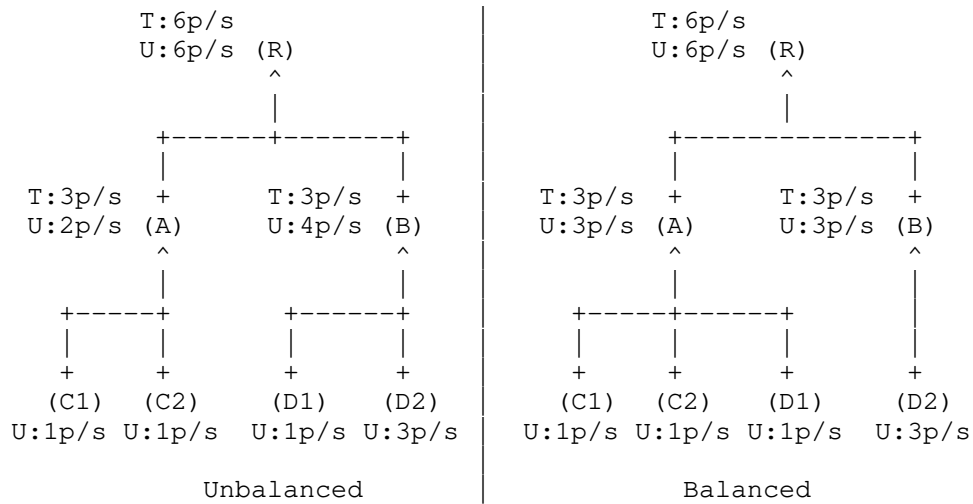


Figure 2: Use of Remaining Throughput with nodes with different requirements

As a second simple example, an unbalanced network with nodes which have different throughput ("U:") is shown in Figure 2. In this case, node B is overloaded and node D1 should move to parent A, which as a space throughput of 1p/s. Its transformation into a balanced equivalent network is shown on the right.

#### 4.2. DODAG selection problem

The purpose of the following example is to show the problem of DODAG selection, and not to focus on selecting the best parent.

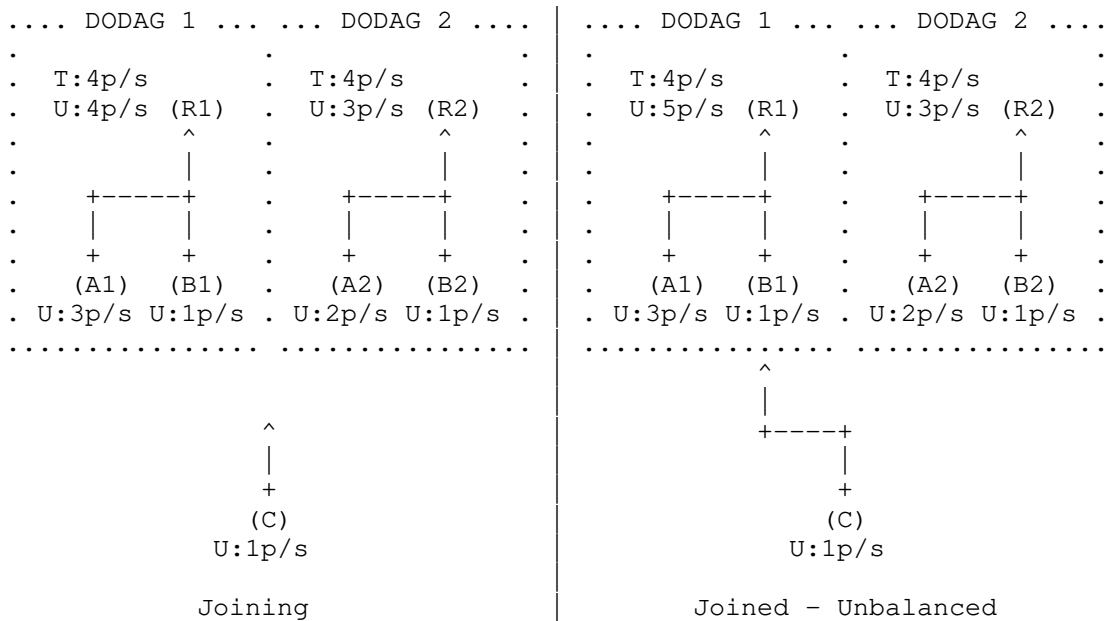


Figure 3: DODAG selection example leading to unbalanced traffic with RT metric

In the example in Figure 3, there are two DODAGs (DODAG 1 and DODAG 2) that belong to the same RPL Instance and a node (C) that must select a DODAG. Node C has to pick from the information provided by its two reachable neighbors: B1 and A2. On the left, node C is shown before selecting the preferred DODAG, while on the right it is shown after the DODAG selection.

Node C might choose B1 in DODAG 1 to be its preferred parent since the traffic information of the two DODAGs is not available. However, at the root node (R1), it can be observed that the total network traffic is higher in DODAG 1 and that after node C joins it, the traffic handling capacity of the root R1 has been exceeded.

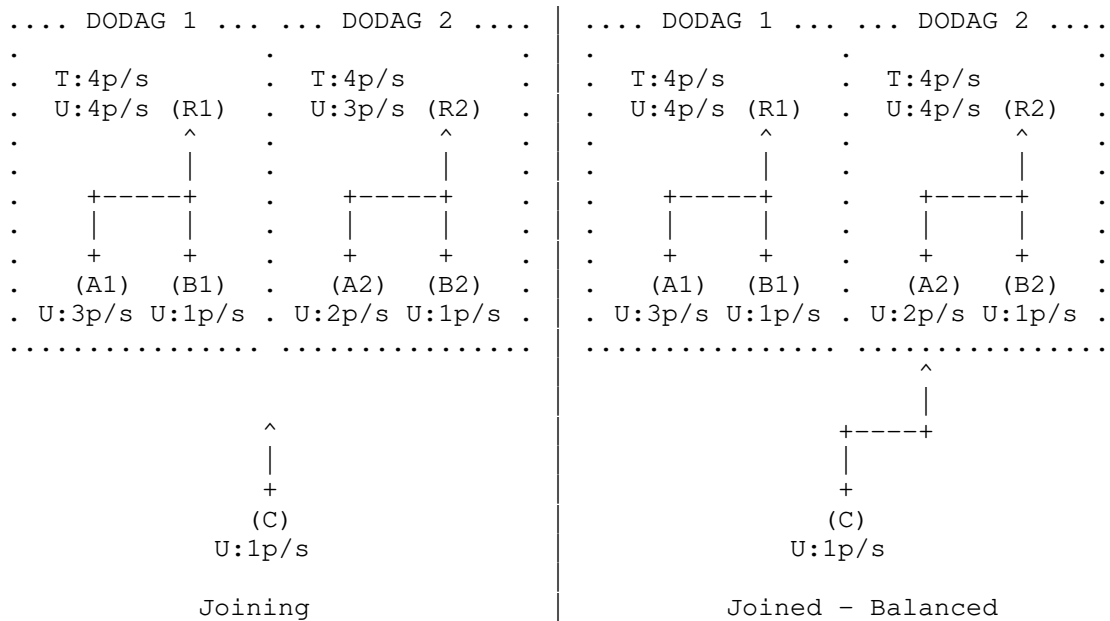


Figure 4: DODAG selection example leading to balanced traffic with RT metric

If the traffic handling capacity information is available, then node C could make a more efficient decision by using DODAG 2 and selecting node A2 as the preferred parent, as shown in Figure 4. Such a selection is based on the traffic of the entire DODAG and would not lead to exceeding the traffic handling capacity of the root R2 since this root had spare capacity.

## 5. TAOF description

In this specification, a metric is proposed to be used in the parent and DODAG selection mechanism, the Remaining Throughput (RT) which represents the number of packets each node can transmit (send or forward) during a certain time period. The period used, named `THROUGHPUT_PERIOD`, is a parameter common to the whole RPL instance. This parameter CAN be pre-configured on all the nodes. The period used SHOULD coincide with a sliding window of the same size used to calculate the packets transferred during this period to facilitate the calculation of the remaining possible packet transmissions. Therefore, whenever the RT value is reported it will refer to the previous `THROUGHPUT_PERIOD` period of time. This information is added in DIO messages and is broadcast to every neighbor.



At first, each node MUST identify from their neighbor set which nodes are acceptable to be selected as a parent. For this purpose, the metric ETX is used as a filter to filter out parent candidates with low link quality with a preference for nodes with link quality below a given threshold. The ETX threshold SHOULD be different depending on application requirements. The suggested value for the relevant threshold MAX\_PATH\_COST from MRHOF [RFC6719] is 32768, which means the specific path has expected transmission counts greater than 256.

When the ETX value is used as a filter, nodes with bad link quality will not be included in the parent set. This ensures that undue retransmissions caused by bad links will be avoided. After all the filtering is done, if any, the node chooses the parent candidate or DODAG with the highest remaining throughput.

For the purpose of DODAG specifically, the A field in the Routing Metric/Constraint Flag field object [RFC6551] SHOULD be set to 1, indicating that the value reported is a maximum. Furthermore, when a node is calculating the value of RT to broadcast in a DIO, the value reported SHOULD be the minimum of two values: its parent RT and the node's own calculated remaining throughput. Thus, the value broadcasted will be the available remaining throughput in the whole path from the node to the DODAG root.

This proposal is expected to increase the frequency of parent changes because the remaining throughput is more likely to be different between DIO messages, even for DIO messages from the same node. There are multiple ways to minimize the frequency of unnecessary parent changes:

- a. Use the remaining throughput in combination with another metric (e.g. child count, hop counts).
- b. Use a threshold when comparing the remaining throughput, similar to the approach in MRHOF [RFC6719]. Switch parents when the difference of remaining throughput between the original parent and the alternative parent is above a threshold. This threshold depends on different factors (e.g. network size, average traffic load) and SHOULD be defined differently for each use case.

#### 6. DIO Metric Container Type extension

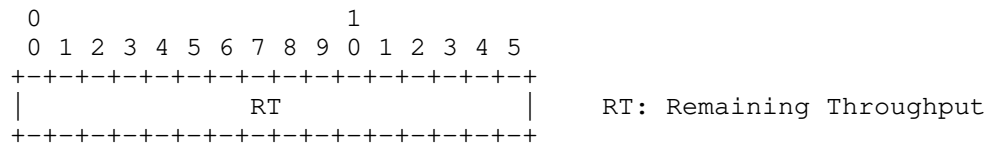


Figure 5: DAG metric container type format.

A DIO message carries fields as described in RFC6550 [RFC6550] and the available options for the DAG metric container are described in RFC6551 [RFC6551]. In this specification, a metric container option is proposed and the detailed format is shown in Figure 5. The information carried is the RT, represented as a 2 byte unsigned integer and the unit is packets per THROUGHPUT\_PERIOD time.

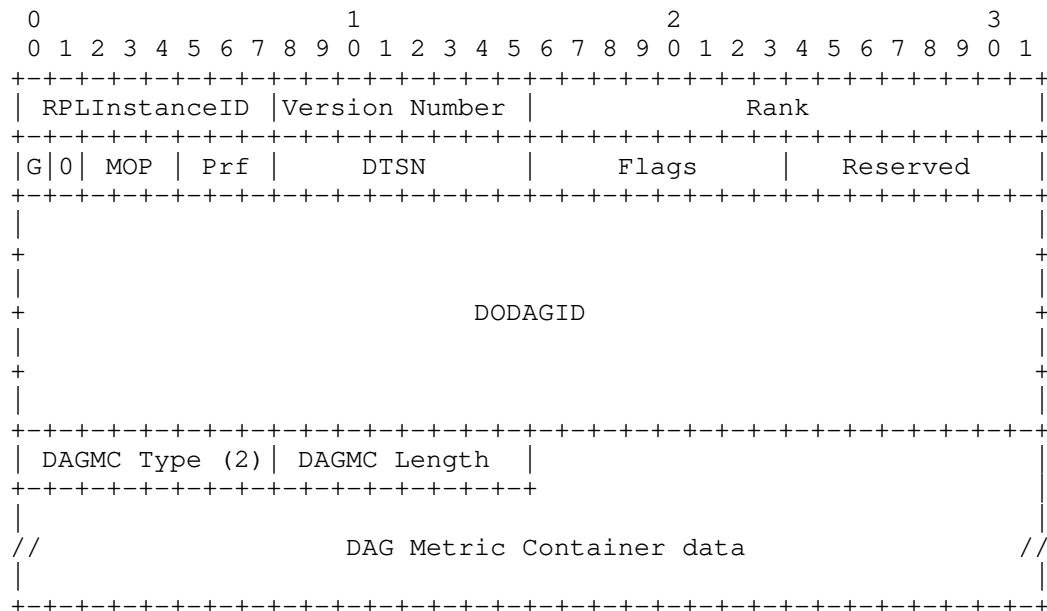


Figure 6: Example DIO Message with a DAG Metric Container option

The structure of the DIO Control Message when a DAG Metric Container option is included is shown in Figure 6. The DAG Metric Container option type (DAGMC Type in Figure 6) has the value 0x02 as per the IANA registry for the RPL Control Message Options and is defined in [RFC6550]. The DAG Metric Container option length (DAGMC Length in Figure 6) expresses the DAG Metric Container length in bytes. DAG Metric Container data holds the actual data and is shown further expanded in Figure 7.

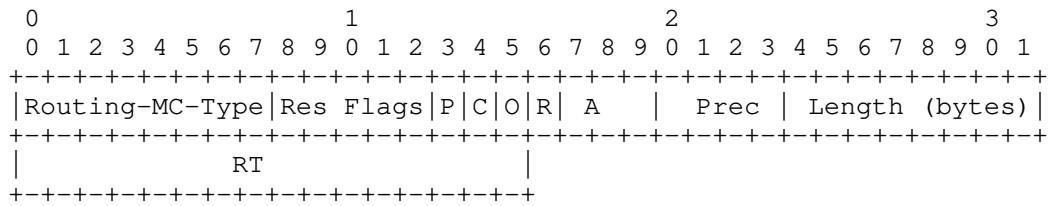


Figure 7: DAG Metric Container (MC) data with Remaining Throughput (RT) object body

An example DAG Metric Container containing the proposed Metric Container object is shown in Figure 7. The explicit definition of the fields is:

**Routing-MC-Type:** TBD1. The type of the proposed DAGMC extension. To be assigned by IANA.

**Remaining Throughput (RT):** The remaining throughput, represented as a 2-byte unsigned integer in units of packets per THROUGHPUT\_PERIOD time.

## 7. Enrollment

The RT metric SHOULD be used not only for ongoing parent selection but also especially within enrollment, i.e. the process a node follows to join a 6TiSCH network. In accordance to [I-D.ietf-6tisch-enrollment-enhanced-beacon], the value in RT SHOULD be used to affect the enrollment process so that a new node will be able to directly select a DODAG which will be able to cover its traffic needs and to spread the traffic load between different DODAGs. More specifically, the pan priority field described in [I-D.ietf-6tisch-enrollment-enhanced-beacon] can be derived from the RT value. For this purpose, the RT value SHOULD be used with the maximum value aggregation mode (A field in the Routing Metric/Constraint Flag field object [RFC6551] set to 1), to report the maximum remaining throughput in the whole path to the DODAG root. The pan priority field is an unsigned 8-bit integer with lower values signifying higher priority while the RT value is a 16-bit unsigned integer with higher values signifying more remaining throughput. To convert the RT value to a pan priority the following formula should be used:

$$\text{pan priority} = 16 - \text{FLOOR}(\text{LOG}_2(\text{RT} + 1))$$

where LOG2 is the logarithm function with a base of 2. The use of the LOG function allows having higher accuracy in the low values of

the remaining throughput, where small value differences are significant, and lower accuracy in the high values of the remaining throughput, where small differences are less significant. The addition of 1 to the RT allows converting RT=0.

## 8. Security Considerations

The structure of the DIO control message is extended, within the pre-defined DIO options. Therefore, the security mechanisms defined in RPL [RFC6550] apply to this proposed extension.

## 9. IANA Considerations

This proposal requests the allocation of a new value TBD1 for the metric type "RT" in the Routing-MC-Type field in the DAG MC from IANA.

Additionally, an Objective Code Point (OCP) with value TBD2 for TAOF needs to be assigned in the Objective Code Point Registry as described in Section 20.5 of [RFC6550].

## 10. Informative references

- [I-D.ietf-6tisch-enrollment-enhanced-beacon]  
Dujovne, D. and M. Richardson, "IEEE802.15.4 Informational Element encapsulation of 6tisch Join and Enrollment Information", draft-ietf-6tisch-enrollment-enhanced-beacon-00 (work in progress), July 2018.
- [I-D.qasem-roll-rpl-load-balancing]  
Qasem, M., Al-Dubai, A., Romdhani, I., Ghaleb, B., Hou, J., and R. Jadhav, "Load Balancing Objective Function in RPL", draft-qasem-roll-rpl-load-balancing-02 (work in progress), October 2017.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <<https://www.rfc-editor.org/info/rfc6550>>.

- [RFC6551] Vasseur, JP., Ed., Kim, M., Ed., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks", RFC 6551, DOI 10.17487/RFC6551, March 2012, <<https://www.rfc-editor.org/info/rfc6551>>.
- [RFC6552] Thubert, P., Ed., "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6552, DOI 10.17487/RFC6552, March 2012, <<https://www.rfc-editor.org/info/rfc6552>>.
- [RFC6719] Gnawali, O. and P. Levis, "The Minimum Rank with Hysteresis Objective Function", RFC 6719, DOI 10.17487/RFC6719, September 2012, <<https://www.rfc-editor.org/info/rfc6719>>.

## Authors' Addresses

Chenyang Ji (editor)  
Alexander TEI of Thessaloniki  
Department of Informatics  
Thessaloniki 57400  
GREECE

Email: [jichenyang920@gmail.com](mailto:jichenyang920@gmail.com)

Remous-Aris Koutsiamanis  
IMT Atlantique  
Office B00 - 126A  
2 Rue de la Chataigneraie  
Cesson-Sevigne - Rennes 35510  
FRANCE

Phone: +33 299 12 70 49  
Email: [aris@ariskou.com](mailto:aris@ariskou.com)

Georgios Papadopoulos  
IMT Atlantique  
Office B00 - 114A  
2 Rue de la Chataigneraie  
Cesson-Sevigne - Rennes 35510  
FRANCE

Phone: +33 299 12 70 04  
Email: [georgios.papadopoulos@imt-atlantique.fr](mailto:georgios.papadopoulos@imt-atlantique.fr)

Diego Dujovne  
Universidad Diego Portales  
Escuela de Informatica y Telecomunicaciones, Av. Ejercito 441  
Santiago, Region Metropolitana  
Chile

Phone: +56 (2) 676-8121  
Email: diego.dujovne@mail.udp.cl

Nicolas Montavont  
IMT Atlantique  
Office B00 - 106A  
2 Rue de la Chataigneraie  
Cesson-Sevigne - Rennes 35510  
FRANCE

Phone: +33 299 12 70 23  
Email: nicolas.montavont@imt-atlantique.fr

ROLL  
Internet-Draft  
Intended status: Standards Track  
Expires: January 4, 2019

R. Koutsiamanis, Ed.  
G. Papadopoulos  
N. Montavont  
IMT Atlantique  
P. Thubert  
Cisco  
July 3, 2018

RPL DAG Metric Container Node State and Attribute object type extension  
draft-koutsiamanis-roll-nsa-extension-02

## Abstract

Implementing 6TiSCH Packet Replication and Elimination from / to the RPL root requires the ability to forward copies of packets over different paths via different RPL parents. Selecting the appropriate parents to achieve ultra-low latency and jitter requires information about a node's parents. This document details what information needs to be transmitted and how it is encoded within a packet to enable this functionality.

## Status of This Memo

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

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

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

This Internet-Draft will expire on January 4, 2019.

## Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	3
3. Tracks . . . . .	3
3.1. Tracks Overview . . . . .	3
3.2. Complex Tracks . . . . .	4
4. Packet Replication and Elimination principles . . . . .	4
5. Alternative Parent Selection Issue . . . . .	5
6. Node State and Attribute (NSA) object type extension . . . . .	6
6.1. Usage . . . . .	8
6.1.1. DAG Metric Container fields . . . . .	9
6.1.2. Node State and Attribute fields . . . . .	9
6.2. Compression . . . . .	9
7. Security Considerations . . . . .	9
8. IANA Considerations . . . . .	10
9. References . . . . .	10
9.1. Informative references . . . . .	10
9.2. Other Informative References . . . . .	11
Authors' Addresses . . . . .	11

## 1. Introduction

Industrial network applications have stringent requirements on reliability and predictability, and typically leverage 1+1 redundancy, aka Packet Replication and Elimination (PRE) [I-D.papadopoulos-6tisch-pre-reqs] to achieve their goal. In order for wireless networks to be able to be used in such applications, the principles of Deterministic Networking [I-D.ietf-detnet-architecture] lead to designs that aim at maximizing packet delivery rate and minimizing latency and jitter. Additionally, given that the network nodes often do not have an unlimited power supply, energy consumption needs to be minimized as well.

To meet this goal, IEEE Std. 802.15.4 [IEEE802154-2015] provides Time-Slotted Channel Hopping (TSCH), a mode of operation which uses a fixed communication schedule to allow deterministic medium access as well as channel hopping to work around radio interference. However, since TSCH uses retransmissions in the event of a failed transmission, end-to-end delay and jitter performance can deteriorate.



The 6TiSCH working group, focusing on IPv6 over IEEE Std. 802.15.4-TSCH, has worked on the issues previously highlighted and produced the "6TiSCH Architecture" [I-D.ietf-6tisch-architecture] to address that case. Building on this architecture, "Exploiting Packet Replication and Elimination in Complex Tracks in 6TiSCH LLNs" [I-D.papadopoulos-6tisch-pre-reqs] leverages PRE to improve the Packet Delivery Ratio (PDR), provide a hard bound to the end-to-end latency, and limit jitter.

PRE achieves a controlled redundancy by laying multiple forwarding paths through the network and using them in parallel for different copies of a same packet. PRE can follow the Destination-Oriented Directed Acyclic Graph (DODAG) formed by RPL from a node to the root. Building a multi-path DODAG can be achieved based on the RPL capability of having multiple parents for each node in a network, a subset of which is used to forward packets. In order for this subset to be defined, a RPL parent subset selection mechanism, which falls within the remit of the RPL Objective Function (OF), needs to have specific path information. The specification of the transmission of this information is the focus of this document.

More concretely, this specification focuses on the extensions to the DAG Metric Container [RFC6551] required for providing the PRE mechanism a part of the information it needs to operate. This information is the RPL [RFC6550] parent address set of a node and it must be sent to potential children nodes of the node. The RPL DIO Control Message is the canonical way of broadcasting this kind of information and therefore its DAG Metric Container [RFC6551] field is used to append a Node State and Attribute (NSA) object. The node's parent address set is stored as an optional TLV within the NSA object. This specification defines the type value and structure for this TLV.

## 2. Terminology

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

## 3. Tracks

### 3.1. Tracks Overview

The concept of Track is introduced in the "6TiSCH Architecture" [I-D.ietf-6tisch-architecture], defined as a sequence of elements, each consisting of the 3-tuple of a transmitter, a receiver, and a given timeslot expressed as a slotOffset/channelOffset tuple. A simple Track is intended to provide the full resources required to

allow the transmission of a single packet from a source 6TiSCH node to a destination 6TiSCH node across a 6TiSCH multihop path.

### 3.2. Complex Tracks

Similarly to, but as a generalization of a simple Track, a Complex Track is defined in the "6TiSCH Architecture" [I-D.ietf-6tisch-architecture] as a DODAG starting at a source 6TiSCH node and leading to a sink 6TiSCH node in order to support multi-path forwarding. Multiple independent paths may be produced by using techniques for Packet Replication and Elimination (PRE) [I-D.papadopoulos-6tisch-pre-reqs] based on DetNet [I-D.ietf-detnet-architecture] principles. As an example, a complex Track allows for branching off and rejoining over non-congruent paths.

In the following Section, we will detail Deterministic Networks PRE techniques.

## 4. Packet Replication and Elimination principles

The idea behind Packet Replication and Elimination (PRE) is to transmit the same data packet through parallel and adjacent paths in a network with the aim of improving reliability and predictability through redundancy.

The process of replication consists of identifying multiple potential paths, selecting a subset to use, and sending copies of a single packet through each path. When receiving packets the process of elimination is required so that multiple copies of the same packet are not replicated again, to avoid an exponential growth in unnecessary traffic. Combined together, these processes enable controlled redundancy which in turn can be used to achieve the previously stated goals of reliability (i.e., ultra-high packet delivery rate) and predictability (i.e., ultra-low end-to-end delay and jitter) in wireless networks. For example, in Figure 1, the source 6TiSCH node S is sending the data packet to its RPL Default Parent (DP) (node A) and Alternative Parent (AP) (node B) in two different timeslots.

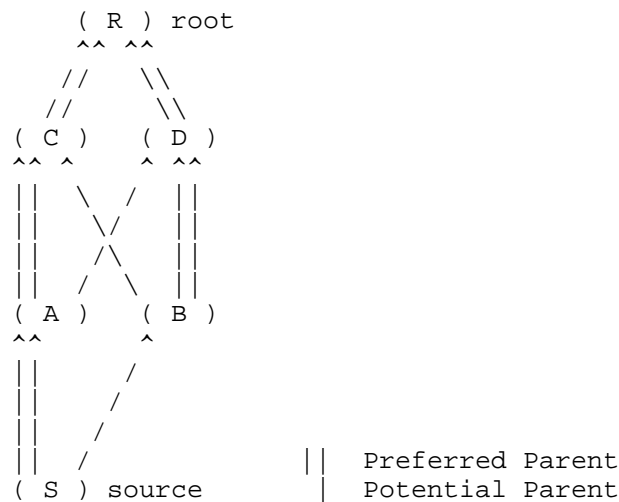


Figure 1: Packet Replication: S transmits the same data packet twice: to its DP (A) and to its AP (B).

In "Exploiting Packet Replication and Elimination in Complex Tracks in 6TiSCH LLNs" [I-D.papadopoulos-6tisch-pre-reqs], the concept of PRE is further expanded along with its requirements.

## 5. Alternative Parent Selection Issue

In the RPL protocol, each node maintains a list of potential parents. For PRE, the DP node is defined as the RPL DODAG preferred parent node. Furthermore, to construct an alternative path toward the root, in addition to the DP node, each 6TiSCH node in the network registers an AP node as well. There are multiple alternative methods of selecting the AP node, functionality which is included in operation of the RPL Objective Function (OF). In "Exploiting Packet Replication and Elimination in Complex Tracks in 6TiSCH LLNs" [I-D.papadopoulos-6tisch-pre-reqs], a scheme which allows the two paths to remain correlated is detailed. More specifically, in this scheme a 6TiSCH node will select an alternative parent node close to its default parent node to allow the operation of overhearing between parents. To do so, the node will check if its Default Grand Parent (DGP), the DP of its DP, is in the set of parents of a potential AP. If multiple potential APs match this condition, the AP with the lowest rank will be registered.

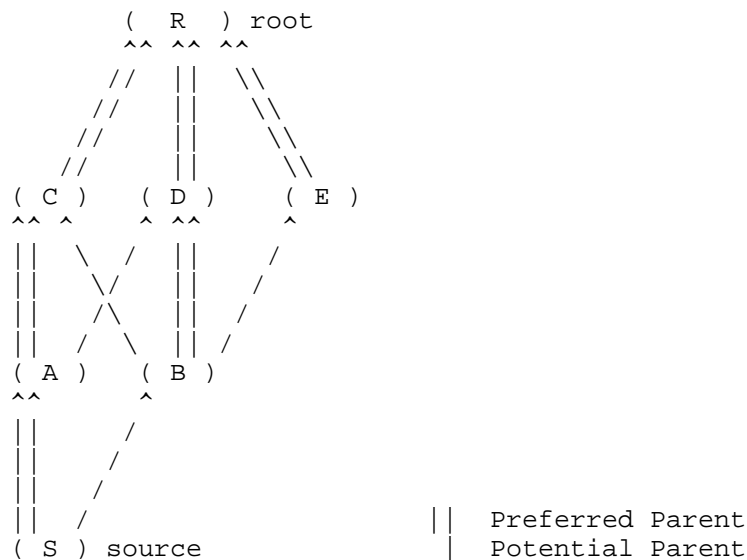


Figure 2: Example Parent Selection mechanism

For instance, in Figure 2, source 6TiSCH node S must know its grandparent sets both through node A and through node B. In this scenario, node A has the parent set {C, D} with C as DP and node B has the parent set {C, D, E} with D as DP. Therefore, node S can decide to use node B as its AP node, since the the DGP of S (via node A) is node C, and node C is in the parent set of node B ({C, D, E}).

In order to select their AP node, 6TiSCH nodes need to be aware of their grandparent node sets. Within RPL [RFC6550], the nodes use the DODAG Information Object (DIO) Control Message to broadcast information about themselves to potential children. However, RPL [RFC6550], does not define how to propagate parent set related information, which is what this document addresses.

## 6. Node State and Attribute (NSA) object type extension

For supporting PRE, nodes need to report their parent set to their potential children. DIO messages can carry multiple options, out of which the DAG Metric Container option [RFC6551] is the most suitable structurally and semantically for the purpose of carrying the parent set. The DAG Metric Container option itself can carry different nested objects, out of which the Node State and Attribute (NSA) [RFC6551] is appropriate for transferring generic node state data. Within the Node State and Attribute it is possible to store optional TLVs representing various node characteristics. As per the Node State and Attribute (NSA) [RFC6551] description, no TLV have been

defined for use. This document defines one TLV for the purpose of transmitting a node's parent set.

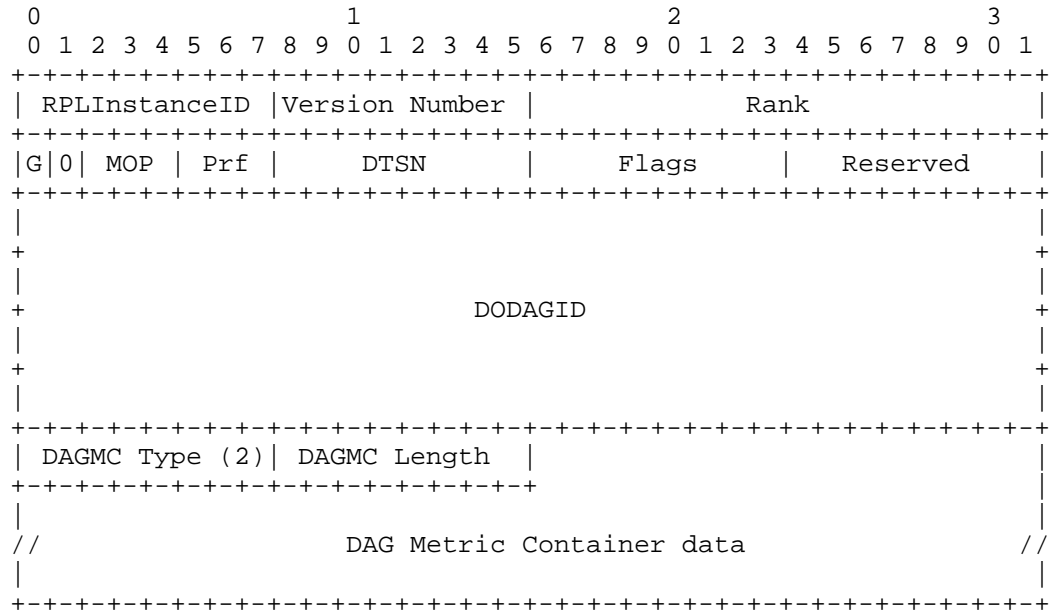


Figure 3: Example DIO Message with a DAG Metric Container option

The structure of the DIO Control Message when a DAG Metric Container option is included is shown in Figure 3. The DAG Metric Container option type (DAGMC Type in Figure 3) has the value 0x02 as per the IANA registry for the RPL Control Message Options, and is defined in [RFC6550]. The DAG Metric Container option length (DAGMC Length in Figure 3) expresses the the DAG Metric Container length in bytes. DAG Metric Container data holds the actual data and is shown further expanded in Figure 4.

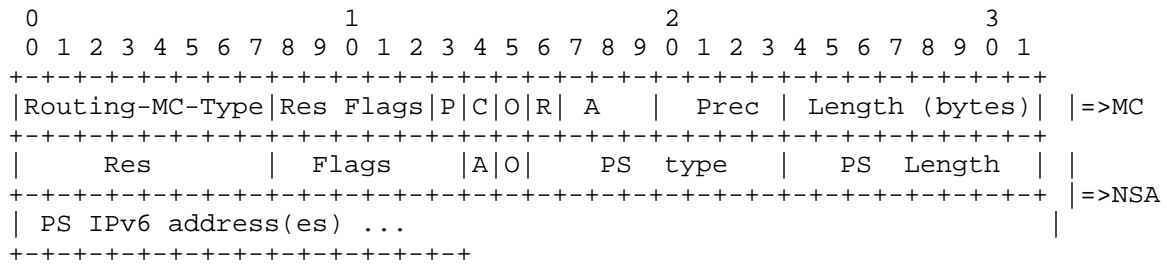


Figure 4: DAG Metric Container (MC) data with Node State and Attribute (NSA) object body and a TLV

The structure of the DAG Metric Container data in the form of a Node State and Attribute (NSA) object with a TLV in the NSA Optional TLVs field is shown in Figure 4. The first 32 bits comprise the DAG Metric Container header and all the following bits are part of the Node State and Attribute object body, as defined in [RFC6551]. This document defines a new TLV, which CAN be carried in the Node State and Attribute (NSA) object Optional TLVs field. The TLV is named Parent Set and is abbreviated as PS in Figure 4.

PS type: The type of the Parent Set TLV. The value is TBD1.

PS Length: The total length of the TLV value field (PS IPv6 address(es)) in bytes.

PS IPv6 address(es): A sequence of zero or more IPv6 addresses belonging to a node's parent set. Each address requires 16 bytes. The order of the parents in the parent set is in decreasing preference based on the Objective Function [RFC6550] used by the node.

#### 6.1. Usage

The PS SHOULD be used in the process of parent selection, and especially in alternative parent selection, since it can help the alternative path from significantly deviating from the preferred path. The Parent Set is information local to the node that broadcasts it. It does not make sense for this information to be aggregated due to the scalability issue created by the space required for many IPv6 addresses. Therefore, the PS MUST NOT be aggregated.

#### 6.1.1.1. DAG Metric Container fields

Given the intended usage, when using the PS, the NSA object it is contained in MUST be used as a constraint in the DAG Metric Container. More specifically, using the PS places the following requirements on the DAG Metric Container header fields:

- o 'P' flag: MUST be cleared, since PS is used only with constraints.
- o 'C' flag: MUST be set, since PS is used only with constraints.
- o 'O' flag: Used as per [RFC6550], to indicated optionality.
- o 'R' flag: MUST be cleared, since PS is used only with constraints.
- o 'A' Field: MUST be set to 0 and ignored, since PS is used only with constraints.
- o 'Prec' Field: Used as per [RFC6550].

#### 6.1.1.2. Node State and Attribute fields

For reasons of clarity, the usage of the PS places no additional restrictions on the NSA flags ('A' and 'O'), which can be used as normally defined in [RFC6550].

#### 6.2. Compression

The PS IPv6 address(es) field in the Parent Set TLV add overhead due to their size. Therefore, compression is highly desirable in order for this extension to be usable. To meet this goal, a good compression method candidate is [RFC8138] 6LoWPAN Routing Header (6LoRH). Furthermore, the PS IPv6 address(es) belong by definition to nodes in the same RPL DODAG and are stored in the form of a list of addresses. This makes this field a good candidate for the use of the same compression as in Source Routing Header 6LoRH (SRH-6LoRH), achieving efficiency and implementation reuse. Therefore, the PS IPv6 address(es) field SHOULD be compressed using the compression method for Source Routing Header 6LoRH (SRH-6LoRH) [RFC8138].

#### 7. Security Considerations

TODO.

## 8. IANA Considerations

TBA.

## 9. References

### 9.1. Informative references

[I-D.ietf-6tisch-architecture]

Thubert, P., "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4", draft-ietf-6tisch-architecture-14 (work in progress), April 2018.

[I-D.ietf-detnet-architecture]

Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-06 (work in progress), June 2018.

[I-D.papadopoulos-6tisch-pre-reqs]

Papadopoulos, G., Montavont, N., and P. Thubert, "Exploiting Packet Replication and Elimination in Complex Tracks in 6TiSCH LLNs", draft-papadopoulos-6tisch-pre-reqs-01 (work in progress), December 2017.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

[RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <<https://www.rfc-editor.org/info/rfc6550>>.

[RFC6551] Vasseur, JP., Ed., Kim, M., Ed., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks", RFC 6551, DOI 10.17487/RFC6551, March 2012, <<https://www.rfc-editor.org/info/rfc6551>>.

[RFC8138] Thubert, P., Ed., Bormann, C., Toutain, L., and R. Cragie, "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing Header", RFC 8138, DOI 10.17487/RFC8138, April 2017, <<https://www.rfc-editor.org/info/rfc8138>>.



## 9.2. Other Informative References

[IEEE802154-2015]

IEEE standard for Information Technology, "IEEE Std  
802.15.4-2015 Standard for Low-Rate Wireless Personal Area  
Networks (WPANs)", December 2015.

## Authors' Addresses

Remous-Aris Koutsiamanis (editor)  
IMT Atlantique  
Office B00 - 126A  
2 Rue de la Chataigneraie  
Cesson-Sevigne - Rennes 35510  
FRANCE

Phone: +33 299 12 70 49  
Email: aris@ariskou.com

Georgios Papadopoulos  
IMT Atlantique  
Office B00 - 114A  
2 Rue de la Chataigneraie  
Cesson-Sevigne - Rennes 35510  
FRANCE

Phone: +33 299 12 70 04  
Email: georgios.papadopoulos@imt-atlantique.fr

Nicolas Montavont  
IMT Atlantique  
Office B00 - 106A  
2 Rue de la Chataigneraie  
Cesson-Sevigne - Rennes 35510  
FRANCE

Phone: +33 299 12 70 23  
Email: nicolas.montavont@imt-atlantique.fr

Pascal Thubert  
Cisco Systems, Inc  
Building D  
45 Allee des Ormes - BP1200  
MOUGINS - Sophia Antipolis 06254  
FRANCE

Phone: +33 497 23 26 34  
Email: pthubert@cisco.com

ROLL  
Internet-Draft  
Intended status: Standards Track  
Expires: May 19, 2019

R. Koutsiamanis, Ed.  
G. Papadopoulos  
N. Montavont  
IMT Atlantique  
P. Thubert  
Cisco  
November 15, 2018

RPL DAG Metric Container Node State and Attribute object type extension  
draft-koutsiamanis-roll-nsa-extension-04

## Abstract

Implementing 6TiSCH Packet Replication and Elimination from / to the RPL root requires the ability to forward copies of packets over different paths via different RPL parents. Selecting the appropriate parents to achieve ultra-low latency and jitter requires information about a node's parents. This document details what information needs to be transmitted and how it is encoded within a packet to enable this functionality.

## Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://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 May 19, 2019.

## Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	3
3. Alternative Parent Selection . . . . .	4
3.1. Common Ancestor Strict . . . . .	4
3.2. Common Ancestor Medium . . . . .	5
3.3. Common Ancestor Relaxed . . . . .	5
4. Node State and Attribute (NSA) object type extension . . . . .	6
4.1. Usage . . . . .	8
4.1.1. DAG Metric Container fields . . . . .	8
4.1.2. Node State and Attribute fields . . . . .	9
4.2. Compression . . . . .	9
5. Controlling PRE . . . . .	9
6. Security Considerations . . . . .	9
7. IANA Considerations . . . . .	9
8. References . . . . .	9
8.1. Informative references . . . . .	10
8.2. Other Informative References . . . . .	10
Appendix A. Implementation Status . . . . .	11
Authors' Addresses . . . . .	13

## 1. Introduction

Industrial network applications have stringent requirements on reliability and predictability, and typically leverage 1+1 redundancy, aka Packet Replication and Elimination (PRE) [I-D.papadopoulos-6tisch-pre-reqs] to achieve their goal. In order for wireless networks to be able to be used in such applications, the principles of Deterministic Networking [I-D.ietf-detnet-architecture] lead to designs that aim at maximizing packet delivery rate and minimizing latency and jitter. Additionally, given that the network nodes often do not have an unlimited power supply, energy consumption needs to be minimized as well.

To meet this goal, IEEE Std. 802.15.4 [IEEE802154-2015] provides Time-Slotted Channel Hopping (TSCH), a mode of operation which uses a fixed communication schedule to allow deterministic medium access as well as channel hopping to work around radio interference. However, since TSCH uses retransmissions in the event of a failed transmission, end-to-end delay and jitter performance can deteriorate.

The 6TiSCH working group, focusing on IPv6 over IEEE Std. 802.15.4-TSCH, has worked on the issues previously highlighted and produced the "6TiSCH Architecture" [I-D.ietf-6tisch-architecture] to address that case. Building on this architecture, "Exploiting Packet Replication and Elimination in Complex Tracks in 6TiSCH LLNs" [I-D.papadopoulos-6tisch-pre-reqs] leverages PRE to improve the Packet Delivery Ratio (PDR), provide a hard bound to the end-to-end latency, and limit jitter.

PRE achieves a controlled redundancy by laying multiple forwarding paths through the network and using them in parallel for different copies of a same packet. PRE can follow the Destination-Oriented Directed Acyclic Graph (DODAG) formed by RPL from a node to the root. Building a multi-path DODAG can be achieved based on the RPL capability of having multiple parents for each node in a network, a subset of which is used to forward packets. In order for this subset to be defined, a RPL parent subset selection mechanism, which falls within the remit of the RPL Objective Function (OF), needs to have specific path information. The specification of the transmission of this information is the focus of this document.

More concretely, this specification focuses on the extensions to the DAG Metric Container [RFC6551] required for providing the PRE mechanism a part of the information it needs to operate. This information is the RPL [RFC6550] parent address set of a node and it must be sent to potential children nodes of the node. The RPL DIO Control Message is the canonical way of broadcasting this kind of information and therefore its DAG Metric Container [RFC6551] field is used to append a Node State and Attribute (NSA) object. The node's parent address set is stored as an optional TLV within the NSA object. This specification defines the type value and structure for this TLV.

## 2. Terminology

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

The draft uses the following Terminology:

**Track** A sequence of 6TiSCH schedule resources to support a single-path multi-hop transmission of a packet. See "6TiSCH Architecture" [I-D.ietf-6tisch-architecture] for more.

**Complex Track** A Track which supports a multi-path multi-hop transmission of a packet. See "6TiSCH Architecture" [I-D.ietf-6tisch-architecture] for more.

Packet Replication and Elimination (PRE) The sending of multiple copies of a packet using multi-path forwarding over a multi-hop network and the consolidation of multiple received packet copies to control flooding. See "Exploiting Packet Replication and Elimination in Complex Tracks in 6TiSCH LLNs" [I-D.papadopoulos-6tisch-pre-reqs] for more.

Alternative Parent (AP) Selection The problem of how to select the next hop target node for a packet copy to be forwarded to when performing packet replication.

### 3. Alternative Parent Selection

In the RPL protocol, each node maintains a list of potential parents. For PRE, the PP node is defined to be the same as the RPL DODAG Preferred Parent (PP) node. Furthermore, to construct an alternative path toward the root, in addition to the PP node, each 6TiSCH node in the network registers an AP node as well from its Parent Set (PS). There are multiple alternative methods of selecting the AP node, functionality which is included in operation of the RPL Objective Function (OF). A scheme which allows the two paths to remain correlated is detailed here. More specifically, in this scheme a 6TiSCH node will select an alternative parent node close to its PP node to allow the operation of overhearing between parents. If multiple potential APs match this condition, the AP with the lowest rank will be registered.

There are at least three methods of performing the alternative parent selection based on common ancestors (CA), named Common Ancestor Strict, Common Ancestor Medium, and Common Ancestor Relaxed, depending on how restrictive the selection process is. A more restrictive method will limit flooding but might fail to select an appropriate alternative parent, while a less restrictive one will more often find an appropriate alternative parent but might increase flooding.

#### 3.1. Common Ancestor Strict

In CA Strict, the node will check if its Preferred Grand Parent (PGP), the PP of its PP, is the same as the PP of the potential AP.

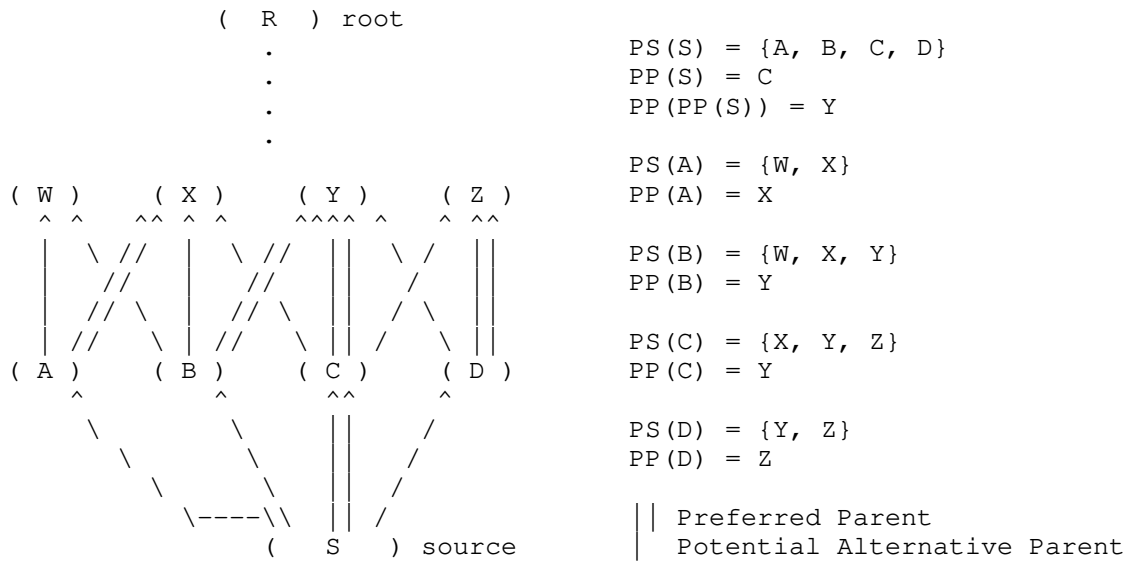


Figure 1: Example Common Ancestor Strict Alternative Parent Selection method

For example, in Figure 1, the source 6TiSCH node S must know its grandparent sets both through nodes A, B, C, and D. The Parent Sets (PS) and the Preferred Parents (PS) of nodes A, B, C, and D are shown on the side of the figure. The CA Strict parent selection method will select an AP for node S for which  $PP(PP(S)) = PP(AP)$ . Therefore, node S can decide to use node B as its AP node, since  $PP(PP(S)) = Y = PP(B)$ .

### 3.2. Common Ancestor Medium

In CA Medium, the node will check if its Preferred Grand Parent (PGP), the PP of its PP, is contained in the PS of the potential AP.

Using the same example, in Figure 1, the CA Medium parent selection method will select an AP for node S for which  $PP(PP(S)) \in PS(AP)$ . Therefore, node S can decide to use node B or D as its AP node, since given that  $PP(PP(S)) = Y$ , for node B  $PS(B) = \{W, X, Y\}$  and for node D  $PS(D) = \{Y, Z\}$ .

### 3.3. Common Ancestor Relaxed

In CA Relaxed, the node will check if its the Parent Set (PS) of its Preferred Parent (PP), has a common node with the PS of the potential AP.

Using the same example, in Figure 1, the CA Relaxed parent selection method will select an AP for node S for which  $PS(PP(S))$  has a non-empty intersection with  $PS(AP)$ . Therefore, node S can decide to use node A, B or D as its AP node. Given that  $PS(PP(S)) = \{X, Y, Z\}$  the alternative parent selection process evaluates the nodes:

- o Node A:  $PS(A) = \{W, X\}$  and the common nodes are  $\{X\}$
- o Node B:  $PS(B) = \{W, X, Y\}$  and the common nodes are  $\{X, Y\}$
- o Node D:  $PS(D) = \{Y, Z\}$  and the common nodes are  $\{Y, Z\}$

#### 4. Node State and Attribute (NSA) object type extension

In order to select their AP node, 6TiSCH nodes need to be aware of their grandparent node sets. Within RPL [RFC6550], the nodes use the DODAG Information Object (DIO) Control Message to broadcast information about themselves to potential children. However, RPL [RFC6550], does not define how to propagate parent set related information, which is what this document addresses.

DIO messages can carry multiple options, out of which the DAG Metric Container option [RFC6551] is the most suitable structurally and semantically for the purpose of carrying the parent set. The DAG Metric Container option itself can carry different nested objects, out of which the Node State and Attribute (NSA) [RFC6551] is appropriate for transferring generic node state data. Within the Node State and Attribute it is possible to store optional TLVs representing various node characteristics. As per the Node State and Attribute (NSA) [RFC6551] description, no TLV has been defined for use. This document defines one TLV for the purpose of transmitting a node's parent set.



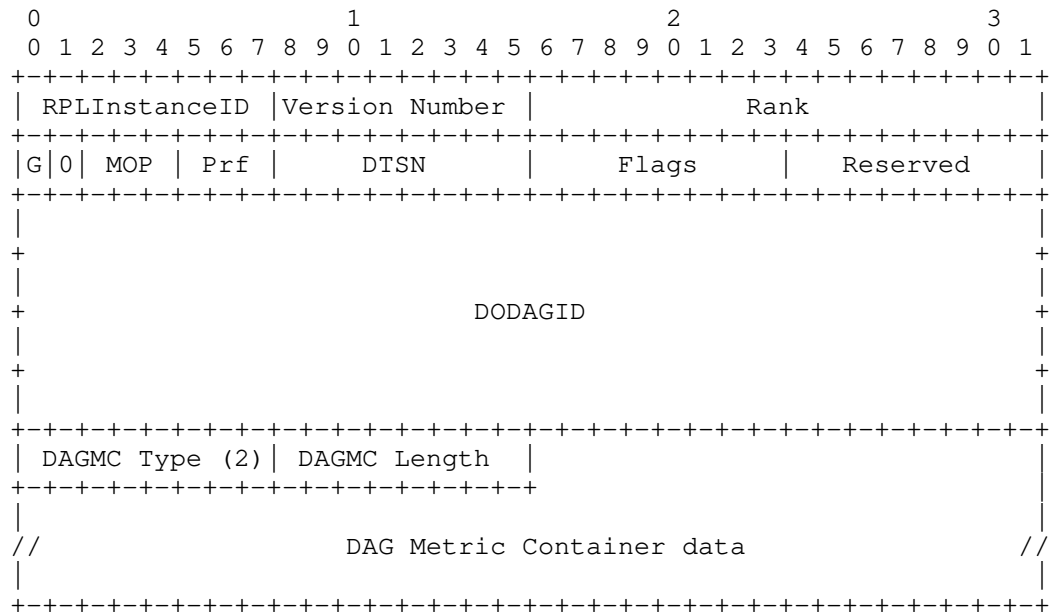


Figure 2: Example DIO Message with a DAG Metric Container option

Figure 2 shows the structure of the DIO Control Message when a DAG Metric Container option is included. The DAG Metric Container option type (DAGMC Type in Figure 2) has the value 0x02 as per the IANA registry for the RPL Control Message Options, and is defined in [RFC6550]. The DAG Metric Container option length (DAGMC Length in Figure 2) expresses the DAG Metric Container length in bytes. DAG Metric Container data holds the actual data and is shown expanded in Figure 3.

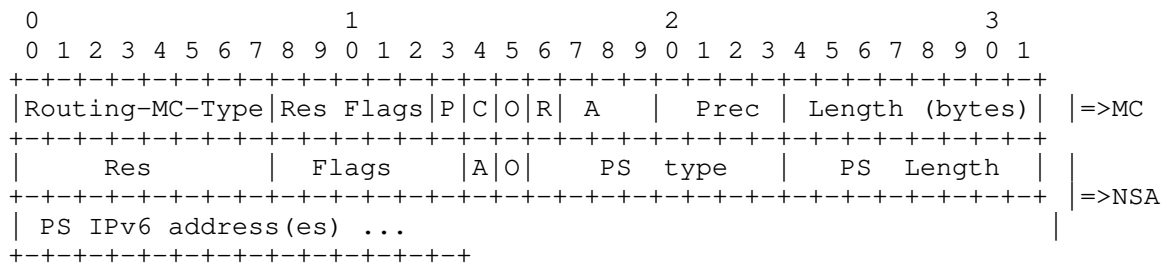


Figure 3: DAG Metric Container (MC) data with Node State and Attribute (NSA) object body and a TLV

The structure of the DAG Metric Container data in the form of a Node State and Attribute (NSA) object with a TLV in the NSA Optional TLVs field is shown in Figure 3. The first 32 bits comprise the DAG Metric Container header and all the following bits are part of the Node State and Attribute object body, as defined in [RFC6551]. This document defines a new TLV, which CAN be carried in the Node State and Attribute (NSA) object Optional TLVs field. The TLV is named Parent Set and is abbreviated as PS in Figure 3.

PS type: The type of the Parent Set TLV. The value is TBD1.

PS Length: The total length of the TLV value field (PS IPv6 address(es)) in bytes.

PS IPv6 address(es): A sequence of zero or more IPv6 addresses belonging to a node's parent set. Each address requires 16 bytes. The order of the parents in the parent set is in decreasing preference based on the Objective Function [RFC6550] used by the node.

#### 4.1. Usage

The PS SHOULD be used in the process of parent selection, and especially in alternative parent selection, since it can help the alternative path from significantly deviating from the preferred path. The Parent Set is information local to the node that broadcasts it.

##### 4.1.1. DAG Metric Container fields

Given the intended usage, when using the PS, the NSA object it is contained in MUST be used as a constraint in the DAG Metric Container. More specifically, using the PS places the following requirements on the DAG Metric Container header fields:

- o 'P' flag: MUST be cleared, since PS is used only with constraints.
- o 'C' flag: MUST be set, since PS is used only with constraints.
- o 'O' flag: Used as per [RFC6550], to indicated optionality.
- o 'R' flag: MUST be cleared, since PS is used only with constraints.
- o 'A' Field: MUST be set to 0 and ignored, since PS is used only with constraints.
- o 'Prec' Field: Used as per [RFC6550].

#### 4.1.2. Node State and Attribute fields

For clarity reasons, the usage of the PS places no additional restrictions on the NSA flags ('A' and 'O'), which can be used as normally defined in [RFC6550].

#### 4.2. Compression

The PS IPv6 address(es) field in the Parent Set TLV add overhead due to their size. Therefore, compression is highly desirable in order for this extension to be usable. To meet this goal, a good compression method candidate is [RFC8138] 6LoWPAN Routing Header (6LoRH). Furthermore, the PS IPv6 address(es) belong by definition to nodes in the same RPL DODAG and are stored in the form of a list of addresses. This makes this field a good candidate for the use of the same compression as in Source Routing Header 6LoRH (SRH-6LoRH), achieving efficiency and implementation reuse. Therefore, the PS IPv6 address(es) field SHOULD be compressed using the compression method for Source Routing Header 6LoRH (SRH-6LoRH) [RFC8138].

#### 5. Controlling PRE

PRE is very helpful when the aim is to increase reliability for a certain track, however it's use creates additional traffic as part of the replication process. It is conceivable that not all tracks have stringent reliability requirements. Therefore, a way to control whether PRE is applied to a track's packets SHOULD be implemented. For example, a traffic class label can be used to determine this behaviour per flow type as described in Deterministic Networking Architecture [I-D.ietf-detnet-architecture].

#### 6. Security Considerations

The structure of the DIO control message is extended, within the pre-defined DIO options. Therefore, the security mechanisms defined in RPL [RFC6550] apply to this proposed extension.

#### 7. IANA Considerations

This proposal requests the allocation of a new value TBD1 for the "Parent Set" TLV in the Routing Metric/Constraint TLVs sub-registry from IANA.

#### 8. References

## 8.1. Informative references

- [I-D.ietf-6tisch-architecture]  
Thubert, P., "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4", draft-ietf-6tisch-architecture-17 (work in progress), November 2018.
- [I-D.ietf-detnet-architecture]  
Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-09 (work in progress), October 2018.
- [I-D.papadopoulos-6tisch-pre-reqs]  
Papadopoulos, G., Montavont, N., and P. Thubert, "Exploiting Packet Replication and Elimination in Complex Tracks in 6TiSCH LLNs", draft-papadopoulos-6tisch-pre-reqs-02 (work in progress), July 2018.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <<https://www.rfc-editor.org/info/rfc6550>>.
- [RFC6551] Vasseur, JP., Ed., Kim, M., Ed., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks", RFC 6551, DOI 10.17487/RFC6551, March 2012, <<https://www.rfc-editor.org/info/rfc6551>>.
- [RFC8138] Thubert, P., Ed., Bormann, C., Toutain, L., and R. Cragie, "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing Header", RFC 8138, DOI 10.17487/RFC8138, April 2017, <<https://www.rfc-editor.org/info/rfc8138>>.

## 8.2. Other Informative References

- [IEEE802154-2015]  
IEEE standard for Information Technology, "IEEE Std 802.15.4-2015 Standard for Low-Rate Wireless Personal Area Networks (WPANs)", December 2015.

### 8.3. URIs

- [1] <https://github.com/ariskou/contiki/tree/draft-koutsiamanis-roll-nsa-extension>
- [2] <https://code.wireshark.org/review/gitweb?p=wireshark.git;a=commit;h=e2f6ba229f45d8ccae2a6405e0ef41f1e61da138>

### Appendix A. Implementation Status

A research-stage implementation of the PRE mechanism using the proposed extension as part of a 6TiSCH IOT use case was developed at IMT Atlantique, France by Tomas Lagos Jenschke and Remous-Aris Koutsiamanis. It was implemented on the open-source Contiki OS and tested with the Cooja simulator. The DIO DAGMC NSA extension is implemented with a configurable number of parents from the parent set of a node to be reported.

( R )

(11)	(12)	(13)	(14)	(15)	(16)
(21)	(22)	(23)	(24)	(25)	(26)
(31)	(32)	(33)	(34)	(35)	(36)
(41)	(42)	(43)	(44)	(45)	(46)
(51)	(52)	(53)	(54)	(55)	(56)

( S )

Figure 4: Simulation Topology

The simulation setup is:

Topology: 32 nodes structured in regular grid as show in Figure 4. Node S (source) is the only data packet sender, and send data to node R (root). The parent set of each node (except R) is all the nodes in the immediatelly higher row, the immediatelly above 6 nodes. For example, each node in {51, 52, 53, 54, 55, 56} is

connected to all of {41, 42, 43, 44, 45, 46}. Node 11, 12, 13, 14, 15, 16 have a single upwards link to R.

MAC: TSCH with 1 retransmission

Platform: Cooja

Schedule: Static, 2 timeslots per link from each node to each parent in its parent set, 1 broadcast EB slot, 1 sender-based shared timeslot (for DIO and DIS) per node (total of 32).

Simulation lifecycle: Allow link formation for 100 seconds before starting to send data packets. Afterwards, S sends data packets to R. The simulation terminates when 1000 packets have been sent by S.

Radio Links: Links are reset uniformly randomly between 70% and 100% every 60 seconds.

Traffic Pattern: CBR, S sends one non-fragmented UDP packet every 5 seconds to R.

PS extension size: 3 parents.

Routing Methods:

- \* RPL: The default RPL non-PRE implementation in Contiki OS.
- \* 2nd ETX: PRE with a parent selection method which picks as AP the 2nd best parent in the parent set based on ETX.
- \* CA Strict: As described in Section 3.1.
- \* CA Medium: As described in Section 3.2.

## Simulation results:

Routing Method	Average Packet Delivery Rate (%)	Average Traversed Nodes/packet (#)	Average Duplications/packet (#)
RPL	82.70	5.56	7.02
2nd ETX	99.38	14.43	31.29
CA	97.32	9.86	18.23
Strict CA	99.66	13.75	28.86
Medium			

## Links:

- o Contiki OS DIO DAGMC NSA extension (draft-koutsiamanis-roll-nsa-extension branch) [1]
- o Wireshark dissectors (for the optional TLV, i.e., PS) - currently merged / in master [2]

## Authors' Addresses

Remous-Aris Koutsiamanis (editor)  
 IMT Atlantique  
 Office B00 - 126A  
 2 Rue de la Chataigneraie  
 Cesson-Sevigne - Rennes 35510  
 FRANCE

Phone: +33 299 12 70 49  
 Email: aris@ariskou.com

Georgios Papadopoulos  
 IMT Atlantique  
 Office B00 - 114A  
 2 Rue de la Chataigneraie  
 Cesson-Sevigne - Rennes 35510  
 FRANCE

Phone: +33 299 12 70 04  
 Email: georgios.papadopoulos@imt-atlantique.fr

Nicolas Montavont  
IMT Atlantique  
Office B00 - 106A  
2 Rue de la Chataigneraie  
Cesson-Sevigne - Rennes 35510  
FRANCE

Phone: +33 299 12 70 23  
Email: nicolas.montavont@imt-atlantique.fr

Pascal Thubert  
Cisco Systems, Inc  
Building D  
45 Allee des Ormes - BP1200  
MOUGINS - Sophia Antipolis 06254  
FRANCE

Phone: +33 497 23 26 34  
Email: pthubert@cisco.com



ROLL  
Internet-Draft  
Intended status: Standards Track  
Expires: October 24, 2018

R. Jadhav, Ed.  
R. Sahoo  
Y. Wu  
Huawei  
April 22, 2018

RPL Observations  
draft-rahul-roll-rpl-observations-01

Abstract

This document describes RPL protocol design issues, various observations and possible consequences of the design and implementation choices.

Status of This Memo

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

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

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

This Internet-Draft will expire on October 24, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	2
1.1. Requirements Language and Terminology . . . . .	3
2. DTSN increment in storing MOP . . . . .	3
2.1. Deliberations . . . . .	4
3. DAO retransmission and use of DAO-ACK in storing MOP . . . . .	5
3.1. Significance of bidirectional Path establishment indication and relevance of DAO-ACK . . . . .	5
3.2. Problems with hop-by-hop DAO-ACK . . . . .	6
3.3. Problems with end-to-end DAO-ACK . . . . .	6
3.4. Deliberations . . . . .	6
3.5. Implementation Notes . . . . .	6
4. Handling resource unavailability . . . . .	7
4.1. Deliberations . . . . .	7
5. Handling aggregated targets . . . . .	7
5.1. Deliberations . . . . .	8
6. RPL Transit Information in DAO . . . . .	8
6.1. Deliberations . . . . .	8
7. Managing persistent variables across node reboots . . . . .	9
7.1. Persistent storage and RPL state information . . . . .	9
7.2. Lollipop Counters . . . . .	9
7.3. RPL State variables . . . . .	10
7.3.1. DODAG Version . . . . .	10
7.3.2. DTSN field in DIO . . . . .	11
7.3.3. PathSequence . . . . .	11
7.4. State variables update frequency . . . . .	11
7.5. Deliberations . . . . .	12
7.6. Implementation Notes . . . . .	12
8. RPL under-specification . . . . .	12
9. Acknowledgements . . . . .	13
10. IANA Considerations . . . . .	13
11. Security Considerations . . . . .	13
12. References . . . . .	13
12.1. Normative References . . . . .	13
12.2. Informative References . . . . .	14
Appendix A. Additional Stuff . . . . .	14
Authors' Addresses . . . . .	14

## 1. Introduction

RPL [RFC6550] specifies a proactive distance-vector routing scheme designed for LLNs (Low Power and Lossy Networks). RPL enables the network to be formed as a DODAG and supports storing mode and non-storing mode of operations. Non-storing mode allows reduced memory resource usage on the nodes by allowing non-BR nodes to operate without managing a routing table and involves use of source routing

by the 6LBR to direct the traffic along a specific path. In storing mode of operation intermediate routers maintain routing tables.

This work aims to highlight various issues with RPL which makes it difficult to handle certain scenarios. This work will highlight such issues in context to RPL's mode of operations (storing versus non-storing). There are cases where RPL does not provide clear rules and implementations have to make their choices hindering interoperability and performance.

[I-D.clausen-lln-rpl-experiences] provides some interesting points. Some sections in this draft may overlap with some observations in [clausen], but this is been done to further extend some scenarios or observations. It is highly encouraged that readers should also visit [I-D.clausen-lln-rpl-experiences] for other insights. Regardless, this draft is self-sufficient in a way that it does not expect to have read [clausen-draft].

### 1.1. Requirements Language and Terminology

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

NS-MOP = RPL Non-storing Mode of Operation

S-MOP = RPL Storing Mode of Operation

This document uses terminology described in [RFC6550] and [RFC6775].

## 2. DTSN increment in storing MOP

DTSN increment has major impact on the overall RPL control traffic and on the efficiency of downstream route update. DTSN is sent as part of DIO message and signals the downstream nodes to trigger the target advertisement. The 6LR needs to decide when to update the DTSN and usually it should do it in a conservative way. The DTSN update mechanism determines how soon the downward routes are established along the new path. RPL specifications does not provide any clear mechanism on how the DTSN update should happen in case of storing mode.

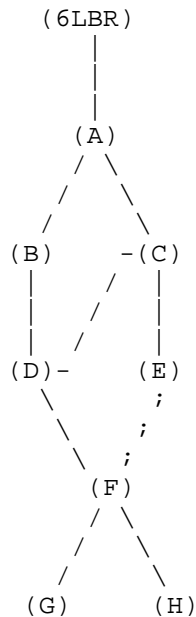


Figure 1: Sample topology

Consider example topology shown in Figure 1, assume that node D switches the parent from node B to C. Ideally the downstream nodes D and its sub-children should send their target advertisement to the new path via node C. To achieve this result in a efficient way is a challenge. Incrementing DTSN is the only way to trigger the DAO on downstream nodes. But this trigger should be sent not only on the first hop but to all the grand-child nodes. Thus DTSN has to be incremented in the complete sub-DODAG rooted at node D thus resulting in DIO/DAO storm along the sub-DODAG. This is specifically a big issue in high density networks where the metric deterioration might happen transiently even though the signal strength is good.

The primary implementation issue is whether a child node increment its own DTSN when it receives DTSN update from its parent node? This would result in DAO-updates in the sub-DODAG, thus the cost could be very high. If not incremented it may result in serious loss of connectivity for nodes in the sub-DODAG.

## 2.1. Deliberations

- (1) In S-MOP, should the child nodes increment its DIO on seeing that its preferred parent has updated its DTSN?

- (2) What are rules for DTSN increment for storing MOP, which multiple implementations can follow thus allowing consistent performance across different implementations?

### 3. DAO retransmission and use of DAO-ACK in storing MOP

[RFC6550] has an optional DAO-ACK mechanism using which an upstream parent confirms the reception of a DAO from the downstream child. In case of storing mode, the DAO is addressed to the immediate hop upstream parent resulting in DAO-ACK from the parent. There are two implementations possible:

- (1) Hop-by-hop ACK: A parent responds with a DAO-ACK immediately after receiving the DAO.
- (2) End-to-End ACK: A node waits for the upstream parent to send DAO-ACK to respond with a DAO-ACK downstream. The upstream parent may do as many attempts to successfully send this DAO upstream. In other words, the parent node accepts the responsibility of sending the DAO upstream till the point it is ACKed the moment it responds back with its own ACK to the child.

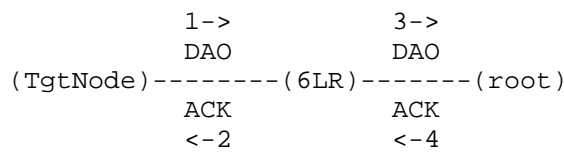


Figure 2: Hop-by-hop DAO-ACK

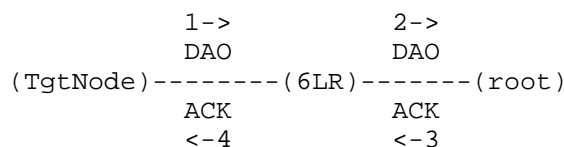


Figure 3: End-to-End DAO-ACK

#### 3.1. Significance of bidirectional Path establishment indication and relevance of DAO-ACK

Lot of application traffic patterns requires that the bidirectional path be established between the target node and the root. A typical example is that COAP request with ACK bit set would require an acknowledgement from the end receiver and thus warrants bidirectional path establishment. It is imperative that the target node first ascertains whether such a bidirectional path is established before initiating such application traffic. In case of non-storing MOP, the

DAO-ACK works perfectly fine to ascertain such bidirectional connectivity since it is an indication that the root which usually is the direct destination of the DAO has received the DAO. But in case of storing MOP, things are more complicated since DAO is sent hop-by-hop and the DAO-ACK semantics are not clear enough as per the current specification. As mentioned in above section, an implementation can choose to implement hop-by-hop ACK or end-to-end ACK.

### 3.2. Problems with hop-by-hop DAO-ACK

The primary issue with this mode is that target node cannot ascertain bidirection path connectivity on the reception of the DAO-ACK.

### 3.3. Problems with end-to-end DAO-ACK

In this case, it is possible for the target node to ascertain if the DAO has indeed reached the root since the reception of DAO-ACK on target node confirms this. However there is extra state information that needs to be maintained on the 6LRs on behalf of all the child nodes. Also it is very difficult for the target node to ascertain a timer value to decide whether the DAO transmission has failed to reach the root.

### 3.4. Deliberations

- (1) How should an implementation interpret the DAO-ACK semantics?
- (2) What is the best way for the target node to know that the end to end bidirectional path is successfully installed or updated? In NS-MOP, the DAO-ACK provides a clear way to do this. Can the same be achieved for storing-MOP?
- (3) What happens if the DAO-ACK with Status!=0 is responded by ancestor node?
- (4) How to selectively NACK subset of targets in case target containers are aggregated?

### 3.5. Implementation Notes

Current RPL open source implementations have both types of DAO-ACK implementations. For e.g. RIOT supports hop-by-hop DAO-ACK. Contiki older versions supported hop-by-hop ACK but the recent version have changed to end-to-end ACK implementation.

The sequence of sending no-path DAO and DAO matters when updating the routing adjacencies on a parent switch. If an implementation chooses to send no-path DAO before DAO then it results in significantly more

overhead for route invalidation. This is because no-path DAO would traverse all the way up to the BR clearing the routes on the way. In case there is a common ancestor post which the old and new path remains same then it is better to send regular DAO first thus limiting the propagation of subsequent no-path DAO till this common ancestor.

#### 4. Handling resource unavailability

The nodes in the constrained networks have to maintain various records such as neighbor cache entries and routing entries on behalf of other targets to facilitate packet forwarding. Because of the constrained nature of the devices the memory available may be very limited and thus the path selection algorithm may have to take into consideration such resource constraints as well.

RPL currently does not have any mechanism to advertise such resource indicator metrics. The primary tables associated with RPL are routing table and the neighbor cache. Even though neighbor cache is not directly linked with RPL protocol, the maintenance of routing adjacencies results in updates to neighbor cache.

##### 4.1. Deliberations

Is it possible to know that an upstream parent/ancestor cannot hold enough routing entries and thus this path should not be used?

Is it possible to know that an upstream parent cannot hold any more neighbor cache entry and thus this upstream parent should not be used?

#### 5. Handling aggregated targets

RPL allows and defines specific procedures so as to aid target aggregation in DAO. Having said that, the specification does not mandate use of aggregated targets nor does it make any comment on whether a receiving node needs to handle it. Target aggregation is an useful tool and especially helps with link layer technologies that does not suffer from low MTUs such as PLC. Even if the implementation does not support aggregating targets, it should atleast mandate reception of aggregated targets in DAO.

RPL has a mechanism currently to ACK the DAO but it does not have a mechanism to ACK the target container. Thus in case of aggregated targets in the DAO, if the subset of the targets fail then it is impossible for the DAO-ACK to signal this to the DAO sender.

### 5.1. Deliberations

Even if the implementation does not support aggregating targets, should it atleast mandate reception and handling of aggregated targets in DAO?

There is a good scope for compressing aggregated targets which can significantly reduce the RPL control overhead.

How to selectively NACK subset of targets in case target containers are aggregated?

The DEFAULT\_DAO\_DELAY of 1sec does not help much with aggregation. The upstream parent nodes should wait for more time then the child nodes so as to effectively aggregate. Can we have DEFAULT\_DAO\_DELAY a function of the level/rank the node is at?

## 6. RPL Transit Information in DAO

RPL allows associating a target or set of targets with a Transit information container which contains attributes for a path to one or more destinations identified by the set of targets. In case of NS-MOP, the transit Information will contain the all critical Parent Address which allows the common ancestor usually the root to identify the source route header for the target node. The Transit Information also contains other information such as Path Sequence and Path Lifetime which are critical for maintaining route adjacencies.

RPL however does not mandate the use of Transit Information container for targets.

### 6.1. Deliberations

Is it ok to let implementations decide on the inclusion of Transit Information container?

Is it possible to achieve interop without mandating use of Transit Information Container?

If the Transit Information container is sent, should the handling of PathSequence be mandated?

The DEFAULT\_DAO\_DELAY of 1sec does not help much with aggregation. The upstream parent nodes should wait for more time then the child nodes so as to effectively aggregate. Can we have DEFAULT\_DAO\_DELAY a function of the level/rank the node is at?



## 7. Managing persistent variables across node reboots

### 7.1. Persistent storage and RPL state information

Devices are required to be functional for several years without manual maintenance. Usually battery power consumption is considered key for operating the devices for several (tens of) years. But apart from battery, flash memory endurance may prove to be a lifetime bottleneck in constrained networks. Endurance is defined as maximum number of erase-write cycles that a NAND/NOR cell can undergo before losing its 'gauranteed' write operation. In some cases (cheaper NAND-MLC/TLC), the endurance can be as less as 2K cycles. Thus for e.g. if a given cell is written 5 times a day, that NAND-flash cell assuming an endurance of 10K cycles may last for less than 6 years.

Wear leveling is a popular technique used in flash memory to minimize the impact of limited cell endurance. Wear leveling works by arranging data so that erasures and re-writes are distributed evenly across the medium. The memory sectors are over-provisioned so that the writes are distributed across multiple sectors. Many IoT platforms do not necessarily consider this over-provisioning and usually provision the memory only to what is required. Some scenarios such as street-lighting may not require the application layer to write any information to the persistent storage and thus the over-provisioning is often ignored. In such cases if the network stack ends up using persistent storage for maintaining its state information then it becomes counter-productive.

In a star topology, the amount of persistent data write done by network protocols is very limited. But ad-hoc networks employing routing protocols such as RPL assume certain state information to be retained across node reboots. In case of IoT devices this storage is mostly floating gate based NAND/NOR based flash memory. The impact of loss of this state information differs depending upon the type (6LN/6LR/6LBR) of the node.

### 7.2. Lollipop Counters

[RFC6550] Section 7.2. explains sequence counter operation defining lollipop [Perlman83] style counters. Lollipop counters specify mechanism in which even if the counter value wraps, the algorithm would be able to tell whether the received value is the latest or not. This mechanism also helps in "some cases" to recover from node reboot, but is not foolproof.

Consider an e.g. where Node A boots up and initialises the seqcnt to 240 as recommended in [RFC6550]. Node A communicates to Node B using this seqcnt and node B uses this seqcnt to determine whether the

information node A sent in the packet is latest. Now lets assume, the counter value reaches 250 after some operations on Node A, and node B keeps receiving updated seqcnt from node A. Now consider that node A reboots, and since it reinitializes the seqcnt value to 240 and sends the information to node B (who has seqcnt of 250 stored on behalf of node A). As per section 7.2. of [RFC6550], when node B receives this packet it will consider the information to be old (since  $240 < 250$ ).

A	B	Output
240	240	A<B, old
240	241	A<B, old
240	::	A<B, old
240	256	A<B, old
240	0	A<B, new
240	1	A>B, new
240	::	A>B, new
240	127	A>B, new

Default values for lollipop counters considered from [RFC6550]  
Section 7.2.

Table 1: Example lollipop counter operation

Based on this figure, there is dead zone (240 to 0) in which if A operates after reboot then the seqcnt will always be considered smaller. Thus node A needs to maintain the seqcnt in persistent storage and reuse this on reboot.

### 7.3. RPL State variables

The impact of loss of RPL state information differs depending upon the node type (6LN/6LR/6LBR). Following sections explain different state variables and the impact in case this information is lost on reboot.

#### 7.3.1. DODAG Version

The tuple (RPLInstanceID, DODAGID, DODAGVersionNumber) uniquely identifies a DODAG Version. DODAGVersionNumber is incremented everytime a global repair is initiated for the instance (global or local). A node receiving an older DODAGVersionNumber will ignore the DIO message assuming it to be from old DODAG version. Thus a 6LBR node (and 6LR node in case of local DODAG) needs to maintain the DODAGVersionNumber in the persistent storage, so as to be available

on reboot. In case the 6LBR could not use the latest DODAGVersionNumber the implication are that it won't be able to recover/re-establish the routing table.

#### 7.3.2. DTSN field in DIO

DTSN (Destination advertisement Trigger Sequence Number) is a DIO message field used as part of procedure to maintain Downward routes. A 6LBR/6LR node may increment a DTSN in case it requires the downstream nodes to send DAO and thus update downward routes on the 6LBR/6LR node. In case of RPL NS-MOP, only the 6LBR maintains the downward routes and thus controls this field update. In case of S-MOP, 6LRs additionally keep downward routes and thus control this field update.

In S-MOP, when a 6LR node switches parent it may have to issue a DIO with incremented DTSN to trigger downstream child nodes to send DAO so that the downward routes are established in all parent/ancestor set. Thus in S-MOP, the frequency of DTSN update might be relatively high (given the node density and hysteresis set by objective function to switch parent).

#### 7.3.3. PathSequence

PathSequence is part of RPL Transit Option, and associated with RPL Target option. A node which owns a target address can associate a PathSequence in the DAO message to denote freshness of the target information. This is especially useful when a node uses multiple paths or multiple parents to advertise its reachability.

Loss of PathSequence information maintained on the target node can result in routing adjacencies been lost on 6LRs/6LBR/6BBR.

#### 7.4. State variables update frequency

State variable	Update frequency	Impacts node type
DODAGVersionNumber	Low	6LBR, 6LR(local DODAG)
DTSN	High(SM),Low(NSM)	6LBR, 6LR
PathSequence	High(SM),Low(NSM)	6LR, 6LN

Low=<5 per day, High=>5 per day; SM=Storing MOP, NSM=Non-Storing MOP

Table 2: RPL State variables

### 7.5. Deliberations

- (1) Is it possible that RPL reduces the use of persistent storage for maintaining state information?
- (2) In most cases, the node reboots will happen very rarely. Thus doing a persistent storage book-keeping for handling node reboot might not make sense. Is it possible to consider signaling (especially after the node reboots) so as to avoid maintaining this persistent state? Is it possible to use one-time on-reboot signalling to recover some state information?
- (3) It is necessary that RPL avoids using persistent storage as far as possible. Ideally, extensions to RPL should consider this as a design requirement especially for 6LR and 6LN nodes. DTSN and PathSequence are the primary state variables which have major impact.

### 7.6. Implementation Notes

An implementation should use a random DAOSequence number on reboot so as to avoid a risk of reusing the same DAOSequence on reboot. Regardless the sequence counter size of 8bits does not provide much guarantees towards choosing a good random number. A parent node will not respond with a DAO-ACK in case it sees a DAO with the same previous DAOSequence.

Write-Before-Use: The state information should be written to the flash before using it in the messaging. If it is done the other way, then the chances are that the node power downs before writing to the persistent storage.

## 8. RPL under-specification

- (a) PathSequence: Is it mandatory to use PathSequence in DAO Transit container? RPL mentions that a 6LR/6LBR hosting the routing entry on behalf of target node should refresh the lifetime on reception of a new Path Sequence. But RPL does not necessarily mandate use of Path Sequence. Most of the open source implementation [RIOT] [CONTIKI] currently do not issue Path Sequence in the DAO message.
- (b) Target Container aggregation in DAO: RPL allows multiple targets to be aggregated in a single DAO message and has introduced a notion of DelayDAO using which a 6LR node could delay its DAO to enable such aggregation. But RPL does not have clear text on handling of aggregated DAOs and thus it hinders interoperability.

- (c) DTSN Update: RPL does not clearly define in which cases DTSN should be updated in case of storing mode of operation. More details for this are presented in Section 2.

## 9. Acknowledgements

Many thanks to Pascal Thubert for hallway chats and for helping understand the existing design rationales. Thanks to Michael Richardson for Unstrung RPL implementation rationale. Thanks to ML discussions, in particular (<https://www.ietf.org/mail-archive/web/roll/current/msg09443.html>).

## 10. IANA Considerations

This memo includes no request to IANA.

## 11. Security Considerations

This is an information draft and does not add any changes to the existing specifications.

## 12. References

### 12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <<https://www.rfc-editor.org/info/rfc6550>>.
- [RFC6551] Vasseur, JP., Ed., Kim, M., Ed., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks", RFC 6551, DOI 10.17487/RFC6551, March 2012, <<https://www.rfc-editor.org/info/rfc6551>>.
- [RFC6552] Thubert, P., Ed., "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6552, DOI 10.17487/RFC6552, March 2012, <<https://www.rfc-editor.org/info/rfc6552>>.

- [RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <<https://www.rfc-editor.org/info/rfc6775>>.
- [RFC6997] Goyal, M., Ed., Baccelli, E., Philipp, M., Brandt, A., and J. Martocci, "Reactive Discovery of Point-to-Point Routes in Low-Power and Lossy Networks", RFC 6997, DOI 10.17487/RFC6997, August 2013, <<https://www.rfc-editor.org/info/rfc6997>>.

## 12.2. Informative References

- [I-D.clausen-lln-rpl-experiences]  
Clausen, T., Verdiere, A., Yi, J., Herberg, U., and Y. Igarashi, "Observations on RPL: IPv6 Routing Protocol for Low power and Lossy Networks", draft-clausen-lln-rpl-experiences-11 (work in progress), March 2018.
- [Perlman83]  
Perlman, R., "Fault-Tolerant Broadcast of Routing Information", North-Holland Computer Networks, Vol.7, December 1983.

## Appendix A. Additional Stuff

### Authors' Addresses

Rahul Arvind Jadhav (editor)  
Huawei  
Kundalahalli Village, Whitefield,  
Bangalore, Karnataka 560037  
India

Phone: +91-080-49160700  
Email: [rahul.ietf@gmail.com](mailto:rahul.ietf@gmail.com)

Rabi Narayan Sahoo  
Huawei  
Kundalahalli Village, Whitefield,  
Bangalore, Karnataka 560037  
India

Phone: +91-080-49160700  
Email: [rabinarayans@huawei.com](mailto:rabinarayans@huawei.com)

Yuefeng Wu  
Huawei  
No.101, Software Avenue, Yuhuatai District,  
Nanjing, Jiangsu 210012  
China

Phone: +86-15251896569  
Email: wuyuefeng@huawei.com

ROLL  
Internet-Draft  
Intended status: Standards Track  
Expires: February 25, 2019

R. Jadhav, Ed.  
R. Sahoo  
Y. Wu  
Huawei  
August 24, 2018

RPL Observations  
draft-rahul-roll-rpl-observations-02

Abstract

This document describes RPL protocol design issues, various observations and possible consequences of the design and implementation choices.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://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 February 25, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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



## Table of Contents

1. Motivation . . . . .	2
2. Introduction . . . . .	3
2.1. Requirements Language and Terminology . . . . .	3
3. DTSN increment in storing MOP . . . . .	3
3.1. Deliberations . . . . .	5
4. DAO retransmission and use of DAO-ACK in storing MOP . . . . .	5
4.1. Significance of bidirectional Path establishment indication and relevance of DAO-ACK . . . . .	6
4.2. Problems with hop-by-hop DAO-ACK . . . . .	6
4.3. Problems with end-to-end DAO-ACK . . . . .	6
4.4. Deliberations . . . . .	6
4.5. Implementation Notes . . . . .	7
5. Handling resource unavailability . . . . .	7
5.1. Deliberations . . . . .	7
6. Handling aggregated targets . . . . .	7
6.1. Deliberations . . . . .	8
7. RPL Transit Information in DAO . . . . .	8
7.1. Deliberations . . . . .	8
8. Managing persistent variables across node reboots . . . . .	9
8.1. Persistent storage and RPL state information . . . . .	9
8.2. Lollipop Counters . . . . .	10
8.3. RPL State variables . . . . .	11
8.3.1. DODAG Version . . . . .	11
8.3.2. DTSN field in DIO . . . . .	11
8.3.3. PathSequence . . . . .	11
8.4. State variables update frequency . . . . .	12
8.5. Deliberations . . . . .	12
8.6. Implementation Notes . . . . .	12
9. RPL under-specification . . . . .	13
10. Acknowledgements . . . . .	13
11. IANA Considerations . . . . .	13
12. Security Considerations . . . . .	13
13. References . . . . .	13
13.1. Normative References . . . . .	13
13.2. Informative References . . . . .	14
Appendix A. Additional Stuff . . . . .	14
Authors' Addresses . . . . .	15

## 1. Motivation

The primary motivation for this draft is to enlist different issues with RPL operation and invoke a discussion within the working group. This draft by itself is not intended for RFC tracks but as a WG discussion track. This draft may in turn result in other work items taken up by the WG which may improvise on the issues mentioned herewith.

## 2. Introduction

RPL [RFC6550] specifies a proactive distance-vector routing scheme designed for LLNs (Low Power and Lossy Networks). RPL enables the network to be formed as a DODAG and supports storing mode and non-storing mode of operations. Non-storing mode allows reduced memory resource usage on the nodes by allowing non-BR nodes to operate without managing a routing table and involves use of source routing by the 6LBR to direct the traffic along a specific path. In storing mode of operation intermediate routers maintain routing tables.

This work aims to highlight various issues with RPL which makes it difficult to handle certain scenarios. This work will highlight such issues in context to RPL's mode of operations (storing versus non-storing). There are cases where RPL does not provide clear rules and implementations have to make their choices hindering interoperability and performance.

[I-D.clausen-lln-rpl-experiences] provides some interesting points. Some sections in this draft may overlap with some observations in [clausen], but this is been done to further extend some scenarios or observations. It is highly encouraged that readers should also visit [I-D.clausen-lln-rpl-experiences] for other insights. Regardless, this draft is self-sufficient in a way that it does not expect to have read [clausen-draft].

### 2.1. Requirements Language and Terminology

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

NS-MOP = RPL Non-storing Mode of Operation

S-MOP = RPL Storing Mode of Operation

This document uses terminology described in [RFC6550] and [RFC6775].

### 3. DTSN increment in storing MOP

DTSN increment has major impact on the overall RPL control traffic and on the efficiency of downstream route update. DTSN is sent as part of DIO message and signals the downstream nodes to trigger the target advertisement. The 6LR needs to decide when to update the DTSN and usually it should do it in a conservative way. The DTSN update mechanism determines how soon the downward routes are established along the new path. RPL specifications does not provide

any clear mechanism on how the DTSN update should happen in case of storing mode.

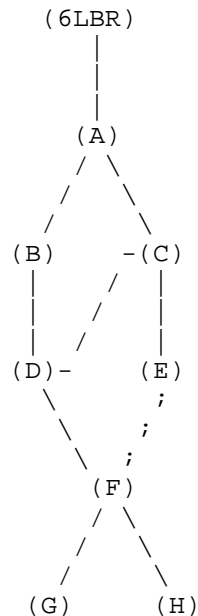


Figure 1: Sample topology

Consider example topology shown in Figure 1, assume that node D switches the parent from node B to C. Ideally the downstream nodes D and its sub-children should send their target advertisement to the new path via node C. To achieve this result in an efficient way is a challenge. Incrementing DTSN is the only way to trigger the DAO on downstream nodes. But this trigger should be sent not only on the first hop but to all the grand-child nodes. Thus DTSN has to be incremented in the complete sub-DODAG rooted at node D thus resulting in DIO/DAO storm along the sub-DODAG. This is specifically a big issue in high density networks where the metric deterioration might happen transiently even though the signal strength is good.

The primary implementation issue is whether a child node increments its own DTSN when it receives DTSN update from its parent node? This would result in DAO-updates in the sub-DODAG, thus the cost could be very high. If not incremented it may result in serious loss of connectivity for nodes in the sub-DODAG.

### 3.1. Deliberations

- (1) In S-MOP, should the child nodes increment its DIO on seeing that its preferred parent has updated its DTSN?
- (2) What are rules for DTSN increment for storing MOP, which multiple implementations can follow thus allowing consistent performance across different implementations?

### 4. DAO retransmission and use of DAO-ACK in storing MOP

[RFC6550] has an optional DAO-ACK mechanism using which an upstream parent confirms the reception of a DAO from the downstream child. In case of storing mode, the DAO is addressed to the immediate hop upstream parent resulting in DAO-ACK from the parent. There are two implementations possible:

- (1) Hop-by-hop ACK: A parent responds with a DAO-ACK immediately after receiving the DAO.
- (2) End-to-End ACK: A node waits for the upstream parent to send DAO-ACK to respond with a DAO-ACK downstream. The upstream parent may do as many attempts to successfully send this DAO upstream. In other words, the parent node accepts the responsibility of sending the DAO upstream till the point it is ACKed the moment it responds back with its own ACK to the child.

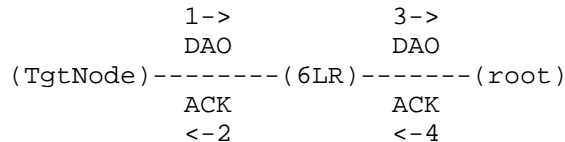


Figure 2: Hop-by-hop DAO-ACK

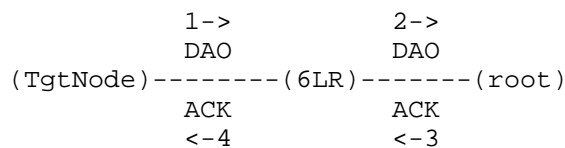


Figure 3: End-to-End DAO-ACK

#### 4.1. Significance of bidirectional Path establishment indication and relevance of DAO-ACK

Lot of application traffic patterns requires that the bidirectional path be established between the target node and the root. A typical example is that COAP request with ACK bit set would require an acknowledgement from the end receiver and thus warrants bidirectional path establishment. It is imperative that the target node first ascertains whether such a bidirectional path is established before initiating such application traffic. In case of non-storing MOP, the DAO-ACK works perfectly fine to ascertain such bidirectional connectivity since it is an indication that the root which usually is the direct destination of the DAO has received the DAO. But in case of storing MOP, things are more complicated since DAO is sent hop-by-hop and the DAO-ACK semantics are not clear enough as per the current specification. As mentioned in above section, an implementation can choose to implement hop-by-hop ACK or end-to-end ACK.

#### 4.2. Problems with hop-by-hop DAO-ACK

The primary issue with this mode is that target node cannot ascertain bidirection path connectivity on the reception of the DAO-ACK.

#### 4.3. Problems with end-to-end DAO-ACK

In this case, it is possible for the target node to ascertain if the DAO has indeed reached the root since the reception of DAO-ACK on target node confirms this. However there is extra state information that needs to be maintained on the 6LRs on behalf of all the child nodes. Also it is very difficult for the target node to ascertain a timer value to decide whether the DAO transmission has failed to reach the root.

#### 4.4. Deliberations

- (1) How should an implementation interpret the DAO-ACK semantics?
- (2) What is the best way for the target node to know that the end to end bidirectional path is successfully installed or updated? In NS-MOP, the DAO-ACK provides a clear way to do this. Can the same be achieved for storing-MOP?
- (3) What happens if the DAO-ACK with Status!=0 is responded by ancestor node?
- (4) How to selectively NACK subset of targets in case target containers are aggregated?

#### 4.5. Implementation Notes

Current RPL open source implementations have both types of DAO-ACK implementations. For e.g. RIOT supports hop-by-hop DAO-ACK. Contiki older versions supported hop-by-hop ACK but the recent version have changed to end-to-end ACK implementation.

The sequence of sending no-path DAO and DAO matters when updating the routing adjacencies on a parent switch. If an implementation chooses to send no-path DAO before DAO then it results in significantly more overhead for route invalidation. This is because no-path DAO would traverse all the way up to the BR clearing the routes on the way. In case there is a common ancestor post which the old and new path remains same then it is better to send regular DAO first thus limiting the propagation of subsequent no-path DAO till this common ancestor.

#### 5. Handling resource unavailability

The nodes in the constrained networks have to maintain various records such as neighbor cache entries and routing entries on behalf of other targets to facilitate packet forwarding. Because of the constrained nature of the devices the memory available may be very limited and thus the path selection algorithm may have to take into consideration such resource constraints as well.

RPL currently does not have any mechanism to advertise such resource indicator metrics. The primary tables associated with RPL are routing table and the neighbor cache. Even though neighbor cache is not directly linked with RPL protocol, the maintenance of routing adjacencies results in updates to neighbor cache.

##### 5.1. Deliberations

Is it possible to know that an upstream parent/ancestor cannot hold enough routing entries and thus this path should not be used?

Is it possible to know that an upstream parent cannot hold any more neighbor cache entry and thus this upstream parent should not be used?

#### 6. Handling aggregated targets

RPL allows and defines specific procedures so as to aid target aggregation in DAO. Having said that, the specification does not mandate use of aggregated targets nor does it make any comment on whether a receiving node needs to handle it. Target aggregation is an useful tool and especially helps with link layer technologies that

does not suffer from low MTUs such as PLC. Even if the implementation does not support aggregating targets, it should at least mandate reception of aggregated targets in DAO.

RPL has a mechanism currently to ACK the DAO but it does not have a mechanism to ACK the target container. Thus in case of aggregated targets in the DAO, if the subset of the targets fail then it is impossible for the DAO-ACK to signal this to the DAO sender.

#### 6.1. Deliberations

Even if the implementation does not support aggregating targets, should it at least mandate reception and handling of aggregated targets in DAO?

There is a good scope for compressing aggregated targets which can significantly reduce the RPL control overhead.

How to selectively NACK subset of targets in case target containers are aggregated?

The DEFAULT\_DAO\_DELAY of 1sec does not help much with aggregation. The upstream parent nodes should wait for more time then the child nodes so as to effectively aggregate. Can we have DEFAULT\_DAO\_DELAY a function of the level/rank the node is at?

#### 7. RPL Transit Information in DAO

RPL allows associating a target or set of targets with a Transit information container which contains attributes for a path to one or more destinations identified by the set of targets. In case of NS-MOP, the transit Information will contain the all critical Parent Address which allows the common ancestor usually the root to identify the source route header for the target node. The Transit Information also contains other information such as Path Sequence and Path Lifetime which are critical for maintaining route adjacencies.

RPL however does not mandate the use of Transit Information container for targets.

#### 7.1. Deliberations

Is it ok to let implementations decide on the inclusion of Transit Information container?

Is it possible to achieve interop without mandating use of Transit Information Container?

If the Transit Information container is sent, should the handling of PathSequence be mandated?

The DEFAULT\_DAO\_DELAY of 1sec does not help much with aggregation. The upstream parent nodes should wait for more time than the child nodes so as to effectively aggregate. Can we have DEFAULT\_DAO\_DELAY a function of the level/rank the node is at?

## 8. Managing persistent variables across node reboots

### 8.1. Persistent storage and RPL state information

Devices are required to be functional for several years without manual maintenance. Usually battery power consumption is considered key for operating the devices for several (tens of) years. But apart from battery, flash memory endurance may prove to be a lifetime bottleneck in constrained networks. Endurance is defined as maximum number of erase-write cycles that a NAND/NOR cell can undergo before losing its 'gauranteed' write operation. In some cases (cheaper NAND-MLC/TLC), the endurance can be as less as 2K cycles. Thus for e.g. if a given cell is written 5 times a day, that NAND-flash cell assuming an endurance of 10K cycles may last for less than 6 years.

Wear leveling is a popular technique used in flash memory to minimize the impact of limited cell endurance. Wear leveling works by arranging data so that erasures and re-writes are distributed evenly across the medium. The memory sectors are over-provisioned so that the writes are distributed across multiple sectors. Many IoT platforms do not necessarily consider this over-provisioning and usually provision the memory only to what is required. Some scenarios such as street-lighting may not require the application layer to write any information to the persistent storage and thus the over-provisioning is often ignored. In such cases if the network stack ends up using persistent storage for maintaining its state information then it becomes counter-productive.

In a star topology, the amount of persistent data write done by network protocols is very limited. But ad-hoc networks employing routing protocols such as RPL assume certain state information to be retained across node reboots. In case of IoT devices this storage is mostly floating gate based NAND/NOR based flash memory. The impact of loss of this state information differs depending upon the type (6LN/6LR/6LBR) of the node.



## 8.2. Lollipop Counters

[RFC6550] Section 7.2. explains sequence counter operation defining lollipop [Perlman83] style counters. Lollipop counters specify mechanism in which even if the counter value wraps, the algorithm would be able to tell whether the received value is the latest or not. This mechanism also helps in "some cases" to recover from node reboot, but is not foolproof.

Consider an e.g. where Node A boots up and initialises the seqcnt to 240 as recommended in [RFC6550]. Node A communicates to Node B using this seqcnt and node B uses this seqcnt to determine whether the information node A sent in the packet is latest. Now lets assume, the counter value reaches 250 after some operations on Node A, and node B keeps receiving updated seqcnt from node A. Now consider that node A reboots, and since it reinitializes the seqcnt value to 240 and sends the information to node B (who has seqcnt of 250 stored on behalf of node A). As per section 7.2. of [RFC6550], when node B receives this packet it will consider the information to be old (since  $240 < 250$ ).

A	B	Output
240	240	A<B, old
240	241	A<B, old
240	::	A<B, old
240	256	A<B, old
240	0	A<B, new
240	1	A>B, new
240	::	A>B, new
240	127	A>B, new

Default values for lollipop counters considered from [RFC6550]  
Section 7.2.

Table 1: Example lollipop counter operation

Based on this figure, there is dead zone (240 to 0) in which if A operates after reboot then the seqcnt will always be considered smaller. Thus node A needs to maintain the seqcnt in persistent storage and reuse this on reboot.

### 8.3. RPL State variables

The impact of loss of RPL state information differs depending upon the node type (6LN/6LR/6LBR). Following sections explain different state variables and the impact in case this information is lost on reboot.

#### 8.3.1. DODAG Version

The tuple (RPLInstanceID, DODAGID, DODAGVersionNumber) uniquely identifies a DODAG Version. DODAGVersionNumber is incremented everytime a global repair is initiated for the instance (global or local). A node receiving an older DODAGVersionNumber will ignore the DIO message assuming it to be from old DODAG version. Thus a 6LBR node (and 6LR node in case of local DODAG) needs to maintain the DODAGVersionNumber in the persistent storage, so as to be available on reboot. In case the 6LBR could not use the latest DODAGVersionNumber the implication are that it won't be able to recover/re-establish the routing table.

#### 8.3.2. DTSN field in DIO

DTSN (Destination advertisement Trigger Sequence Number) is a DIO message field used as part of procedure to maintain Downward routes. A 6LBR/6LR node may increment a DTSN in case it requires the downstream nodes to send DAO and thus update downward routes on the 6LBR/6LR node. In case of RPL NS-MOP, only the 6LBR maintains the downward routes and thus controls this field update. In case of S-MOP, 6LRs additionally keep downward routes and thus control this field update.

In S-MOP, when a 6LR node switches parent it may have to issue a DIO with incremented DTSN to trigger downstream child nodes to send DAO so that the downward routes are established in all parent/ancestor set. Thus in S-MOP, the frequency of DTSN update might be relatively high (given the node density and hysteresis set by objective function to switch parent).

#### 8.3.3. PathSequence

PathSequence is part of RPL Transit Option, and associated with RPL Target option. A node whichs owns a target address can associate a PathSequence in the DAO message to denote freshness of the target information. This is especially useful when a node uses multiple paths or multiple parents to advertise its reachability.

Loss of PathSequence information maintained on the target node can result in routing adjacencies been lost on 6LRs/6LBR/6BBR.

## 8.4. State variables update frequency

State variable	Update frequency	Impacts node type
DODAGVersionNumber	Low	6LBR, 6LR(local DODAG)
DTSN	High(SM),Low(NSM)	6LBR, 6LR
PathSequence	High(SM),Low(NSM)	6LR, 6LN

Low=<5 per day, High=>5 per day; SM=Storing MOP, NSM=Non-Storing MOP

Table 2: RPL State variables

## 8.5. Deliberations

- (1) Is it possible that RPL reduces the use of persistent storage for maintaining state information?
- (2) In most cases, the node reboots will happen very rarely. Thus doing a persistent storage book-keeping for handling node reboot might not make sense. Is it possible to consider signaling (especially after the node reboots) so as to avoid maintaining this persistent state? Is it possible to use one-time on-reboot signalling to recover some state information?
- (3) It is necessary that RPL avoids using persistent storage as far as possible. Ideally, extensions to RPL should consider this as a design requirement especially for 6LR and 6LN nodes. DTSN and PathSequence are the primary state variables which have major impact.

## 8.6. Implementation Notes

An implementation should use a random DAOSequence number on reboot so as to avoid a risk of reusing the same DAOSequence on reboot. Regardless the sequence counter size of 8bits does not provide much gurantees towards choosing a good random number. A parent node will not respond with a DAO-ACK in case it sees a DAO with the same previous DAOSequence.

Write-Before-Use: The state information should be written to the flash before using it in the messaging. If it is done the other way, then the chances are that the node power downs before writing to the persistent storage.

## 9. RPL under-specification

- (a) PathSequence: Is it mandatory to use PathSequence in DAO Transit container? RPL mentions that a 6LR/6LBR hosting the routing entry on behalf of target node should refresh the lifetime on reception of a new Path Sequence. But RPL does not necessarily mandate use of Path Sequence. Most of the open source implementation [RIOT] [CONTIKI] currently do not issue Path Sequence in the DAO message.
- (b) Target Container aggregation in DAO: RPL allows multiple targets to be aggregated in a single DAO message and has introduced a notion of DelayDAO using which a 6LR node could delay its DAO to enable such aggregation. But RPL does not have clear text on handling of aggregated DAOs and thus it hinders interoperability.
- (c) DTSN Update: RPL does not clearly define in which cases DTSN should be updated in case of storing mode of operation. More details for this are presented in Section 3.

## 10. Acknowledgements

Many thanks to Pascal Thubert for hallway chats and for helping understand the existing design rationales. Thanks to Michael Richardson for Unstrung RPL implementation rationale. Thanks to ML discussions, in particular (<https://www.ietf.org/mail-archive/web/roll/current/msg09443.html>).

## 11. IANA Considerations

This memo includes no request to IANA.

## 12. Security Considerations

This is an information draft and does add any changes to the existing specifications.

## 13. References

### 13.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

- [RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <<https://www.rfc-editor.org/info/rfc6550>>.
- [RFC6551] Vasseur, JP., Ed., Kim, M., Ed., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks", RFC 6551, DOI 10.17487/RFC6551, March 2012, <<https://www.rfc-editor.org/info/rfc6551>>.
- [RFC6552] Thubert, P., Ed., "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6552, DOI 10.17487/RFC6552, March 2012, <<https://www.rfc-editor.org/info/rfc6552>>.
- [RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <<https://www.rfc-editor.org/info/rfc6775>>.
- [RFC6997] Goyal, M., Ed., Baccelli, E., Philipp, M., Brandt, A., and J. Martocci, "Reactive Discovery of Point-to-Point Routes in Low-Power and Lossy Networks", RFC 6997, DOI 10.17487/RFC6997, August 2013, <<https://www.rfc-editor.org/info/rfc6997>>.

### 13.2. Informative References

- [I-D.clausen-lln-rpl-experiences]  
Clausen, T., Verdiere, A., Yi, J., Herberg, U., and Y. Igarashi, "Observations on RPL: IPv6 Routing Protocol for Low power and Lossy Networks", draft-clausen-lln-rpl-experiences-11 (work in progress), March 2018.
- [Perlman83]  
Perlman, R., "Fault-Tolerant Broadcast of Routing Information", North-Holland Computer Networks, Vol.7, December 1983.

### Appendix A. Additional Stuff

Authors' Addresses

Rahul Arvind Jadhav (editor)  
Huawei  
Kundalahalli Village, Whitefield,  
Bangalore, Karnataka 560037  
India

Phone: +91-080-49160700  
Email: rahul.ietf@gmail.com

Rabi Narayan Sahoo  
Huawei  
Kundalahalli Village, Whitefield,  
Bangalore, Karnataka 560037  
India

Phone: +91-080-49160700  
Email: rabinarayans@huawei.com

Yuefeng Wu  
Huawei  
No.101, Software Avenue, Yuhuatai District,  
Nanjing, Jiangsu 210012  
China

Phone: +86-15251896569  
Email: wuyuefeng@huawei.com

ROLL  
Internet-Draft  
Updates: 6550, 6775 (if approved)  
Intended status: Standards Track  
Expires: November 24, 2018

P. Thubert, Ed.  
Cisco  
May 23, 2018

Routing for RPL Leaves  
draft-thubert-roll-unaware-leaves-05

Abstract

This specification updates RFC 6550 and RFC 6775 unicast routing service in a RPL domain to 6LoWPAN ND nodes that do not participate to the routing protocol.

Status of This Memo

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

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

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

This Internet-Draft will expire on November 24, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	3
2.1. BCP 14 . . . . .	3
2.2. References . . . . .	4
2.3. Subset of a 6LoWPAN Glossary . . . . .	5
3. 6LoWPAN Neighbor Discovery . . . . .	6
4. Updating RFC 6550 . . . . .	7
5. Updating RFC 6775 Update . . . . .	7
6. Dependencies on the 6LN . . . . .	8
7. Protocol Operations . . . . .	8
7.1. General Flow . . . . .	8
7.2. 6LN Operation . . . . .	10
7.3. 6LR Operation . . . . .	11
7.4. RPL Root Operation . . . . .	12
7.5. 6LBR Operation . . . . .	13
8. Implementation Status . . . . .	14
9. Security Considerations . . . . .	14
10. IANA Considerations . . . . .	14
11. Acknowledgments . . . . .	14
12. References . . . . .	14
12.1. Normative References . . . . .	14
12.2. Informative References . . . . .	16
Author's Address . . . . .	16

## 1. Introduction

The design of Low Power and Lossy Networks (LLNs) is generally focused on saving energy, which is the most constrained resource of all. Other design constraints, such as a limited memory capacity, duty cycling of the LLN devices and low-power lossy transmissions, derive from that primary concern.

The IETF produced the "Routing Protocol for Low Power and Lossy Networks" [RFC6550] (RPL) to provide routing services within such constraints. RPL is a Distance-Vector protocol, which, compared to link-state protocols, limits the amount of topological knowledge that needs to be installed and maintained in each node. In order to operate in constrained networks, RPL allows a Routing Stretch (see [RFC6687]), whereby routing is only performed along a DODAG as opposed to straight along a shortest path between 2 peers, whatever that would mean in a given LLN. This trades the quality of peer-to-peer (P2P) paths for a vastly reduced amount of control traffic and routing state that would be required to operate a any-to-any shortest path protocol. Finally, broken routes may be fixed lazily and on-demand, based on dataplane inconsistency discovery, which avoids wasting energy in the proactive repair of unused paths.



In order to cope with lossy transmissions, RPL forms Direction-Oriented Directed Acyclic Graphs (DODAGs) using DODAG Information Solicitation (DIS) and DODAG Information Object (DIO) messages. For most of the nodes, though not all, a DODAG provides multiple forwarding solutions towards the Root of the topology via so-called parents. RPL is designed to adapt to fuzzy connectivity, whereby the physical topology cannot be expected to reach a stable state, with a lazy control that creates routes proactively but only fixes them when they are used by actual traffic. It results that RPL provides reachability for most of the LLN nodes, most of the time, but does not really converge in the classical sense. RPL provides unicast and multicast routing services back to RPL-Aware nodes (RANs). A RAN will inject routes to self using Destination Advertisement Object (DAO) messages sent to either their parents in Storing Mode or to the Root indicating their parent in Non-Storing mode. This process effectively forms a DODAG back to the device that is a subset of the DODAG to the Root with all links reversed.

When a routing protocol such as RPL is used to maintain reachability within a Non-Broadcast Multi-Access (NBMA) subnet, some nodes may act as routers and participate to the routing operations whereas others may be plain hosts. In RPL terms, a plain host that does not participate to the routing protocol is called a Leaf. It must be noted that a 6LN could participate to RPL and inject DAO routes to self, but refrain from advertising DIO and get children. In that case, the 6LN is still a host but not a Leaf.

This specification enables a RPL-Unaware Leaf (RUL) to announce itself as a host and demand that the 6LR that accepts the registration also inject the relevant routing information for the Registered Address in the RPL domain on its behalf. The packet forwarding operation by the 6LR serving a Leaf 6LN is described in "When to use RFC 6553, 6554 and IPv6-in-IPv6" [I-D.ietf-roll-useofrplinfo]. This document adds the capability by a 6LR to advertise the IPv6 address(es) of the 6LN in the RPL protocol. Examples of routing-agnostic 6LN may include lightly-powered sensors such as window smash sensor (alarm system), or the kinetically powered light switch.

## 2. Terminology

### 2.1. BCP 14

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 BCP 14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

## 2.2. References

The Terminology used in this document is consistent with and incorporates that described in Terms Used in Routing for Low-Power and Lossy Networks (LLNs). [RFC7102].

Other terms in use in LLNs are found in Terminology for Constrained-Node Networks [RFC7228].

A glossary of classical 6LoWPAN acronyms is given in Section 2.3.

The term "byte" is used in its now customary sense as a synonym for "octet".

"RPL", "RPL Packet Information" (RPI) and "RPL Instance", DIO, DAO and DIS messages are defined in the "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks" [RFC6550] specification.

This document introduces the term RPL-Unaware Leaf (RUL) to refer to a node that uses a RPL router (without necessarily knowing it) as 6LR and depends on that router to obtain reachability for its addresses inside the RPL domain. On the contrary, the term RPL-Aware Leaf (RAL) is used to refer to a host or a router that participates to RPL and advertises its addresses of prefixes by itself.

Other terms in use in LLNs are found in Terminology for Constrained-Node Networks [RFC7228].

Readers are expected to be familiar with all the terms and concepts that are discussed in

- o "Neighbor Discovery for IP version 6" [RFC4861],
- o "IPv6 Stateless Address Autoconfiguration" [RFC4862],
- o "Problem Statement and Requirements for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing" [RFC6606],
- o "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals" [RFC4919],
- o "Neighbor Discovery Optimization for Low-power and Lossy Networks" [RFC6775], and
- o "Registration Extensions for 6LoWPAN Neighbor Discovery" [I-D.ietf-6lo-rfc6775-update].

### 2.3. Subset of a 6LoWPAN Glossary

This document often uses the following acronyms:

6BBR: 6LoWPAN Backbone Router (proxy for the registration)

6LBR: 6LoWPAN Border Router (authoritative on DAD)

6LN: 6LoWPAN Node

6LR: 6LoWPAN Router (relay to the registration process)

6CIO: Capability Indication Option

(E)ARO: (Extended) Address Registration Option

(E)DAR: (Extended) Duplicate Address Request

(E)DAC: (Extended) Duplicate Address Confirmation

DAD: Duplicate Address Detection

DODAG: Destination-Oriented Directed Acyclic Graph

LLN: Low-Power and Lossy Network (a typical IoT network)

NA: Neighbor Advertisement

NCE: Neighbor Cache Entry

ND: Neighbor Discovery

NDP: Neighbor Discovery Protocol

NS: Neighbor Solicitation

ROVR: Registration Ownership Verifier (pronounced rover)

RPL: IPv6 Routing Protocol for LLNs (pronounced ripple)

RA: Router Advertisement

RS: Router Solicitation

TSCH: Timeslotted Channel Hopping

TID: Transaction ID (a sequence counter in the EARO)

### 3. 6LoWPAN Neighbor Discovery

The IPv6 [RFC8200]Neighbor Discovery (IPv6 ND) Protocol (NDP) suite [RFC4861] [RFC4862] defined for fast media such a Ethernet, relies heavily on multicast operations for address discovery and duplicate address detection (DAD).

"Neighbor Discovery Optimizations for 6LoWPAN networks" [RFC6775] (6LoWPAN ND) adapts IPv6 ND for operations over energy-constrained LLNs. In particular, 6LoWPAN ND introduces a unicast host address registration mechanism that contributes to reduce the use of multicast messages that are present in the classical IPv6 ND protocol. 6LoWPAN ND defines a new Address Registration Option (ARO) that is carried in the unicast Neighbor Solicitation (NS) and Neighbor Advertisement (NA) messages between the 6LoWPAN Node (6LN) and the 6LoWPAN Router (6LR). 6LoWPAN ND also defines the Duplicate Address Request (DAR) and Duplicate Address Confirmation (DAC) messages between the 6LR and the 6LoWPAN Border Router (6LBR). In an LLN, the 6LBR is the central repository of all the Registered Addresses in its domain.

"Registration Extensions for 6LoWPAN Neighbor Discovery" [I-D.ietf-6lo-rfc6775-update] defines an Extended ARO (EARO). The format of the EARO is shown in Figure 1:

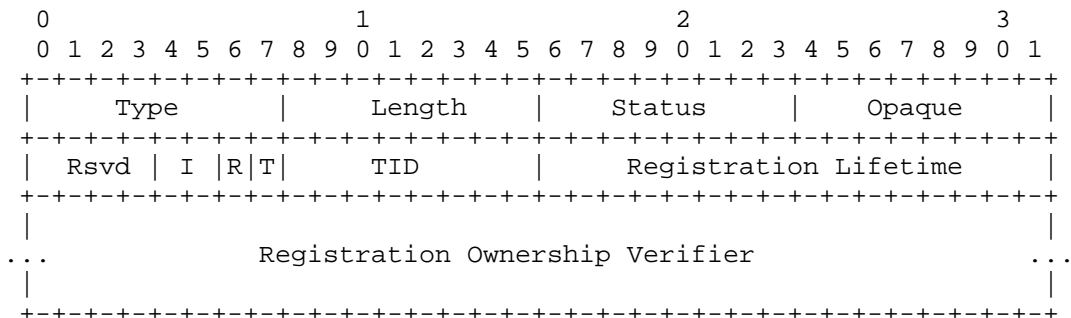


Figure 1: EARO Option Format

The 'R' flag that is set if the Registering Node expects that the 6LR ensures reachability for the Registered Address, e.g., by means of routing or proxying ND.

The EARO also includes a sequence counter called Transaction ID (TID), which maps to the Path Sequence Field found in Transit Options in RPL DAO messages. It is a prerequisite for this specification.

Finally, the EARO transports an Opaque field and an 'I' field that describes what the Opaque field transports and how to use it. This specification requires that the I field is left to 0 and to use the Opaque field to carry the RPL InstanceID if one is known, else to leave the Opaque field to zero.

#### 4. Updating RFC 6550

This document specifies a new behavior whereby a 6LR injects DAO messages for unicast addresses registered through the updated 6LoWPAN ND [I-D.ietf-6lo-rfc6775-update] on behalf of 6LN nodes that are not RPL-aware.

Upon the renewal of a 6lowPAN ND registration, this specification changes the behavior of the 6LR as follows. If the 'R' flag is set, the 6LR injects a DAO targeting the Registered Address, and refrains from sending a DAR message. the DAR/DAC exchange that refreshes the state in the 6LBR happens instead between the RPL Root and the 6LBR. In that flow, the RPL Root acts as a proxy on behalf of the 6LR upon the reception of the DAO propagation initiated at the 6LR.

#### 5. Updating RFC 6775 Update

The behavior defined in this specification whereby the 6LR that processes the registration advertises the Registered Address in DAO messages and bypasses the DAR/DAC process for the renewal of a registration, is only triggered by an NS(EARO) that has the 'R' flag set. If the 'R' flag is not set, then the Registering Node is expected to be a RAN router that handles the reachability of the Registered Address by itself.

This document also specifies a keep-alive EDAR message that the RPL Root may use to maintain an existing state in the 6LBR upon receiving DAO messages. The keep-alive EDAR message may only act as a refresher and can only update the Lifetime and the TID of the state in the 6LBR.

This document similarly specifies a keep-alive NS(EARO) message that the RPL Root may use to maintain an existing state in a 6BBR upon receiving DAO messages. The keep-alive NS(EARO) message may only act as a refresher and can only update the Lifetime and the TID of the state in the 6BBR.

As prescribed by [I-D.ietf-6lo-rfc6775-update], a RPL router SHOULD NOT set the 'R' flag.

## 6. Dependencies on the 6LN

This document provides RPL routing for a 6LN acting as a plain host and not aware of RPL. Still, a minimal RPL-independent functionality is expected from the 6LN in order to operate properly as a RLU; in particular:

- o the 6LN MUST implement [I-D.ietf-6lo-rfc6775-update] and set the 'R' flag in the EARO option. The 'R' flag is used to determine whether the Registering Node is a RUL, not aware of the RPL operation in the network, and thus does not participate to it. A 6LN is considered to be a RUL if and only if it sets the 'R' flag in the EARO.
- o RPL data packets typically carry a Hop-by-Hop Header to transport a RPL Packet Information (RPI) [RFC6550]. The 6LN MUST ignore the RPI and skip the HbH header.
- o RPL data packets are often encapsulated using IP in IP. The 6LN MUST be able to decapsulate a packet when it is the destination of the outer header and process correctly the inner header.

## 7. Protocol Operations

### 7.1. General Flow

This specification enables to save the exchange of Extended Duplicate Address messages, EDAR and EDAC, from a 6LN all the way to the 6LBR across a RPL mesh, for the sole purpose of refreshing an existing state in the 6LBR. Instead, the EDAR/EDAC exchange is proxied by the RPL Root upon a DAO message that refreshes the RPL routing state. To achieve this, the lifetimes and sequence counters in 6LoWPAN ND and RPL are aligned. In other words, the Path Sequence and the Path Lifetime in the DAO message are derived from the Transaction ID and the registration lifetime in the NS(EARO) message from the 6LN.

From the perspective of the 6LN, the registration flow happens transparently; it is not delayed by the proxy RPL operation, so the device does not need to wait more whether RPL proxy operation happens or not. The flows below are RPL Non-Storing Mode examples. In Storing Mode, the DAO ACK may not be present, and the DAO messages cascade from child to parent all the way to the DODAG Root.

On the first registration, illustrated in Figure 2, from the perspective of the 6LR, the Extended Duplicate Address message takes place as prescribed by [I-D.ietf-6lo-rfc6775-update]. When successful, the flow creates a Neighbor Cache Entry (NCE) in the 6LR, and the 6LR injects the Registered Address in RPL using DAO/DAO-ACK

exchanges all the way to the RPL DODAG Root. The protocol does not carry a specific information that the Extended Duplicate Address messages were already exchanged, so the Root proxies them anyway.

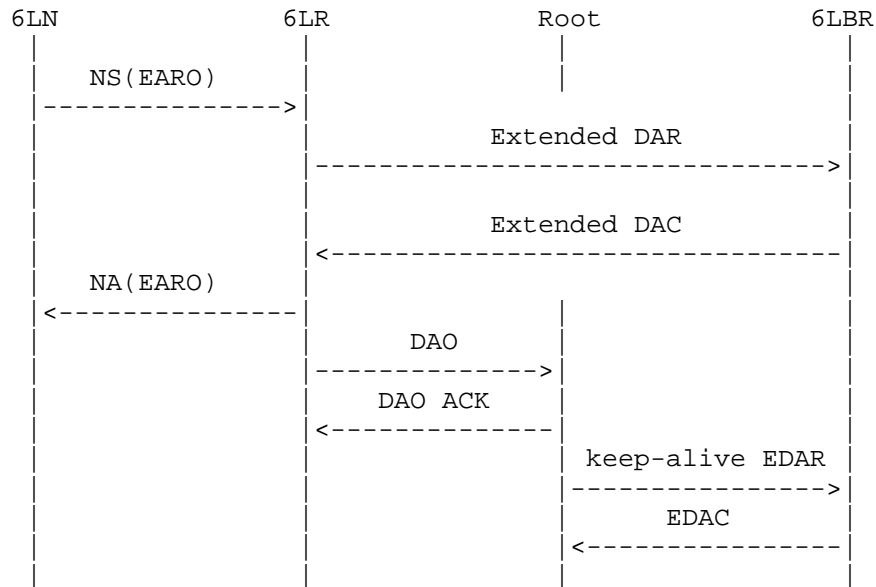


Figure 2: First Registration Flow

A re-registration is performed by the 6LN to maintain the NCE in the 6LR alive before lifetime expires. Upon a re-registration, as illustrated in Figure 2, the 6LR redistributes the Registered Address NS(EARO) in RPL. This causes the RPL DODAG Root to refresh the state in the 6LBR with a keep-alive EDAC message. The keep-alive EDAC lacks the Registration Ownership Verifier (ROVR) information, since it is not present in RPL DAO messages, but the EDAC message sent in response by the 6LBR contains the actual value of the ROVR field for that registration. This enables the RPL Root to perform the proxy-registration for the Registered Address and attract traffic captured over the backbone by the 6BBR and route it back to the device.

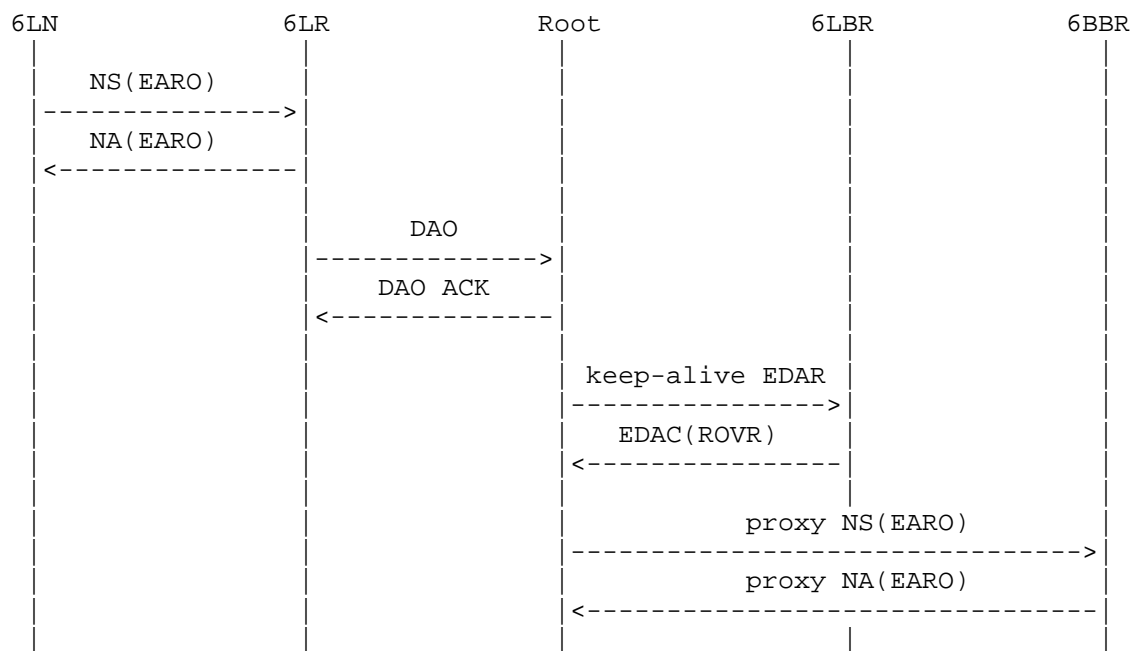


Figure 3: Next Registration Flow

Note that any of the functions 6LR, Root and 6LBR might be collapsed in a single node, in which case the flow above happens internally, and possibly through internal API calls as opposed to messaging.

## 7.2. 6LN Operation

This specification does not alter the operation of a 6LowPAN ND-compliant 6LN, which is expected to operate as follows:

- o The 6LN obtains an IPv6 global address, for instance using autoconfiguration [RFC4862] based on a Prefix Information Option (PIO) [RFC4861] found in a Router Advertisement message or by some other means such as DHCPv6 [RFC3315].
- o Once it has formed an address, the 6LN (re)registers its address periodically, within the Lifetime of the previous registration, as prescribed by [I-D.ietf-6lo-rfc6775-update].
- o Upon each consecutive registration, the 6LN MUST increase the TID field.
- o If the 6LN is aware of the RPL Instance the packet should be injected into, then it SHOULD set the Opaque field to the



InstanceID, else it MUST leave the Opaque field to zero. In any fashion the 6LN MUST set the 'I' field to zero.

- o A 6LN acting as a RUL MUST set the 'R' flag in the EARO whereas a 6LN acting as a RAN SHOULD NOT set the 'R' flag.
- o The 6LN MAY register to more than one 6LR at the same time. In that case, a same value of TID is used for each registration.
- o The 6LN MAY use any of the 6LRs to which it register to forward its packets.

### 7.3. 6LR Operation

Also as prescribed by [I-D.ietf-6lo-rfc6775-update], the 6LR generates a DAR message upon reception of a valid NS(EARO) message for the registration of a new IPv6 Address by a 6LN. If the Duplicate Address exchange succeeds, then the 6LR installs a Neighbor Cache Entry (NCE). If the 'R' flag was set in the EARO of the NS message, and this 6LR can manage the reachability of Registered Address, then the 6LR sets the 'R' flag in the ARO of the response NA message.

From then on, the 6LN periodically sends a new NS(EARO) to refresh the NCE state before the lifetime indicated in the EARO expires, with TID that is incremented each time till it wraps in a lollipop fashion. As long as the 'R' flag is set and this router can still manage the reachability of Registered Address, the 6LR keeps setting the 'R' flag in the EARO of the response NA message, but the exchange of Extended Duplicate Address messages is skipped.

The Opaque field in the EARO hints the 6LR on the RPL Instance that should be used for the DAO advertisements, and for the forwarding of packets sourced at the registered address when there is no RPL Packet Information (RPI) in the packet, in which case the 6LR SHOULD add one to the packet. if the 'I' field is not zero, then the 6LR MUST consider that the Opaque field is left to zero. If the Opaque field is not set to zero, then it should carry a RPL InstanceID for the Instance suggested by the 6LN. If the 6LR does not participate to the associated Instance, then the 6LR MUST consider that the Opaque field is left to zero. If the Opaque field left to zero, the 6LR is free to use the default Instance (zero) for the registered address or to select an Instance of its choice; else, that is if the 6LR participates to the suggested Instance, then the 6LR SHOULD use that Instance for the registered address.

Upon a successful NS/NA(EARO) exchange: if the 'R' flag was set in the EARO of the NS message, then the 6LR SHOULD inject the Registered

Address in RPL by sending a DAO message on behalf of the 6LN; else the 6LR MUST NOT inject the Registered Address into RPL.

The DAO message advertising the Registered Address MUST be constructed as follows:

- o The Registered Address is placed in a RPL Target Option in the DAO message as the Target Prefix, and the Prefix Length is set to 128
- o the External 'E' flag in the Transit Information Option (TIO) associated to the Target Option is set to indicate that the 6LR redistributes an external target into the RPL network
- o the Path Lifetime in the TIO is computed from the Lifetime in the EARO Option to adapt it to the Lifetime Units used in the RPL operation. Note that if the lifetime is 0, then the 6LR generates a No-Path DAO message that cleans up the routes down to the Address of the 6LN.
- o the Path Sequence in the TIO is set to the TID value found in the EARO option.
- o Additionally, in Non-Storing Mode the 6LR indicates one of its global IPv6 unicast addresses as the Parent Address in the TIO.

If a 6LR receives a valid NS(EARO) message with the 'R' flag reset and the 6LR was redistributing the Registered Address due to previous NS(EARO) messages with the flag set, then it MUST stop injecting the address. It is up to the Registering Node to maintain the corresponding route from then on, either keeping it active by sending further DAO messages, or destroying it using a No-Path DAO.

#### 7.4. RPL Root Operation

In RPL Storing Mode of Operation (MOP), the DAO message is propagated from child to parent all the way to the Root along the DODAG, populating routing state as it goes. In Non-Storing Mode, The DAO message is sent directly to the route. Upon reception of a DAO message that creates or updates an existing RPL state:

- o the Root notifies the 6LBR using an internal API if they are collocated, or performs a keep-alive DAR/DAC exchange on behalf of the registering node if they are separated.
- o In an extended topology with a Backbone Link, the Root notifies the 6LBR by proxying a keep-alive NS(EARO) on behalf of the 6LN that owns the address indicated in the Target Option.

The keep-alive EDAR and the NS(EARO) messages MUST be constructed as follows:

- o The Target IPv6 address from in the RPL Target Option is placed in the Registered Address field of the EDAR message and in the Target field of the NS message, respectively
- o the ROVR field in the keep-alive EDAR is set to 64-bits of all ones to indicate that it is not provided and this is a keep-alive EDAR. The actual value of the ROVR for that registration is returned by the 6LBR in an EDAC, and used in the proxy NS(EARO).
- o the Registration Lifetime is adapted from the Path Lifetime in the TIO by converting the Lifetime Units used in RPL into units of 60 seconds used in the 6LoWPAN ND messages.
- o The RPL Root indicates its own MAC Address as Source Link Layer Address (SLLA) in the NS(EARO).
- o the TID value is set to the Path Sequence in the TIO. The 'T' flag and an ICMP code of 1 are used in the NS(EARO) and the DAR message, respectively.

Upon a status in a DAC message that is not "Success", the Root MAY destroy the formed paths using a No-Path DAO downwards as specified in [I-D.ietf-roll-efficient-npdao].

In Non-Storing Mode, the outer IPv6 header that is used by the Root to transport the source routing information in data packets down the DODAG has the 6LR that serves the 6LN as final destination. This way, when the final 6LR decapsulates the outer header, it also removes all the RPL artifacts from the packet.

#### 7.5. 6LBR Operation

Upon reception of a DAR message with the Owner Unique ID field is set to all ones, the 6LBR checks whether an entry exists for the and computes whether the TID in the DAR message is fresher than that in the entry as prescribed in section 4.2.1. of [I-D.ietf-6lo-rfc6775-update].

If the entry does not exist, the 6LBR does not create the entry, and answers with a Status "Removed" in the DAC message.

If the entry exists but is not fresher, the 6LBR does not update the entry, and answers with a Status "Success" in the DAC message.

If the entry exists and the TID in the DAR message is fresher, the 6LBR updates the TID in the entry, and if the lifetime of the entry is extended by the Registration Lifetime in the DAR message, it also updates the lifetime of the entry. In that case, the 6LBR replies with a Status "Success" in the DAC message.

## 8. Implementation Status

## 9. Security Considerations

The LLN nodes depend on the 6LBR and the RPL participants for their operation. A trust model must be put in place to ensure that the right devices are acting in these roles, so as to avoid threats such as black-holing, or bombing attack whereby an impersonated 6LBR would destroy state in the network by using the "Removed" Status code. This trust model could be at a minimum based on a Layer-2 access control, or could provide role validation as well. This is a generic 6LoWPAN requirement, see Req5.1 in Appendix of [I-D.ietf-6lo-rfc6775-update].

The keep-alive EDAR message does not carry a valid Registration Unique ID [I-D.ietf-6lo-rfc6775-update] and it cannot be used to create a binding state in the 6LBR. The 6LBR MUST NOT create an entry based on a keep-alive EDAR that does not match an existing entry. All it can do is refresh the lifetime and the TID of an existing entry.

## 10. IANA Considerations

This specification has no requirement on IANA.

## 11. Acknowledgments

The author wishes to thank Michael Richardson and Georgios Papadopoulos for their early reviews of and contributions to this document

## 12. References

### 12.1. Normative References

[I-D.ietf-6lo-rfc6775-update]  
Thubert, P., Nordmark, E., Chakrabarti, S., and C. Perkins, "Registration Extensions for 6LoWPAN Neighbor Discovery", draft-ietf-6lo-rfc6775-update-19 (work in progress), April 2018.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <<https://www.rfc-editor.org/info/rfc4861>>.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <<https://www.rfc-editor.org/info/rfc4862>>.
- [RFC4919] Kushalnagar, N., Montenegro, G., and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", RFC 4919, DOI 10.17487/RFC4919, August 2007, <<https://www.rfc-editor.org/info/rfc4919>>.
- [RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <<https://www.rfc-editor.org/info/rfc6550>>.
- [RFC6606] Kim, E., Kaspar, D., Gomez, C., and C. Bormann, "Problem Statement and Requirements for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing", RFC 6606, DOI 10.17487/RFC6606, May 2012, <<https://www.rfc-editor.org/info/rfc6606>>.
- [RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <<https://www.rfc-editor.org/info/rfc6775>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.

## 12.2. Informative References

- [I-D.ietf-6lo-ap-nd]  
Thubert, P., Sarikaya, B., and M. Sethi, "Address Protected Neighbor Discovery for Low-power and Lossy Networks", draft-ietf-6lo-ap-nd-06 (work in progress), February 2018.
- [I-D.ietf-roll-efficient-npdao]  
Jadhav, R., Thubert, P., Sahoo, R., and Z. Cao, "Efficient Route Invalidation", draft-ietf-roll-efficient-npdao-03 (work in progress), March 2018.
- [I-D.ietf-roll-useofrplinfo]  
Robles, I., Richardson, M., and P. Thubert, "When to use RFC 6553, 6554 and IPv6-in-IPv6", draft-ietf-roll-useofrplinfo-23 (work in progress), May 2018.
- [IEEEstd802154]  
IEEE standard for Information Technology, "IEEE Standard for Local and metropolitan area networks-- Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)".
- [RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <<https://www.rfc-editor.org/info/rfc3315>>.
- [RFC6687] Tripathi, J., Ed., de Oliveira, J., Ed., and JP. Vasseur, Ed., "Performance Evaluation of the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6687, DOI 10.17487/RFC6687, October 2012, <<https://www.rfc-editor.org/info/rfc6687>>.
- [RFC7102] Vasseur, JP., "Terms Used in Routing for Low-Power and Lossy Networks", RFC 7102, DOI 10.17487/RFC7102, January 2014, <<https://www.rfc-editor.org/info/rfc7102>>.
- [RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, DOI 10.17487/RFC7228, May 2014, <<https://www.rfc-editor.org/info/rfc7228>>.

Author's Address

Pascal Thubert (editor)  
Cisco Systems, Inc  
Building D  
45 Allee des Ormes - BP1200  
MOUGINS - Sophia Antipolis 06254  
FRANCE

Phone: +33 497 23 26 34  
Email: pthubert@cisco.com

ROLL  
Internet-Draft  
Updates: 6550, 8505 (if approved)  
Intended status: Standards Track  
Expires: October 10, 2019

P. Thubert, Ed.  
Cisco  
April 8, 2019

Routing for RPL Leaves  
draft-thubert-roll-unaware-leaves-07

Abstract

This specification leverages 6LoWPAN ND to provide a unicast and multicast routing service in a RPL domain to 6LNs that do not participate to RPL.

Status of This Memo

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

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

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

This Internet-Draft will expire on October 10, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

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



## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	4
2.1. BCP 14 . . . . .	4
2.2. References . . . . .	4
2.3. Subset of a 6LoWPAN Glossary . . . . .	5
3. 6LoWPAN Neighbor Discovery . . . . .	6
4. Updating RFC 6550 . . . . .	7
5. Updating RFC 8505 . . . . .	7
6. Dependencies on the 6LN . . . . .	8
7. Protocol Operations for Unicast Addresses . . . . .	8
7.1. General Flow . . . . .	8
7.2. 6LN Operation . . . . .	11
7.3. 6LR Operation . . . . .	12
7.4. RPL Root Operation . . . . .	13
7.5. 6LBR Operation . . . . .	14
8. Protocol Operations for Multicast Addresses . . . . .	15
9. Implementation Status . . . . .	17
10. Security Considerations . . . . .	17
11. IANA Considerations . . . . .	17
12. Acknowledgments . . . . .	17
13. References . . . . .	17
13.1. Normative References . . . . .	17
13.2. Informative References . . . . .	19
Author's Address . . . . .	20

## 1. Introduction

The design of Low Power and Lossy Networks (LLNs) is generally focused on saving energy, which is the most constrained resource of all. Other design constraints, such as a limited memory capacity, duty cycling of the LLN devices and low-power lossy transmissions, derive from that primary concern.

The IETF produced the "Routing Protocol for Low Power and Lossy Networks" [RFC6550] (RPL) to provide routing services within such constraints. RPL is a Distance-Vector protocol, which, compared to link-state protocols, limits the amount of topological knowledge that needs to be installed and maintained in each node. In order to operate in constrained networks, RPL allows a Routing Stretch (see [RFC6687]), whereby routing is only performed along a DODAG as opposed to straight along a shortest path between 2 peers, whatever that would mean in a given LLN. This trades the quality of peer-to-peer (P2P) paths for a vastly reduced amount of control traffic and routing state that would be required to operate a any-to-any shortest path protocol. Finally, broken routes may be fixed lazily and on-

demand, based on dataplane inconsistency discovery, which avoids wasting energy in the proactive repair of unused paths.

In order to cope with lossy transmissions, RPL forms Direction-Oriented Directed Acyclic Graphs (DODAGs) using DODAG Information Solicitation (DIS) and DODAG Information Object (DIO) messages. For most of the nodes, though not all, a DODAG provides multiple forwarding solutions towards the Root of the topology via so-called parents. RPL is designed to adapt to fuzzy connectivity, whereby the physical topology cannot be expected to reach a stable state, with a lazy control that creates routes proactively but only fixes them when they are used by actual traffic. It results that RPL provides reachability for most of the LLN nodes, most of the time, but does not really converge in the classical sense. RPL provides unicast and multicast routing services back to RPL-Aware nodes (RANs). A RAN will inject routes to self using Destination Advertisement Object (DAO) messages sent to either their parents in Storing Mode or to the Root indicating their parent in Non-Storing mode. This process effectively forms a DODAG back to the device that is a subset of the DODAG to the Root with all links reversed.

When a routing protocol such as RPL is used to maintain reachability within a Non-Broadcast Multi-Access (NBMA) subnet, some nodes may act as routers and participate to the routing operations whereas others may be plain hosts. In RPL terms, a plain host that does not participate to the routing protocol is called a Leaf. It must be noted that a 6LN could participate to RPL and inject DAO routes to self, but refrain from advertising DIO and get children. In that case, the 6LN is still a host but not a Leaf.

This specification enables a RPL-Unaware Leaf (RUL) to announce itself as a host and demand that the 6LR that accepts the registration also inject the relevant routing information for the Registered Address in the RPL domain on its behalf. The unicast packet forwarding operation by the 6LR serving a Leaf 6LN is described in "When to use RFC 6553, 6554 and IPv6-in-IPv6" [I-D.ietf-roll-useofrplinfo]. This document adds the capability by a 6LR to advertise the Global, Unique-Local and Multicast IPv6 address(es) of the 6LN in the RPL protocol.

Examples of routing-agnostic 6LN may include lightly-powered sensors such as window smash sensor (alarm system), or the kinetically powered light switch. Other application of this specification may include a smart grid network that controls appliances - such as washing machines or the heating system - in the home. Appliances may not participate to the RPL protocol operated in the smart grid network but can still receive control packet from the smart grid.

## 2. Terminology

### 2.1. BCP 14

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 BCP 14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

### 2.2. References

The Terminology used in this document is consistent with and incorporates that described in Terms Used in Routing for Low-Power and Lossy Networks (LLNs). [RFC7102].

Other terms in use in LLNs are found in Terminology for Constrained-Node Networks [RFC7228].

A glossary of classical 6LoWPAN acronyms is given in Section 2.3.

The term "byte" is used in its now customary sense as a synonym for "octet".

"RPL", "RPL Packet Information" (RPI) and "RPL Instance", DIO, DAO and DIS messages are defined in the "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks" [RFC6550] specification.

This document introduces the term RPL-Unaware Leaf (RUL) to refer to a node that uses a RPL router (without necessarily knowing it) as 6LR and depends on that router to obtain reachability for its addresses inside the RPL domain. On the contrary, the term RPL-Aware Leaf (RAL) is used to refer to a host or a router that participates to RPL and advertises its addresses of prefixes by itself.

Other terms in use in LLNs are found in Terminology for Constrained-Node Networks [RFC7228].

Readers are expected to be familiar with all the terms and concepts that are discussed in

- o "Neighbor Discovery for IP version 6" [RFC4861],
- o "IPv6 Stateless Address Autoconfiguration" [RFC4862],
- o "Problem Statement and Requirements for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing" [RFC6606],

- o "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals" [RFC4919],
- o "Neighbor Discovery Optimization for Low-power and Lossy Networks" [RFC6775], and
- o "Registration Extensions for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Neighbor Discovery" [RFC8505].

### 2.3. Subset of a 6LoWPAN Glossary

This document often uses the following acronyms:

6BBR: 6LoWPAN Backbone Router (proxy for the registration)

6LBR: 6LoWPAN Border Router (authoritative on DAD)

6LN: 6LoWPAN Node

6LR: 6LoWPAN Router (relay to the registration process)

6CIO: Capability Indication Option

(E)ARO: (Extended) Address Registration Option

(E)DAR: (Extended) Duplicate Address Request

(E)DAC: (Extended) Duplicate Address Confirmation

DAD: Duplicate Address Detection

DODAG: Destination-Oriented Directed Acyclic Graph

LLN: Low-Power and Lossy Network (a typical IoT network)

NA: Neighbor Advertisement

NCE: Neighbor Cache Entry

ND: Neighbor Discovery

NDP: Neighbor Discovery Protocol

NS: Neighbor Solicitation

ROVR: Registration Ownership Verifier (pronounced rover)

RPL: IPv6 Routing Protocol for LLNs (pronounced ripple)

RA: Router Advertisement

RS: Router Solicitation

TSCH: Timeslotted Channel Hopping

TID: Transaction ID (a sequence counter in the EARO)

### 3. 6LoWPAN Neighbor Discovery

The IPv6 [RFC8200]Neighbor Discovery (IPv6 ND) Protocol (NDP) suite [RFC4861] [RFC4862] defined for fast media such a Ethernet, relies heavily on multicast operations for address discovery and duplicate address detection (DAD).

"Neighbor Discovery Optimizations for 6LoWPAN networks" [RFC6775] (6LoWPAN ND) adapts IPv6 ND for operations over energy-constrained LLNs. In particular, 6LoWPAN ND introduces a unicast host address registration mechanism that contributes to reduce the use of multicast messages that are present in the classical IPv6 ND protocol. 6LoWPAN ND defines a new Address Registration Option (ARO) that is carried in the unicast Neighbor Solicitation (NS) and Neighbor Advertisement (NA) messages between the 6LoWPAN Node (6LN) and the 6LoWPAN Router (6LR). 6LoWPAN ND also defines the Duplicate Address Request (DAR) and Duplicate Address Confirmation (DAC) messages between the 6LR and the 6LoWPAN Border Router (6LBR). In an LLN, the 6LBR is the central repository of all the Registered Addresses in its domain.

"Registration Extensions for 6LoWPAN Neighbor Discovery" [RFC8505] updates the behavior of RFC 6775 to enable a generic registration to routing services and defines an Extended ARO (EARO). The format of the EARO is shown in Figure 1:

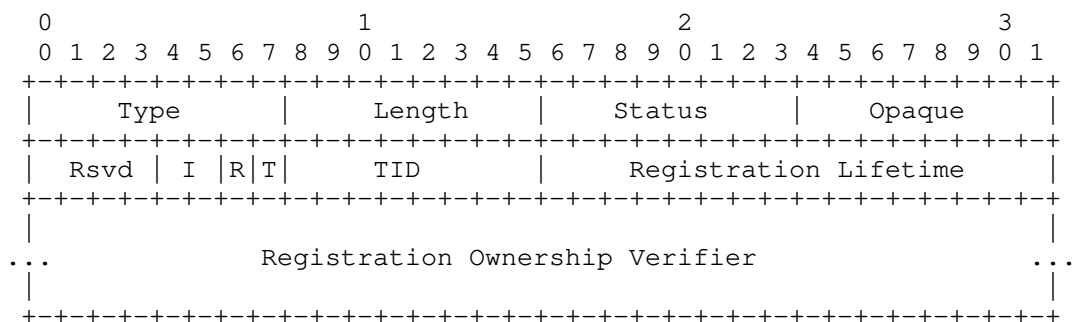


Figure 1: EARO Option Format

The 'R' flag that is set if the Registering Node expects that the 6LR ensures reachability for the Registered Address, e.g., by means of routing or proxying ND.

The EARO also includes a sequence counter called Transaction ID (TID), which maps to the Path Sequence Field found in Transit Options in RPL DAO messages. It is a prerequisite for this specification.

Finally, the EARO transports an Opaque field and an 'I' field that describes what the Opaque field transports and how to use it. This specification requires that the I field is left to 0 and to use the Opaque field to carry the RPL InstanceID if one is known, else to leave the Opaque field to zero.

#### 4. Updating RFC 6550

This document specifies a new behavior whereby a 6LR injects DAO messages for unicast addresses registered through the updated 6LoWPAN ND [RFC8505] on behalf of 6LN nodes that are not RPL-aware.

Upon the renewal of a 6LoWPAN ND registration, this specification changes the behavior of the 6LR as follows. If the 'R' flag is set, the 6LR injects a DAO targeting the Registered Address, and refrains from sending a DAR message. the DAR/DAC exchange that refreshes the state in the 6LBR happens instead between the RPL Root and the 6LBR. In that flow, the RPL Root acts as a proxy on behalf of the 6LR upon the reception of the DAO propagation initiated at the 6LR.

#### 5. Updating RFC 8505

The behavior defined in this specification whereby the 6LR that processes the registration advertises the Registered Address in DAO messages and bypasses the DAR/DAC process for the renewal of a registration, is only triggered by an NS(EARO) that has the 'R' flag set. If the 'R' flag is not set, then the Registering Node is expected to be a RAN router that handles the reachability of the Registered Address by itself.

This document also specifies a keep-alive EDAR message that the RPL Root may use to maintain an existing state in the 6LBR upon receiving DAO messages. The keep-alive EDAR message may only act as a refresher and can only update the Lifetime and the TID of the state in the 6LBR.

This document similarly specifies a keep-alive NS(EARO) message that the RPL Root may use to maintain an existing state in a 6BBR upon receiving DAO messages. The keep-alive NS(EARO) message may only act

as a refresher and can only update the Lifetime and the TID of the state in the 6BBR.

As prescribed by [RFC8505], a RPL router SHOULD NOT set the 'R' flag.

## 6. Dependencies on the 6LN

This document provides RPL routing for a 6LN acting as a plain host and not aware of RPL. Still, a minimal RPL-independent functionality is expected from the 6LN in order to operate properly as a RLU; in particular:

- o the 6LN MUST implement [RFC8505] and set the 'R' flag in the EARO option. The 'R' flag is used to determine whether the Registering Node is a RUL, not aware of the RPL operation in the network, and thus does not participate to it. A 6LN is considered to be a RUL if and only if it sets the 'R' flag in the EARO.
- o RPL data packets are often encapsulated using IP in IP and in non-storing mode, packets going down will carry an SRH as well. RPL data packets also typically carry a Hop-by-Hop Header to transport a RPL Packet Information (RPI) [RFC6550]. These additional headers are called RPL artifacts.
- o When IP-in-IP is used and the outer headers terminate at the 6LR that generated the DAO, then the 6LR decapsulates the packet to the 6LN. In that case the 6LN gets a packet that is free of RPL artifacts. IP-in-IP to the 6LR MUST be used if the 6LN cannot handle the RPL artifacts or the way they are compressed [RFC8138]. It SHOULD be used if there is a particular bandwidth or power constraint at the 6LN.
- o In order to save the IP-in-IP encapsulation and to support storing mode of operation, it is preferred that the 6LN can ignore an RPI and consume a routing header in both the native and compressed forms. In order to enable IP-in-IP to a 6LN in non storing mode, it is also of interest that the 6LN supports decapsulating IP-in-IP in both forms. But since the preferred behaviour when using IP-in-IP is that the outer headers terminate at the 6LR, supporting this capability is secondary.

## 7. Protocol Operations for Unicast Addresses

### 7.1. General Flow

This specification enables to save the exchange of Extended Duplicate Address messages, EDAR and EDAC, from a 6LN all the way to the 6LBR across a RPL mesh, for the sole purpose of refreshing an existing

state in the 6LBR. Instead, the EDAR/EDAC exchange is proxied by the RPL Root upon a DAO message that refreshes the RPL routing state. To achieve this, the lifetimes and sequence counters in 6LoWPAN ND and RPL are aligned. In other words, the Path Sequence and the Path Lifetime in the DAO message are derived from the Transaction ID and the registration lifetime in the NS(EARO) message from the 6LN.

From the perspective of the 6LN, the registration flow happens transparently; it is not delayed by the proxy RPL operation, so the device does not need to wait more whether RPL proxy operation happens or not. The flows below are RPL Non-Storing Mode examples. In Storing Mode, the DAO ACK may not be present, and the DAO messages cascade from child to parent all the way to the DODAG Root.

On the first registration, illustrated in Figure 2, from the perspective of the 6LR in non-storing mode, the Extended Duplicate Address message takes place as prescribed by [RFC8505]. When successful, the flow creates a Neighbor Cache Entry (NCE) in the 6LR, and the 6LR injects the Registered Address in RPL using DAO/DAO-ACK exchanges all the way to the RPL DODAG Root. The protocol does not carry a specific information that the Extended Duplicate Address messages were already exchanged, so the Root proxies them anyway. Note that in Storing Mode the DAO ACK is generated from the parent that does not necessary wait for the grand parent to acknowledge, so the DAO-ACK is no guarantee that the keep-alive EDAR succeeded. On the other hand, the flows can be nested in non storing mode, and it is possible to carry information such as an updated lifetime from the 6LBR all the way to the 6LN.



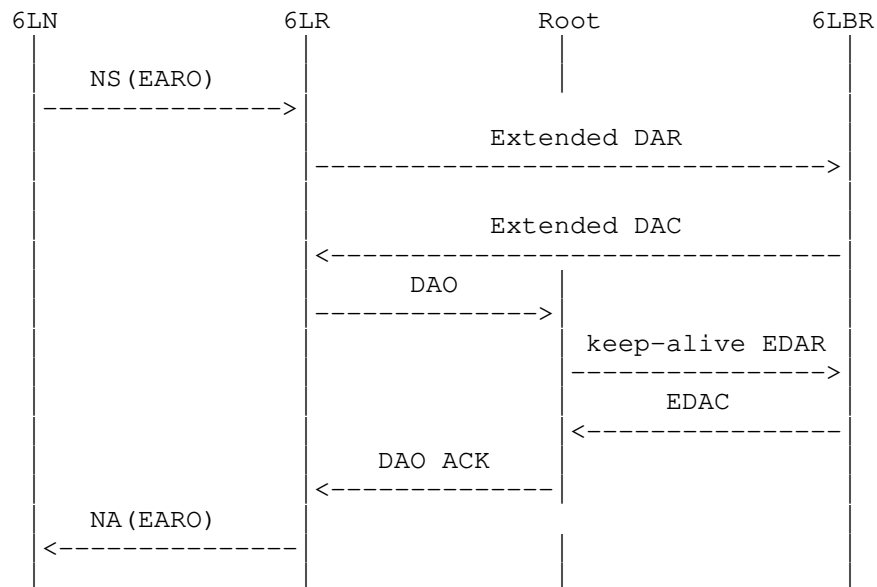


Figure 2: First Registration Flow

A re-registration is performed by the 6LN to maintain the NCE in the 6LR alive before lifetime expires. Upon a re-registration, as illustrated in Figure 3, the 6LR redistributes the Registered Address NS(EARO) in RPL. This causes the RPL DODAG Root to refresh the state in the 6LBR with a keep-alive EDAC message. The keep-alive EDAC lacks the Registration Ownership Verifier (ROVR) information, since it is not present in RPL DAO messages, but the EDAC message sent in response by the 6LBR contains the actual value of the ROVR field for that registration. This enables the RPL Root to perform the proxy-registration for the Registered Address and attract traffic captured over the backbone by the 6BBR and route it back to the device.

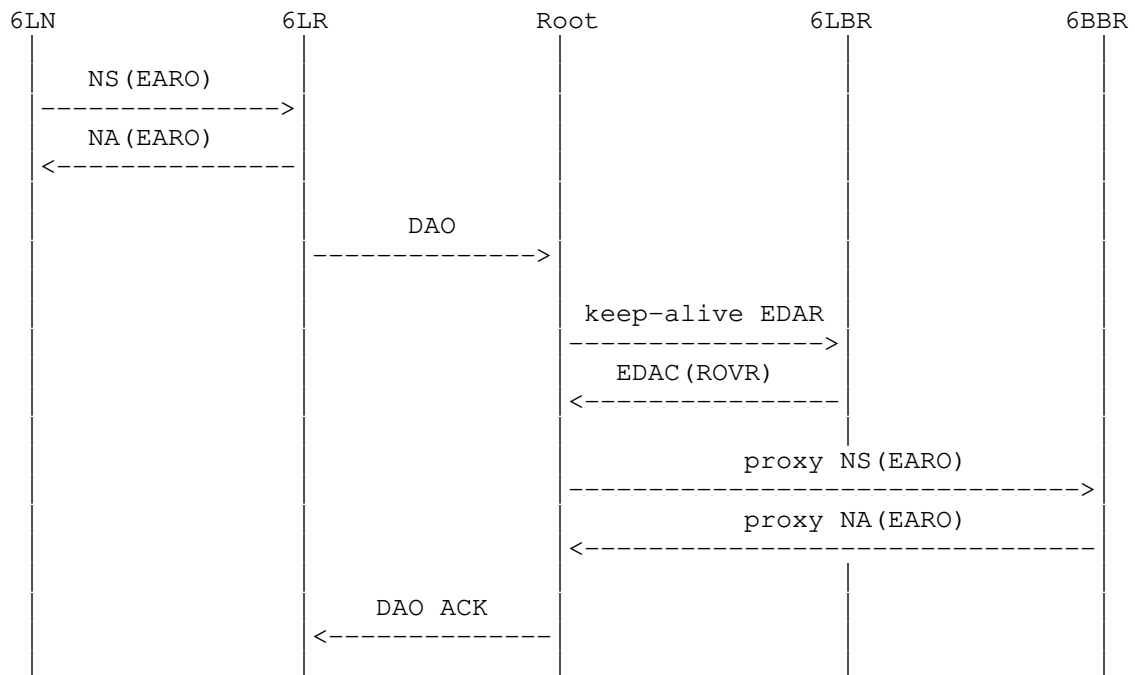


Figure 3: Next Registration Flow

Note that any of the functions 6LR, Root and 6LBR might be collapsed in a single node, in which case the flow above happens internally, and possibly through internal API calls as opposed to messaging.

## 7.2. 6LN Operation

This specification does not alter the operation of a 6LoWPAN ND-compliant 6LN, which is expected to operate as follows:

- o The 6LN obtains an IPv6 global address, for instance using autoconfiguration [RFC4862] based on a Prefix Information Option (PIO) [RFC4861] found in a Router Advertisement message or by some other means such as DHCPv6 [RFC3315].
- o Once it has formed an address, the 6LN (re)registers its address periodically, within the Lifetime of the previous registration, as prescribed by [RFC8505].
- o Upon each consecutive registration, the 6LN MUST increase the TID field.

- o If the 6LN is aware of the RPL Instance the packet should be injected into, then it SHOULD set the Opaque field to the InstanceID, else it MUST leave the Opaque field to zero. In any fashion the 6LN MUST set the 'I' field to zero.
- o A 6LN acting as a RUL MUST set the 'R' flag in the EARO whereas a 6LN acting as a RAN SHOULD NOT set the 'R' flag.
- o The 6LN MAY register to more than one 6LR at the same time. In that case, a same value of TID is used for each registration.
- o The 6LN MAY use any of the 6LRs to which it register to forward its packets.
- o the 6LN is not expected to be aware of RPL so it is not expected to produce RPL artifacts in the data packets.

### 7.3. 6LR Operation

Also as prescribed by [RFC8505], the 6LR generates a DAR message upon reception of a valid NS(EARO) message for the registration of a new IPv6 Address by a 6LN. If the Duplicate Address exchange succeeds, then the 6LR installs a Neighbor Cache Entry (NCE). If the 'R' flag was set in the EARO of the NS message, and this 6LR can manage the reachability of Registered Address, then the 6LR sets the 'R' flag in the ARO of the response NA message.

From then on, the 6LN periodically sends a new NS(EARO) to refresh the NCE state before the lifetime indicated in the EARO expires, with TID that is incremented each time till it wraps in a lollipop fashion. As long as the 'R' flag is set and this router can still manage the reachability of Registered Address, the 6LR keeps setting the 'R' flag in the EARO of the response NA message, but the exchange of Extended Duplicate Address messages is skipped.

The Opaque field in the EARO hints the 6LR on the RPL Instance that should be used for the DAO advertisements, and for the forwarding of packets sourced at the registered address when there is no RPL Packet Information (RPI) in the packet, in which case the 6LR SHOULD add one to the packet. if the 'I' field is not zero, then the 6LR MUST consider that the Opaque field is left to zero. If the Opaque field is not set to zero, then it should carry a RPL InstanceID for the Instance suggested by the 6LN. If the 6LR does not participate to the associated Instance, then the 6LR MUST consider that the Opaque field is left to zero. If the Opaque field left to zero, the 6LR is free to use the default Instance (zero) for the registered address or to select an Instance of its choice; else, that is if the 6LR

participates to the suggested Instance, then the 6LR SHOULD use that Instance for the registered address.

Upon a successful NS/NA(EARO) exchange: if the 'R' flag was set in the EARO of the NS message, then the 6LR SHOULD inject the Registered Address in RPL by sending a DAO message on behalf of the 6LN; else the 6LR MUST NOT inject the Registered Address into RPL.

The DAO message advertising the Registered Address MUST be constructed as follows:

- o The Registered Address is placed in a RPL Target Option in the DAO message as the Target Prefix, and the Prefix Length is set to 128
- o the External 'E' flag in the Transit Information Option (TIO) associated to the Target Option is set to indicate that the 6LR redistributes an external target into the RPL network. This is how the root knows in non-storing mode to use IP-in-IP and terminate the outters headers at the 6LR that generated the DAO.
- o the Path Lifetime in the TIO is computed from the Lifetime in the EARO Option to adapt it to the Lifetime Units used in the RPL operation. Note that if the lifetime is 0, then the 6LR generates a No-Path DAO message that cleans up the routes down to the Address of the 6LN.
- o the Path Sequence in the TIO is set to the TID value found in the EARO option.
- o Additionally, in Non-Storing Mode the 6LR indicates one of its global IPv6 unicast addresses as the Parent Address in the TIO.

If a 6LR receives a valid NS(EARO) message with the 'R' flag reset and the 6LR was redistributing the Registered Address due to previous NS(EARO) messages with the flag set, then it MUST stop injecting the address. It is up to the Registering Node to maintain the corresponding route from then on, either keeping it active by sending further DAO messages, or destroying it using a No-Path DAO.

#### 7.4. RPL Root Operation

In RPL Storing Mode of Operation (MOP), the DAO message is propagated from child to parent all the way to the Root along the DODAG, populating routing state as it goes. In Non-Storing Mode, The DAO message is sent directly to the route. Upon reception of a DAO message that creates or updates an existing RPL state:

- o the Root notifies the 6LBR using an internal API if they are collocated, or performs a keep-alive DAR/DAC exchange on behalf of the registering node if they are separated.
- o In an extended topology with a Backbone Link, the Root notifies the 6LBR by proxying a keep-alive NS(EARO) on behalf of the 6LN that owns the address indicated in the Target Option.

The keep-alive EDAR and the NS(EARO) messages MUST be constructed as follows:

- o The Target IPv6 address from in the RPL Target Option is placed in the Registered Address field of the EDAR message and in the Target field of the NS message, respectively
- o the ROVR field in the keep-alive EDAR is set to 64-bits of all ones to indicate that it is not provided and this is a keep-alive EDAR. The actual value of the ROVR for that registration is returned by the 6LBR in an EDAC, and used in the proxy NS(EARO).
- o the Registration Lifetime is adapted from the Path Lifetime in the TIO by converting the Lifetime Units used in RPL into units of 60 seconds used in the 6LoWPAN ND messages.
- o The RPL Root indicates its own MAC Address as Source Link Layer Address (SLLA) in the NS(EARO).
- o the TID value is set to the Path Sequence in the TIO. The 'T' flag and an ICMP code of 1 are used in the NS(EARO) and the DAR message, respectively.

Upon a status in a DAC message that is not "Success", the Root MAY destroy the formed paths using a No-Path DAO downwards as specified in [I-D.ietf-roll-efficient-npdao].

In Non-Storing Mode, the outer IPv6 header that is used by the Root to transport the source routing information in data packets down the DODAG has the 6LR that serves the 6LN as final destination. This way, when the final 6LR decapsulates the outer header, it also removes all the RPL artifacts from the packet.

#### 7.5. 6LBR Operation

Upon reception of a DAR message with the Owner Unique ID field is set to all ones, the 6LBR checks whether an entry exists for the and computes whether the TID in the DAR message is fresher than that in the entry as prescribed in section 4.2.1. of [RFC8505].

If the entry does not exist, the 6LBR does not create the entry, and answers with a Status "Removed" in the DAC message.

If the entry exists but is not fresher, the 6LBR does not update the entry, and answers with a Status "Success" in the DAC message.

If the entry exists and the TID in the DAR message is fresher, the 6LBR updates the TID in the entry, and if the lifetime of the entry is extended by the Registration Lifetime in the DAR message, it also updates the lifetime of the entry. In that case, the 6LBR replies with a Status "Success" in the DAC message.

## 8. Protocol Operations for Multicast Addresses

Section 12 of [RFC6550] details the RPL support for multicast flows. This support is not source-specific and only operates as an extension to the Storing Mode of Operation for unicast packets. Note that it is the RPL model that the multicast packet is passed as a Layer-2 unicast to each of the interested children. This remains true when forwarding between the 6LR and the listener 6LN.

"Multicast Listener Discovery (MLD) for IPv6" [RFC2710] and its updated version "Multicast Listener Discovery Version 2 (MLDv2) for IPv6" [RFC3810] provide an interface for a listener to register to multicast flows. MLDv2 is backwards compatible with MLD, and adds in particular the capability to filter the sources via black lists and white lists. In the MLD model, the router is a "querier" and the host is a multicast listener that registers to the querier to obtain copies of the particular flows it is interested in.

On the first registration, as illustrated in Figure 4, the 6LN, as an MLD listener, sends an unsolicited Report to the 6LR in order to start receiving the flow immediately. Since multicast Layer-2 messages are avoided, it is important that the asynchronous messages for unsolicited Report and Done are sent reliably, for instance using an Layer-2 acknowledgement, or attempted multiple times.

The 6LR acts as a generic MLD querier and generates a DAO for the multicast target. The lifetime of the DAO is set to be in the order of the Query Interval, yet larger to account for variable propagation delays.

The root proxies the MLD exchange as listener with the 6BBR acting as the querier, so as to get packets from a source external to the RPL domain. Upon a DAO with a multicast target, the RPL root checks if it is already registered as a listener for that address, and if not, it performs its own unsolicited Report for the multicast target.

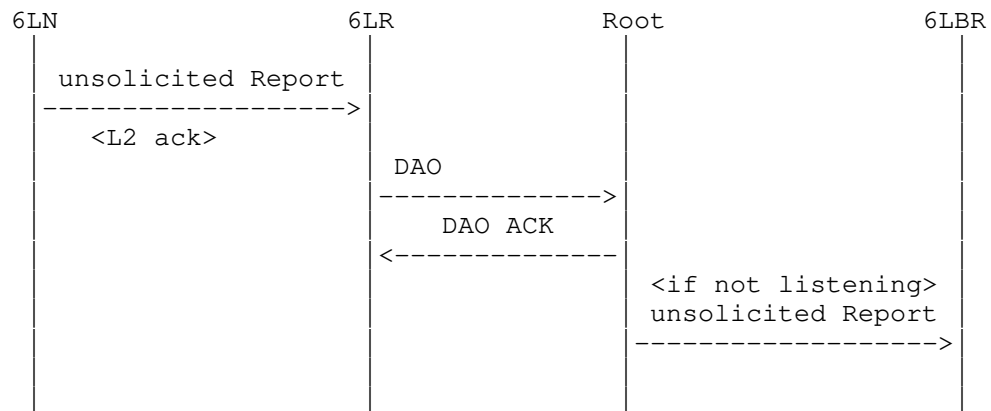


Figure 4: First Multicast Registration Flow

A re-registration is pulled by 6LR acting as querier. Note that the message may be sent unicast to all the known individual listeners. Upon a time out of the Query Interval, the 6LR sends a Query to each of its listeners, and gets a Report back that is mapped into a DAO, as illustrated in Figure 5,

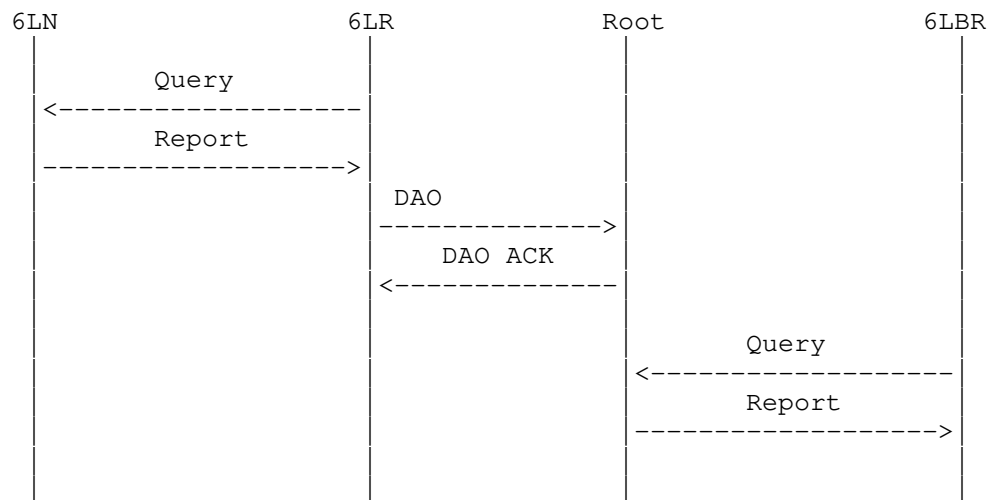


Figure 5: Next Registration Flow

Note that any of the functions 6LR, Root and 6LBR might be collapsed in a single node, in which case the flow above happens internally, and possibly through internal API calls as opposed to messaging.

## 9. Implementation Status

## 10. Security Considerations

The LLN nodes depend on the 6LBR and the RPL participants for their operation. A trust model must be put in place to ensure that the right devices are acting in these roles, so as to avoid threats such as black-holing, or bombing attack whereby an impersonated 6LBR would destroy state in the network by using the "Removed" Status code. This trust model could be at a minimum based on a Layer-2 access control, or could provide role validation as well. This is a generic 6LoWPAN requirement, see Req5.1 in Appendix of [RFC8505].

The keep-alive EDAR message does not carry a valid Registration Unique ID [RFC8505] and it cannot be used to create a binding state in the 6LBR. The 6LBR MUST NOT create an entry based on a keep-alive EDAR that does not match an existing entry. All it can do is refresh the lifetime and the TID of an existing entry.

## 11. IANA Considerations

This specification has no requirement on IANA.

## 12. Acknowledgments

The author wishes to thank Michael Richardson and Georgios Papadopoulos for their early reviews of and contributions to this document

## 13. References

### 13.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, DOI 10.17487/RFC2710, October 1999, <<https://www.rfc-editor.org/info/rfc2710>>.
- [RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, DOI 10.17487/RFC3810, June 2004, <<https://www.rfc-editor.org/info/rfc3810>>.



- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <<https://www.rfc-editor.org/info/rfc4861>>.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <<https://www.rfc-editor.org/info/rfc4862>>.
- [RFC4919] Kushalnagar, N., Montenegro, G., and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", RFC 4919, DOI 10.17487/RFC4919, August 2007, <<https://www.rfc-editor.org/info/rfc4919>>.
- [RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <<https://www.rfc-editor.org/info/rfc6550>>.
- [RFC6606] Kim, E., Kaspar, D., Gomez, C., and C. Bormann, "Problem Statement and Requirements for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing", RFC 6606, DOI 10.17487/RFC6606, May 2012, <<https://www.rfc-editor.org/info/rfc6606>>.
- [RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <<https://www.rfc-editor.org/info/rfc6775>>.
- [RFC8138] Thubert, P., Ed., Bormann, C., Toutain, L., and R. Cragie, "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing Header", RFC 8138, DOI 10.17487/RFC8138, April 2017, <<https://www.rfc-editor.org/info/rfc8138>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.

- [RFC8505] Thubert, P., Ed., Nordmark, E., Chakrabarti, S., and C. Perkins, "Registration Extensions for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Neighbor Discovery", RFC 8505, DOI 10.17487/RFC8505, November 2018, <<https://www.rfc-editor.org/info/rfc8505>>.

### 13.2. Informative References

- [I-D.ietf-6lo-ap-nd]  
Thubert, P., Sarikaya, B., Sethi, M., and R. Struik,  
"Address Protected Neighbor Discovery for Low-power and  
Lossy Networks", draft-ietf-6lo-ap-nd-11 (work in  
progress), February 2019.
- [I-D.ietf-roll-efficient-npdao]  
Jadhav, R., Thubert, P., Sahoo, R., and Z. Cao, "Efficient  
Route Invalidation", draft-ietf-roll-efficient-npdao-09  
(work in progress), October 2018.
- [I-D.ietf-roll-useofrplinfo]  
Robles, I., Richardson, M., and P. Thubert, "Using RPL  
Option Type, Routing Header for Source Routes and IPv6-in-  
IPv6 encapsulation in the RPL Data Plane", draft-ietf-  
roll-useofrplinfo-25 (work in progress), March 2019.
- [IEEEstd802154]  
IEEE standard for Information Technology, "IEEE Standard  
for Local and metropolitan area networks-- Part 15.4: Low-  
Rate Wireless Personal Area Networks (LR-WPANs)".
- [RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins,  
C., and M. Carney, "Dynamic Host Configuration Protocol  
for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July  
2003, <<https://www.rfc-editor.org/info/rfc3315>>.
- [RFC6687] Tripathi, J., Ed., de Oliveira, J., Ed., and JP. Vasseur,  
Ed., "Performance Evaluation of the Routing Protocol for  
Low-Power and Lossy Networks (RPL)", RFC 6687,  
DOI 10.17487/RFC6687, October 2012,  
<<https://www.rfc-editor.org/info/rfc6687>>.
- [RFC7102] Vasseur, JP., "Terms Used in Routing for Low-Power and  
Lossy Networks", RFC 7102, DOI 10.17487/RFC7102, January  
2014, <<https://www.rfc-editor.org/info/rfc7102>>.

[RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, DOI 10.17487/RFC7228, May 2014, <<https://www.rfc-editor.org/info/rfc7228>>.

Author's Address

Pascal Thubert (editor)  
Cisco Systems, Inc  
Building D  
45 Allee des Ormes - BP1200  
MOUGINS - Sophia Antipolis 06254  
FRANCE

Phone: +33 497 23 26 34  
Email: [pthubert@cisco.com](mailto:pthubert@cisco.com)