

ROLL
Internet-Draft
Intended status: Standards Track
Expires: September 6, 2018

S. Anamalamudi
Huaiyin Institute of Technology
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
March 5, 2018

Asymmetric AODV-P2P-RPL in Low-Power and Lossy Networks (LLNs)
draft-ietf-roll-aodv-rpl-03

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 September 6, 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
2. Terminology	4
3. Overview of AODV-RPL	6
4. AODV-RPL DIO Options	6
4.1. AODV-RPL DIO RREQ Option	6
4.2. AODV-RPL DIO RREP Option	8
4.3. AODV-RPL DIO Target Option	10
5. Symmetric and Asymmetric Routes	11
6. AODV-RPL Operation	13
6.1. Generating Route Request at OrigNode	13
6.2. Receiving and Forwarding Route Request	14
6.3. Generating Route Reply at TargNode	15
6.3.1. RREP-DIO for Symmetric route	15
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. ETX/RSSI Values to select S bit	21
Appendix B. Changes to version 02	22
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 sub-optimal routes and may suffer severe congestion near the DAG root [RFC6997], [RFC6998].

To discover optimal 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). TargNode 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. In other words, it is efficient to use only routing tables for P2P-RDO message instead of "Address vector" for hop-by-hop routes (H=1) over symmetric links.

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 eliminates the possibility to use 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] describes 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 control 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]. Additionally, 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. If the OrigNode doesn't require an upward route towards itself, the route is also considered as asymmetric.

Bi-directional Asymmetric Link

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

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.

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 of an application.

P2P

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

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

The 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. Both directions fulfill the constraints in route discovery.

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 point-to-point; in other words the routes are not constrained to traverse a common ancestor. Unlike core 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, route discovery is initiated by forming a temporary DAG rooted at the OrigNode. Paired DODAGs (Instances) are constructed according to a new 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 (as shown in Figure 4). 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. Similarly, the RREP-DIO message means the AODV-RPL mode DIO message from TargNode to OrigNode, containing the RREP option. Subsequently, the route discovered in the RREQ-Instance is used for data transmission from TargNode to OrigNode, and the route discovered in RREP-Instance is used for Data transmission from OrigNode to TargNode.

4. AODV-RPL DIO Options

4.1. AODV-RPL DIO RREQ Option

A RREQ-DIO message MUST carry exactly one RREQ option.

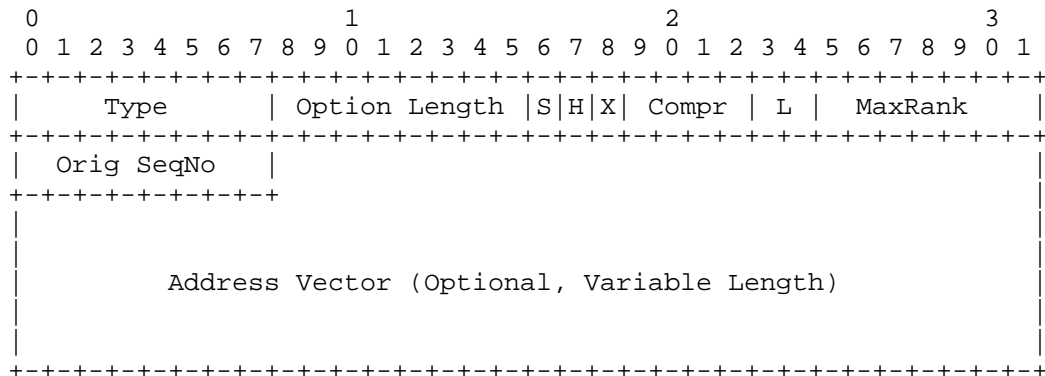


Figure 1: DIO RREQ option format for AODV-RPL MoP

OrigNode supplies the following information in the RREQ option of the RREQ-Instance message:

Type

The type of the RREQ option(see Section 9.2).

Option Length

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 issuing this RREQ-DIO. The bit SHOULD be set to 1 in the RREQ-DIO when the OrigNode initiates the route discovery.

X

Reserved.

H

The OrigNode sets this flag to one if it desires a hop-by-hop route. It sets this flag to zero if it desires a source route. This flag is valid to both downstream route and upstream route.

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.

L

2-bit unsigned integer. This field indicates the duration that a node joining 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 detailed definition can be found in [RFC6997].

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

It should be indicated here that L is not 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. A node MUST NOT join a temporary DODAG if its own rank would equal to or higher than the limit. A value of 0 in this field indicates the limit is infinity. For more details please refer to [RFC6997].

OrigNode Sequence Number

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

Address Vector (Optional)

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.

4.2. AODV-RPL DIO RREP Option

A RREP-DIO message MUST carry exactly one RREP option.

The TargNode supplies the following information in the RREP option:

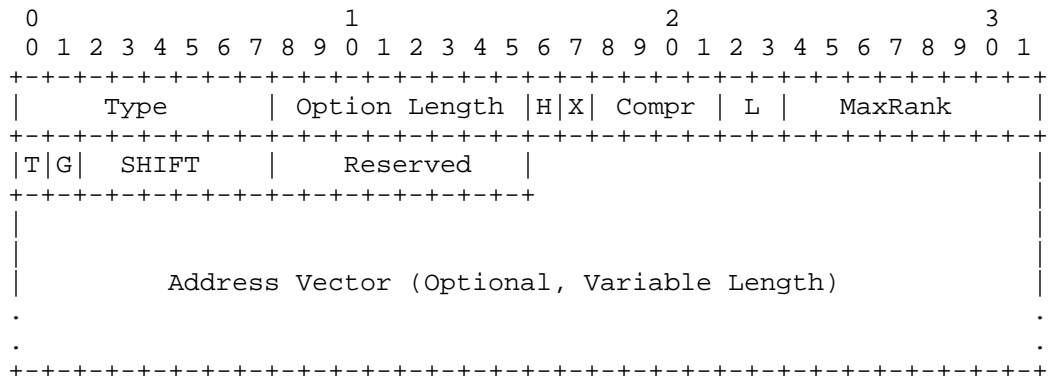


Figure 2: DIO RREP option format for AODV-RPL MoP

Type

The type of the RREP option (see Section 9.2)

Option Length

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.

H

This bit indicates the downstream route is source routing (H=0) or hop-by-hop (H=1). It SHOULD 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 with the same definition as in Section 4.1.

MaxRank

Same definition as in RREQ option.

T

'T' is set to 1 to indicate that the RREP-DIO MUST include exactly one AODV-RPL Target Option. Otherwise, the Target Option is not necessary in the RREP-DIO.

G

Gratuitous route (see Section 7).

SHIFT

6-bit unsigned integer. This field indicates the how many the original InstanceID (see Section 6.3.3) is shifted (added an integer from 0 to 63). 0 indicates that the original InstanceID is used.

Reserved

Reserved for future usage; MUST be initialized to zero and MUST be ignored upon reception.

Address Vector (Optional)

It is only present when the 'H' bit is set to 0. For an asymmetric route, it is a vector of IPv6 addresses representing the route that the RREP-DIO has passed. For symmetric route, it is the accumulated vector when the RREQ-DIO arrives at the TargNode.

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 encapsulating the address of the OrigNode if the 'T' bit is set to 1.

If an OrigNode want to discover routes to multiple TargNodes, and these routes share the same constraints, then the OrigNode can include all the addresses of the TargNodes into multiple AODV-RPL Target Options in the RREQ-DIO, so that the cost can be reduced to building only one DODAG. Different addresses of the TargNodes can merge if they share the same prefix.

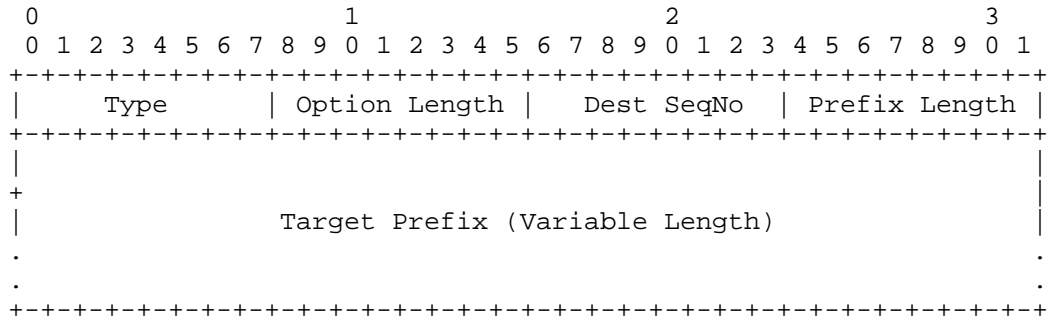


Figure 3: Target option format for AODV-RPL MoP

Type

The type of the AODV-RPL Target Option (see Section 9.2)

Destination Sequence Number

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 BorderRouter, 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. An intermediate router sends out RREQ-DIO with the 'S' bit set to 1, meaning that all the one-hop links on the route from the OrigNode to this router meet the requirements of route discovery; thus the route can be used symmetrically.

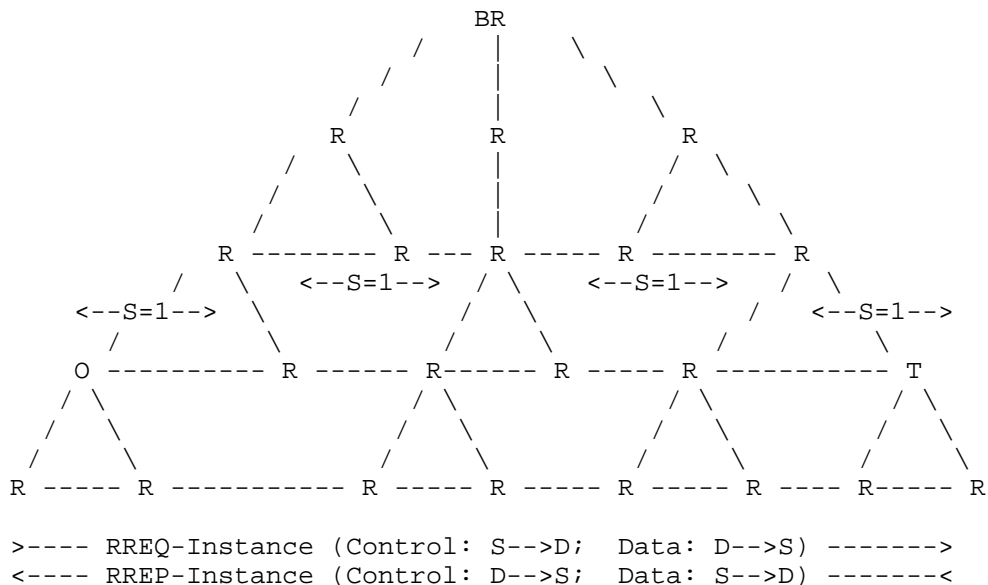


Figure 4: AODV-RPL with Symmetric Paired Instances

Upon receiving a RREQ-DIO with the 'S' bit set to 1, a node MUST decide if 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. Moreover, 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 decides whether or not the route is symmetric before transmitting the RREP-DIO message upstream towards the OrigNode.

The criterion and the corresponding metric used to determine if a one-hop link is symmetric or not is implementation specific and beyond the scope of the document. Also, the difference in the metric values for upward and downward directions of a link that can be establish its symmetric and asymmetric nature is implementation specific. For instance, the intermediate routers MAY choose to 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 estimate the metric using averaging techniques or any other means that is appropriate to the application context.

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.

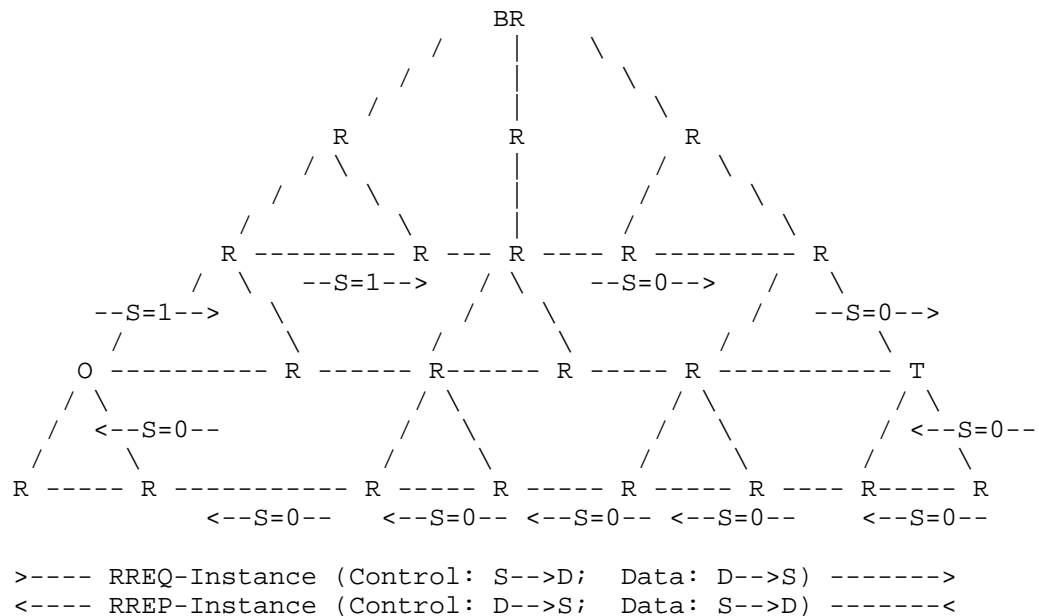


Figure 5: AODV-RPL with Asymmetric Paired Instances

6. AODV-RPL Operation

6.1. Generating Route Request at OrigNode

The route discovery process is initiated on-demand when an application at the OrigNode has data to be transmitted to the TargNode, but no route for the target exists or the current routes don't fulfill the requirements of the data transmission. In this case, the OrigNode MUST build a local RPLInstance and a DODAG rooted at itself. Then it begins to send out DIO message in AODV-RPL MoP via link-local multicast. The DIO MUST contain exactly one RREQ option as defined in Section 4.1, and at least one AODV-RPL Target Option as defined in Figure 3. This DIO message is noted as RREQ-DIO. The 'S' bit in RREQ-DIO sent out by the OrigNode is set as 1.

The maintenance of Originator and Destination Sequence Number in the RREQ option is as defined in AODV [RFC3561].

The address in the AODV-RPL Target Option can be a unicast IPv6 address, a prefix or a multicast address. 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.

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

The transmission of RREQ-DIO follows the Trickle timer. When the L duration has transpired, the OrigNode MUST leave the DODAG and stop sending any RREQ-DIOs in the related RPLInstance.

6.2. Receiving and Forwarding Route Request

Upon receiving a RREQ-DIO, a router out of 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 look into 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, then the link can't be used symmetrically, thus the 'S' bit of the RREQ-DIO to be send out MUST be set as 0. If the 'S' bit in the received RREQ-DIO is set to 0, the router MUST look only into the upward direction (i.e. 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 be inferior to the MaxRank limit, the router chooses to join in the DODAG of the RREQ-Instance. The router issuing the received RREQ-DIO is selected as the preferred parent. Afterwards, other RREQ-DIO message can be received. How to maintain the parent set, select the preferred parent, and update the router's rank follows 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 NOT join in the DODAG. Otherwise, go to the following steps 2, 3, 4 and 5.

A router MUST discard a received RREQ-DIO if the advertised rank equals or exceeds the MaxRank limit.

Step 2:

Then the router checks if one of its addresses is included in one of the AODV-RPL Target Options or belongs to the indicated

multicast group. 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 route entry towards its preferred parent. The route entry SHOULD be stored along with the associated RPLInstanceID and DODAGID. 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:

If there are multiple AODV-RPL Target Options in the received RREQ-DIO, a TargNode SHOULD continue sending RREQ-DIO to reach other targets. When preparing its own RREQ-DIO, the TargNode MUST delete the AODV-RPL Target Option related to its own address, so that the routers which higher ranks would know the route to this target has already been found. When an intermediate router receives several RREQ-DIOs which include different lists of AODV-RPL Target Options, the intersection of these lists will be included in its own RREQ-DIO. If the intersection is empty, 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.

Step 5:

For an intermediate router, it sends out its own RREQ-DIO via link-local multicast. For a TargNode, it can begin to prepare the RREP-DIO.

6.3. Generating Route Reply at TargNode

6.3.1. RREP-DIO for Symmetric route

When a RREQ-DIO arrives at a TargNode with the 'S' bit set to 1, it means there exists a symmetric route in which the two directions can fulfill the requirements. Other RREQ-DIOs can bring the upward direction of asymmetric routes (i.e. S=0). How to choose between a qualified symmetric route and an asymmetric route hopefully having better performance is implementation-specific and out of scope. If the implementation choose to use the symmetric route, the TargNode MAY send out the RREP-DIO after a duration RREP_WAIT_TIME to wait for the convergence of RD to an optimal symmetric route.

For symmetric route, the RREP-DIO message is sent via unicast to the OrigNode; therefore the DODAG in RREP-Instance doesn't need to be actually built. The RPLInstanceID in the RREP-Instance is paired as defined in Section 6.3.3. The 'S' bit in the base DIO remains as 1. In the RREP option, The 'SHIFT' field and the 'T' bit are set as defined in Section 6.3.3. The address vector received in the RREQ-DIO MUST be included in this RREP option in case the 'H' bit is set to 0 (both in RREQ-DIO and RREP-DIO). If the 'T' bit is set to 1, the address of the OrigNode MUST be encapsulated in an AODV-RPL Target Option and included in this RREP-DIO message, and the Destination Sequence Number is set according to 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 send out via link-local multicast until the OrigNode is reached or the MaxRank limit is exceeded.

The settings of the RREP-DIO are the same as in symmetric route.

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 (RPLInstanceID, DODAGID, Address in the AODV-RPL Target Option) is needed to uniquely identify a DODAG in an AODV-RPL instance. Between the OrigNode and the TargNode, there can be multiple AODV-RPL instances when applications upper layer have different requirements. Therefore the RREQ-Instance and the RREP-Instance in the same route discovery MUST be paired. The way to realize this is to pair their RPLInstance IDs.

Typically, the two InstanceIDs are set as the local InstanceID in core RPL:

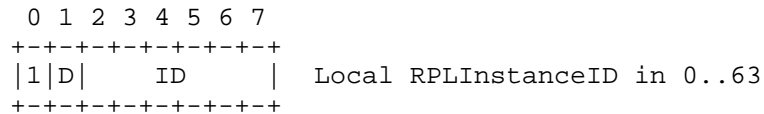


Figure 6: Local Instance ID

The first bit is set to 1 indicating the RPLInstanceID is local. The 'D' bit here is used to distinguish the two AODV-RPL instances: D=0 for RREQ-Instance, D=1 for RREP-Instance. The ID of 6 bits SHOULD be

the same for RREQ-Instance and RREP-Instance. Here, the 'D' bit is used slightly differently than in RPL.

When preparing the RREP-DIO, a TargNode could find the RPLInstanceID to be used for the RREP-Instance is already occupied by another instance from an earlier route discovery operation which is still active. In other words, two 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 this RPLInstanceID SHOULD be shifted into another integer and shifted back to the original one at the OrigNode. In RREP option, the SHIFT field indicates the how many the original RPLInstanceID is shifted. When the new InstanceID after shifting exceeds 63, it will come back counting from 0. For example, the original InstanceID is 60, and shifted by 6, the new InstanceID will be 2. The 'T' MUST be set to 1 to make sure the two RREP-DIOs can be distinguished by the address of the OrigNode in the AODV-RPL Target Option.

6.4. Receiving and Forwarding Route Reply

Upon receiving a RREP-DIO, a router out of the RREP-Instance goes through the following steps:

Step 1:

If the 'S' bit of the RREP-DIO is set to 0, the router MUST look into the downward direction of the link (towards the TargNode) by 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 be inferior to the MaxRank limit, the router chooses to join in the DODAG of the RREP-Instance. The router issuing the received RREP-DIO is selected as the preferred parent. Afterwards, other RREQ-DIO messages can be received. How to maintain the parent set, select the preferred parent, and update the router's rank follows the core RPL and the OFs defined in ROLL WG.

If the constraints are not fulfilled, the router MUST NOT join in the DODAG, and will not go through steps 2, 3, and 4.

A router MUST discard a received RREQ-DIO if the advertised rank equals or exceeds the MaxRank limit.

If the 'S' bit is set to 1, the router does nothing in this step.

Step 2:

Then the router 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 route entry including the RPLInstanceID of RREP-Instance and the DODAGID. For symmetric route, the route entry is to the router from which the RREP-DIO is received. For asymmetric route, the route entry is to the preferred parent in the DODAG of RREQ-Instance.

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

Step 4:

For an intermediate router, in case of asymmetric route, the RREP-DIO is sent out via link-local multicast; in case of symmetric route, the RREP-DIO is unicasted to the OrigNode via the next hop in source routing (H=0), or via the next hop in the route entry built in the RREQ-Instance (H=1). For the OrigNode, it can start transmitting the application data to TargNode along the path as discovered through RREP-Instance.

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 forward and reverse 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 address vector between O and itself. Then T MUST unicast a RREP-DIO including the address vector between T and R.

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 7: 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 8: 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>>.
- [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>>.

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.

Appendix A. ETX/RSSI Values to select S bit

We have tested the combination of "RSSI(downstream)" and "ETX (upstream)" to decide 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 9: 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. Changes to version 02

- o Include the support for source routing.
- o Bring some features from [RFC6997], e.g., choice between hop-by-hop and source routing, 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.

Authors' Addresses

Satish Anamalamudi
Huaiyin Institute of Technology
No.89 North Beijing Road, Qinghe District
Huaian 223001
China

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: November 8, 2020

S. Anamalamudi
SRM University-AP
M. Zhang
Huawei Technologies
C. Perkins
Deep Blue Sky Networks
S.V.R.Anand
Indian Institute of Science
B. Liu
Huawei Technologies
May 7, 2020

AODV based RPL Extensions for Supporting Asymmetric P2P Links in Low-
Power and Lossy Networks
draft-ietf-roll-aodv-rpl-08

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 (AODV-RPL). 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 November 8, 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
2. Terminology	3
3. Overview of AODV-RPL	5
4. AODV-RPL DIO Options	6
4.1. AODV-RPL RREQ Option	6
4.2. AODV-RPL RREP Option	8
4.3. AODV-RPL 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	17
6.4. Receiving and Forwarding Route Reply	17
7. Gratuitous RREP	19
8. Operation of Trickle Timer	19
9. IANA Considerations	20
9.1. New Mode of Operation: AODV-RPL	20
9.2. AODV-RPL Options: RREQ, RREP, and Target	20
10. Security Considerations	20
11. Link State Determination	21
12. References	21
12.1. Normative References	21
12.2. Informative References	22
Appendix A. Example: ETX/RSSI Values to select S bit	23
Appendix B. Changelog	24
B.1. Changes from version 07 to version 08	24

B.2.	Changes from version 06 to version 07	24
B.3.	Changes from version 05 to version 06	25
B.4.	Changes from version 04 to version 05	25
B.5.	Changes from version 03 to version 04	25
B.6.	Changes from version 02 to version 03	25
Appendix C.	Contributors	26
Authors' Addresses	26

1. Introduction

RPL [RFC6550] (Routing Protocol for Low-Power and Lossy Networks) 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., 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 (for more information see [RFC6997], [RFC6998]).

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.

AODV-RPL reuses and provides a natural extension to the core RPL functionality to support routes with bidirectional asymmetric links. It retains RPL's DODAG formation, RPL Instance and the associated Objective Function, trickle timers, and support for storing and non-storing modes. AODV adds basic messages RREQ and RREP as part of RPL DIO (DODAG Information Object) control messages, and does not utilize the DAO message of RPL. AODV-RPL specifies a new MOP running in a separate instance dedicating to discover P2P routes, which may differ from the P2MP routes discoverable by native RPL. AODV-RPL can be operated whether or not native RPL is running otherwise.

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

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 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 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.

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 is thus functional 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. 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 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, otherwise it SHOULD be dropped.

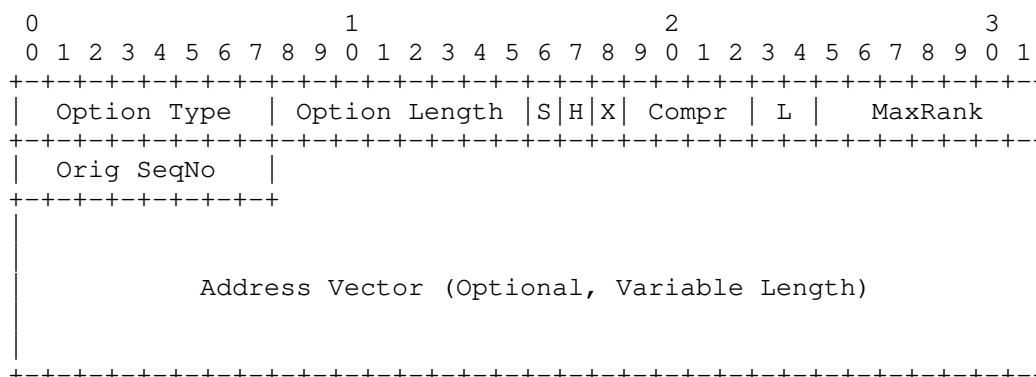


Figure 1: Format for AODV-RPL RREQ Option

OrigNode supplies the following information in the RREQ option:

Option Type
TBD2

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.

- * 0x00: No time limit imposed.
- * 0x01: 16 seconds
- * 0x02: 64 seconds
- * 0x03: 256 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 node no longer maintains DAG connectivity or messaging.

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. 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 equal to the MaxRank. Other nodes MUST NOT join a RREQ instance if its own Rank would be equal to or higher than MaxRank. A router MUST discard a received RREQ if the integer part of the advertised Rank equals or exceeds the MaxRank limit.

4.2. AODV-RPL 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, otherwise the message SHOULD be dropped. TargNode supplies the following information in the RREP option:

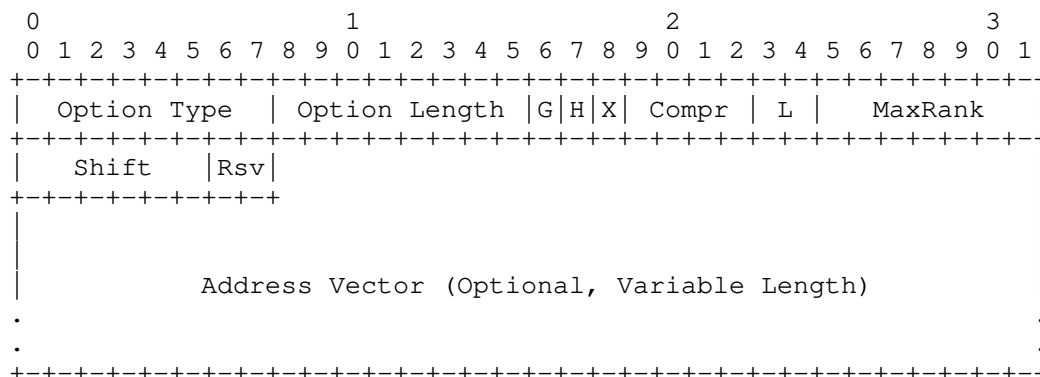


Figure 2: Format for AODV-RPL RREP option

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
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 RPLInstanceID (see Section 6.3.3); 0 indicates that the original RPLInstanceID 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 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 SHOULD 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.

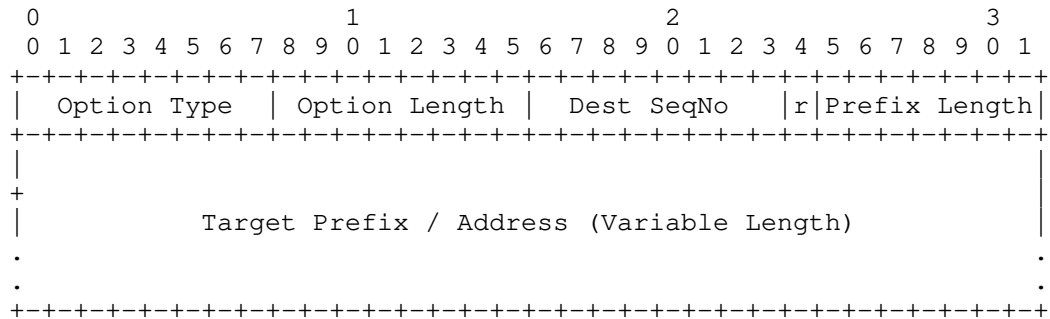


Figure 3: Target option format for AODV-RPL MOP

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 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.

r

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 length of the field is the least number of octets that can contain all of the bits of the Prefix, in other words $\text{Floor}((7+(\text{Prefix Length}))/8)$ octets. The remaining bits in the Target Prefix / Address field after the prefix length (if any) MUST be set to zero on transmission and MUST be ignored on receipt.

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.

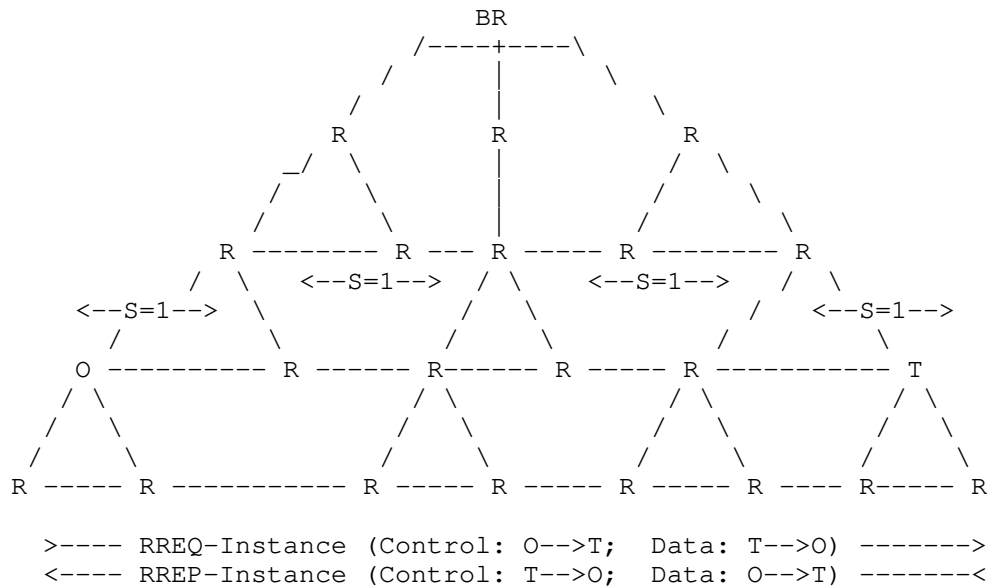


Figure 4: AODV-RPL with Symmetric Paired Instances

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). 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.

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 can 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.

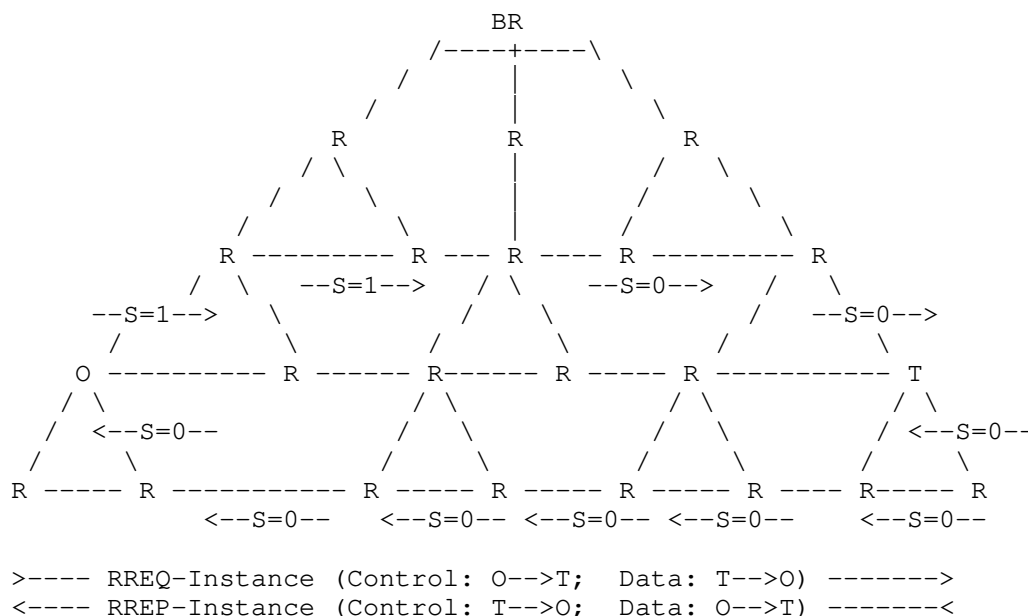


Figure 5: AODV-RPL with Asymmetric Paired Instances

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 that satisfies the Objective Function for the target 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 ART Option (see Section 4.3). 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,

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 [RFC6206]. 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 goes through the steps below. If the router does not belong to the RREQ-Instance, then the maximum useful rank (MaxUseRank) is MaxRank. Otherwise, MaxUseRank 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.

Step 1:

If the S bit in the received RREQ-DIO is set to 1, the router MUST determine whether each direction of the link (by which the RREQ-DIO is received) satisfies the Objective Function. In case that the downward (i.e. towards the TargNode) direction of the link does not satisfy the Objective Function, 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 MUST only check into the upward direction (towards the OrigNode) of the link.

If the upward direction of the link can satisfy the Objective Function (defined in [RFC6551]), and the router's Rank would not exceed the MaxUseRank limit, the router joins the DODAG of the RREQ-Instance. The router that transmitted the received RREQ-DIO is selected as the preferred parent. Otherwise, if the Objective Function is not satisfied or the MaxUseRank limit is exceeded, 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 ART 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 an upward route entry towards OrigNode which MUST include at least the following items: Source Address, InstanceID, Destination Address, Next Hop, Lifetime, and Sequence Number. The Destination Address and the InstanceID respectively can be learned from the DODAGID and the RPLInstanceID of the RREQ-DIO, and the Source Address is the address used by the local router to send data to the OrigNode. The Next Hop is the preferred parent. The lifetime is set according to DODAG configuration (i.e., not the L bit) and can be extended when the route is actually used. The sequence number represents the freshness of the route entry, and it is copied from the Orig SeqNo field of the RREQ option. A route entry with the same source and destination address, same InstanceID, but stale sequence number, MUST be deleted.

Step 4:

If the router is an intermediate router, then it transmits a RREQ-DIO via link-local multicast; if the H bit is set to 0, the intermediate router MUST include the address of the interface receiving the RREQ-DIO into the address vector.. Otherwise, if the router (i.e., TargNode) was not already associated with the RREQ-Instance, it prepares a RREP-DIO Section 6.3. If, on the other hand TargNode was already associated with the RREQ-Instance, it takes no further action and does not send an 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 MUST 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 Rank values do not try to create a route to this TargetNode.

An intermediate router could receive several RREQ-DIOs from routers with lower Rank values in the same RREQ-Instance but have different lists of Target Options. When rebroadcasting the RREQ-DIO, the intersection of these lists MUST 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 MUST NOT send out any RREQ-DIO. 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 associated with the RREQ-Instance. Any RREQ-DIO message with different ART 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 satisfy the Objective Function. 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 selects the symmetric route, and the L bit is not 0, the TargNode MAY delay transmitting the RREP-DIO for duration RREP_WAIT_TIME to await a symmetric route with a lower Rank. The value of RREP_WAIT_TIME is set by default to 1/4 of the time duration determined by the L bit.

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. TargNode increments its current sequence number and uses the incremented result in the Dest SeqNo in the ART option of the RREQ-DIO. The address of the OrigNode MUST be encapsulated in the ART Option and included in this RREP-DIO message.

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 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 time duration determined by the L bit.

The settings of the fields in RREP option and ART option are the same as for the 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. 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. To avoid any mismatch, the RREQ-Instance and the RREP-Instance in the same route discovery MUST be paired 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 MUST proceed 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 satisfy the Objective Function, 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.

If the Objective Function is not satisfied, 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 ART 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 MUST include at least the following items: OrigNode Address, InstanceID, 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 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 ART 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. 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 InstanceID, but stale sequence number, SHOULD be deleted.

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. The local knowledge for the TargNode's sequence number SHOULD be updated.

Upon receiving a RREP-DIO, a router which already belongs to the RREQ-Instance SHOULD drop the 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 receives a RREQ-DIO for a TargNode, and the router has a more recent (larger destination sequence number) pair of downward and upward routes to the TargNode which also satisfy the Objective Function.

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 send out the gratuitous RREP-DIO including the address vector from the router itself to the TargNode.

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

The trickle timer operation to control RREQ-Instance/RREP-Instance multicast uses [RFC6206] to control RREQ-DIO and RREP-DIO transmissions. The Trickle control of these DIO transmissions follow the procedures described in the Section 8.3 of [RFC6550] entitled "DIO Transmission".

9. IANA Considerations

9.1. New Mode of Operation: AODV-RPL

IANA is asked to assign a new Mode of Operation, named "AODV-RPL" for Point-to-Point (P2P) hop-by-hop routing from the "Mode of Operation" Registry [RFC6550].

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

Figure 6: Mode of Operation

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

IANA is asked to assign three new AODV-RPL options "RREQ", "RREP" and "ART", as described in Figure 7 from the "RPL Control Message Options" Registry [RFC6550].

Value	Meaning	Reference
TBD2 (0x0A)	RREQ Option	This document
TBD3 (0x0B)	RREP Option	This document
TBD4 (0x0C)	ART Option	This document

Figure 7: AODV-RPL Options

10. Security Considerations

In general, 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 security framework, which provides data confidentiality, authentication, replay protection, and delay protection services.

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, 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 a secure P2P-RPL 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. In this type of scenario, RPL's authenticated mode of operation, where a node can obtain the key to use for a P2P-RPL route discovery only after proper authentication, SHOULD be used.

When RREQ-DIO message uses source routing option with 'H' set to 0, some of the security concerns that led to the deprecation of Type 0 routing headers [RFC5095] may apply. To avoid the possibility of a RREP-DIO message traveling in a routing loop, if one of its addresses are present as part of the Source Route listed inside the message, the Intermediate Router MUST NOT forward the message.

11. Link State Determination

This document specifies that links are considered symmetric until additional information is collected. 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.

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>>.

- [RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, DOI 10.17487/RFC5095, December 2007, <<https://www.rfc-editor.org/info/rfc5095>>.
- [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>>.
- [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>>.
- [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, Jan 2018.
- [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>>.

- [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>>.
- [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>>.

Appendix A. Example: ETX/RSSI Values to select 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 nodes. ETX and RSSI values may be used in conjunction as explained below:

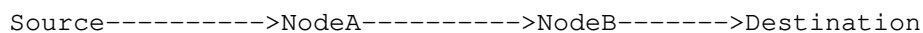


Figure 8: Communication link from Source to Destination

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	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 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. Later on, the link from NodeA to Destination is asymmetric with S bit remains set to 0.

Appendix B. Changelog

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

B.1. Changes from version 07 to version 08

- o Instead of describing the need for routes to "fulfill the requirements", specify that routes need to "satisfy the Objective Function".
- o Removed all normative dependencies on [RFC6997]
- o Rewrote Section 10 to avoid duplication of language in cited specifications.
- o Added Section 11 with text and citations to more fully describe how implementations determine whether links are symmetric.
- o Modified text comparing AODV-RPL to other protocols to emphasize the need for AODV-RPL instead of the problems with the other protocols.
- o Clarified that AODV-RPL uses some of the base RPL specification but does not require an instance of RPL to run.
- o Improved capitalization, quotation, and spelling variations.
- o Specified behavior upon reception of a RREQ-DIO or RREP-DIO message for an already existing DODAGID (e.g, Section 6.4).
- o Fixed numerous language issues in IANA Considerations Section 9.
- o For consistency, adjusted several mandates from SHOULD to MUST and from SHOULD NOT to MUST NOT.
- o Numerous editorial improvements and clarifications.

B.2. Changes from version 06 to version 07

- o 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.

- o Modified the language from [RFC6550] Target Option definition so that the trailing zero bits of the Prefix Length are no longer described as "reserved".
 - o Reclassified [RFC3561] and [RFC6998] as Informative.
 - o Added citation for [RFC8174] to Terminology section.
- B.3. Changes from version 05 to version 06
- o Added Security Considerations based on the security mechanisms defined in [RFC6550].
 - o Clarified the nature of improvements due to P2P route discovery versus bidirectional asymmetric route discovery.
 - o Editorial improvements and corrections.
- B.4. Changes from version 04 to version 05
- o Add description for sequence number operations.
 - o Extend the residence duration L in section 4.1.
 - o Change AODV-RPL Target option to ART option.
- B.5. Changes from version 03 to version 04
- 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.
- B.6. Changes from version 02 to version 03
- 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.

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

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

Charles E. Perkins
Deep Blue Sky Networks
Saratoga 95070
United 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 Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2018

O. Bergmann
C. Bormann
S. Gerdes
Universitaet Bremen TZI
H. Chen
Huawei Technologies
October 30, 2017

Constrained-Cast: Source-Routed Multicast for RPL
draft-ietf-roll-ccast-01

Abstract

This specification defines a protocol for forwarding multicast traffic in a constrained node network employing the RPL routing protocol in non-storing mode.

Status of This Memo

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

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

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

This Internet-Draft will expire on May 3, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Terminology	3
2. The BIER Approach	3
3. The Constrained-Cast Approach	3
4. False Positives	4
5. Protocol	4
5.1. Multicast Listener Advertisement Object (MLAO)	4
5.2. Routing Header	5
6. Implementation	7
7. Benefits	7
8. Security Considerations	7
9. IANA Considerations	8
9.1. ICMPv6 Parameter Registration	8
9.2. Critical 6LowPAN Routing Header Type Registration	8
10. Acknowledgments	8
11. References	8
11.1. Normative References	8
11.2. Informative References	9
Authors' Addresses	9

1. Introduction

As defined in [RFC6550], RPL Multicast assumes that the RPL network operates in Storing Mode. Multicast DAOs are used to indicate subscription to multicast address to a parent; these DAOs percolate up and create bread-crumbs. This specification, although part of RFC 6550, appears to be incomplete and untested. More importantly, Storing Mode is not in use in constrained node networks outside research operating environments.

The present specification addresses multicast forwarding for RPL networks in the much more common Non-Storing Mode. Non-Storing is based on the root node adding source-routing information to downward packets. Evidently, to make this work, RPL multicast needs to source-route multicast packets. A source route here is a list of identifiers to instruct forwarders to relay the respective IP datagram.

As every forwarder in an IP-based constrained node network has at least one network interface, it is straight-forward to use the address of an outgoing interface as identifiers in this source-route. (Typically, this is a globally unique public address of the node's only network adapter.)

The source-route subsets the whole set of potential forwarders available in the RPL DODAG to those that need to forward in order to reach known multicast listeners.

Including an actual list of outgoing interfaces is rarely applicable, as this is likely to be a large list of 16-byte IPv6 addresses. Even with [RFC6554] style compression, the size of the list becomes prohibitively quickly.

1.1. Terminology

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

In this specification, the term "byte" is used in its now customary sense as a synonym for "octet".

All multi-byte integers in this protocol are interpreted in network byte order.

2. The BIER Approach

Bit-Indexed Explicit Replication [I-D.ietf-bier-architecture] lists all egress routers in a bitmap included in each multicast packet. This requires creating a mostly contiguous numbering of all egress routers; more importantly, BIER requires the presence of a network map in each forwarders to be able to interpret the bitmap and map it to a set of local outgoing interfaces.

3. The Constrained-Cast Approach

Constrained-Cast employs Bloom Filters [BLOOM] as a compact representation of a match or non-match for elements in a large set: Each element to be included is hashed with multiple hash functions; the result is used to index a bitmap and set the corresponding bit. To check for the presence of an element, the same hash functions are applied to obtain bit positions; if all corresponding bits are set, this is used to indicate a match. (Multiple hash functions are most easily obtained by adding a varying seed value to a single hash algorithm.)

By including a bloom filter in each packet that matches all outgoing interfaces that need to forward the packet, each forwarder can efficiently decide whether (and on which interfaces) to forward the packet.

4. False Positives

Bloom filters are probabilistic. A false positive might be indicating a match where the bits are set by aliasing of the hash values. In case of Constrained-Cast, this causes spurious transmission and wastes some energy and radio bandwidth. However, there is no semantic damage (hosts still filter out unneeded multicasts). The total waste in energy and spectrum can be visualized as the false-positive-rate multiplied by the density of the RPL network. A network can easily live with a significant percentage of false positives. By changing the set of hash functions (i.e., seed) over time, the root can avoid a single node with a false positive to become an unnecessary hotspot for that multicast group.

5. Protocol

The protocol uses DAO-like "MLAO" messages to announce membership to the root as specified in Section 5.1.

For downward messages, the root adds a new routing header that includes a hash function identifier and a seed value; another one of its fields gives the number of hash functions (k) to ask for k instances of application of the hash function, with increasing seed. The format of the new routing header is specified in Section 5.2.

Typical sizes of the bloom filter bitmap that the root inserts into the packet can be 64, 128, or 256 bit, which may lead to acceptable false positive rates if the total number of forwarders in the 10s and 100s. (To do: write more about the math here. Note that this number tallies forwarding routers, not end hosts.)

A potential forwarder that receives a multicast packet adorned with a constrained-cast routing header first checks that the packet is marked with a RPL rank smaller than its own (loop prevention). If yes, it then forwards the packet to all outgoing interfaces that match the bloom filter in the packet.

5.1. Multicast Listener Advertisement Object (MLAO)

The header format of the MLAO is depicted in Figure 1. The basic structure of the MLAO message is similar to the RPL Destination Advertisement Object (DAO). In particular, it starts with RPL ICMP base header with a type value of 155 and the code {IANA TBD1} (MLAO), followed by the Checksum, RPLInstanceID, parameters and flags as in a DAO.

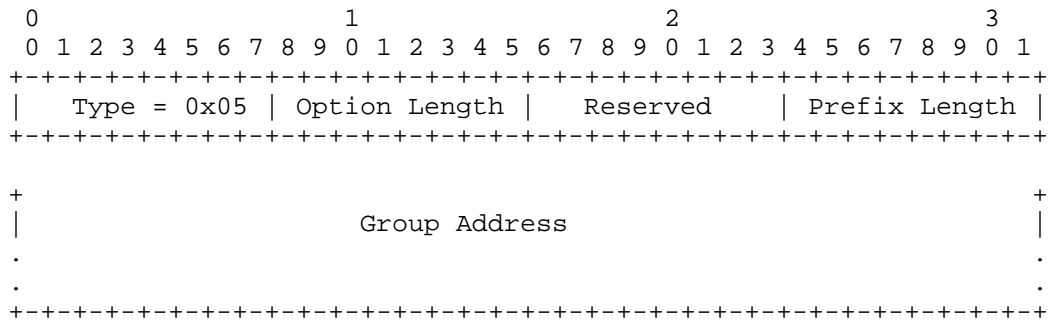


Figure 1: RPL Target Option for MLA0

The group address field indicates the group that the sender of the MLA0 is interested in. This field usually contains a 128 bit IPv6 multicast group address. Shorter group identifiers could be used together with a protocol for explicit creation of groups. The MLA0 message must have at least one RPL target option to specify the address of the listener that has generated the MLA0. The message is directed to the global unicast address of the DODAG root and travels upwards the routing tree.

Note: It has been suggested to use the RPL Transit Option (Type 0x06) instead as it is used in Non-Storing mode to inform the DODAG root of path attributes. Specifically, this option can be used to limit the subscription by providing a proper Path Lifetime.

5.2. Routing Header

This specification uses a new Source Routing 6LowPAN Routing Header (SRH-6LoRH) type [RFC8138] to convey the Bloom filter that describes the source route for the IPv6 multicast packet to take within the RPL routing tree. The 6LoRH Type for this Constrained Cast Routing Header (CCRH) is set to TBD7. Figure 2 depicts the format of this new routing header.

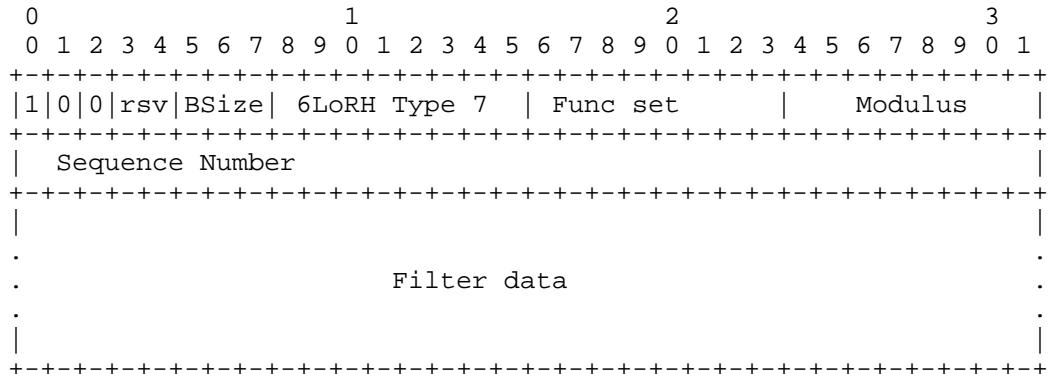


Figure 2: Routing header

rsv: This field is reserved for future use and MUST be set to 0 when sending a packet containing a CCRH. A receiver MUST ignore the value of the rsv field.

BSize: Specifies the size of the included Bloom filter. Currently, only 64 bits, 128 bits, or 256 bits are supported. The BSize field is encoded as specified in Table 1.

BSize value	Filter Size (Bits)
0	64
1	128
2	256

Table 1: Possible Bloom Filter Lengths

Func set: The set of hash functions used to generate the Filter data value.

Note: As the function set contains a combination of several distinct hash functions, it is currently unclear if 8 bits number space is large enough.

Modulus: The modulus that is used by the hash functions, minus 64 (the minimum filter data size that can be used). The DAG root chooses the modulus (and thus the filter data size) to achieve its objectives for false positive rates (Section 4).

Sequence Number: 32 bits sequence number. The number space is unique for a sequence of multicast datagrams for a specific group that arrive at the DAG root on their way up. The DAG root increments the number for each datagram it sends down the respective DODAG.

Filter data: A bit field that indicates which nodes should relay this multicast datagram. The length of this field is a multiple of 8 bytes ($2^{(BSize + 3)}$ bytes). The actual length is derived from the contents of the field BSize.

6. Implementation

In 2013, Constrained-Cast was implemented in Contiki. It turns out that forwarders can compute the hash functions once for their outgoing interfaces and then cache them, simply bit-matching their outgoing interface hash bits against the bloom filter in the packet (a match is indicated when all bits in the outgoing interface hash are set in the bloom filter).

The Root computes the tree for each multicast group, computes the bloom filter for it, caches these values, and then simply adds the bloom filter routing header to each downward packet. For adding a new member, the relevant outgoing interfaces are simply added to the bloom filter. For removing a leaving member, however, the bloom filter needs to be recomputed (which can be sped up logarithmically if desired).

7. Benefits

Constrained-Cast:

- o operates in Non-Storing Mode, with the simple addition of a membership information service;
- o performs all routing decisions at the root.

Further optimizations might include using a similar kind of bloom filter routing header for unicast forwarding as well (representing, instead of the outgoing interface list, a list of children that forwarding parents need to forward to).

8. Security Considerations

TODO

9. IANA Considerations

The following registrations are done following the procedure specified in [RFC6838].

Note to RFC Editor: Please replace all occurrences of "[RFC-XXXX]" with the RFC number of this specification and "TBD1" with the code selected for TBD1 below and "TBD7" with the code selected for TBD7.

9.1. ICMPv6 Parameter Registration

IANA is requested to add the following entry to the Code fields of the RPL Control Codes registry:

Code	Name	Reference
TBD1	MLAO	[RFC-XXXX]

9.2. Critical 6LowPAN Routing Header Type Registration

IANA is requested to add the following entries to the Critical 6LowPAN Routing Header Type Registration registry:

Value	Name	Reference
TBD7	CCast Routing Header	[RFC-XXXX]

10. Acknowledgments

Thanks to Yasuyuki Tanaka for valuable comments.

This work has been supported by Siemens Corporate Technology.

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>>.
- [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

- [BLOOM] Bloom, B., "Space/time trade-offs in hash coding with allowable errors", ISSN 0001-0782, ACM Press Communications of the ACM vol 13 no 7 pp 422-426, 1970, <<http://doi.acm.org/10.1145/362686.362692>>.
- [I-D.ietf-bier-architecture] Wijnands, I., Rosen, E., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast using Bit Index Explicit Replication", draft-ietf-bier-architecture-08 (work in progress), September 2017.
- [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>>.
- [RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <<https://www.rfc-editor.org/info/rfc6838>>.

Authors' Addresses

Olaf Bergmann
Universitaet Bremen TZI
Postfach 330440
Bremen D-28359
Germany

Phone: +49-421-218-63904
Email: bergmann@tzi.org

Carsten Bormann
Universitaet Bremen TZI
Postfach 330440
Bremen D-28359
Germany

Phone: +49-421-218-63921
Email: cabo@tzi.org

Stefanie Gerdes
Universitaet Bremen TZI
Postfach 330440
Bremen D-28359
Germany

Phone: +49-421-218-63906
Email: gerdes@tzi.org

Hao Chen
Huawei Technologies
12, E. Mozhou Rd
Nanjing 211111
China

Phone: +86-25-5662-7052
Email: philips.chenhao@huawei.com

ROLL
Internet-Draft
Intended status: Standards Track
Expires: September 20, 2018

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

Root initiated routing state in RPL
draft-ietf-roll-dao-projection-03

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 September 20, 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. New RPL Control Message Options	3
3.1. Via Information Option	4
4. Projected DAO	5
4.1. Non-storing Mode Projected DAO	6
4.2. Storing-Mode Projected DAO	8
5. Applications	10
5.1. Loose Source Routing in Non-storing Mode	10
5.2. Transversal Routes in storing and non-storing modes	11
6. RPL Instances	13
7. Security Considerations	14
8. IANA Considerations	14
9. Acknowledgments	14
10. References	15
10.1. Normative References	15
10.2. Informative References	15
Appendix A. Examples	16
A.1. Using storing mode P-DAO in non-storing mode MOP	16
A.2. Projecting a storing-mode transversal route	17
Authors' Addresses	19

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, with optionally the assistance of a PCE, to install and maintain additional storing and non-storing mode routes within the RPL domain, along a selected set of nodes and for a selected duration, thus providing routes more suitable than those obtained with the distributed operation of RPL. Those routes may be installed in either storing and non-storing modes RPL instances, resulting in potentially hybrid situations where the mode of the projected routes is different from that of the other routes in the instance.

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 Terminology used in this document is consistent with and incorporates that described in "Terminology in Low power And Lossy Networks"[RFC7102] and [RFC6550].

3. New RPL Control Message Options

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 a new Control Message Option, the Via Information option (VIO). Like the TIO, the VIO MUST be preceded by one or more RPL Target options to which it applies. Unlike the TIO, the VIO are not factorized: multiple contiguous Via options indicate an ordered sequence of routers to reach the target(s), presented in the order of the packet stream, source to destination, and in which a routing state must be installed.

The Via Information option MUST contain at least one Via Address.

3.1. Via Information Option

The Via Information option MAY be present in DAO messages, and its format is as follows:

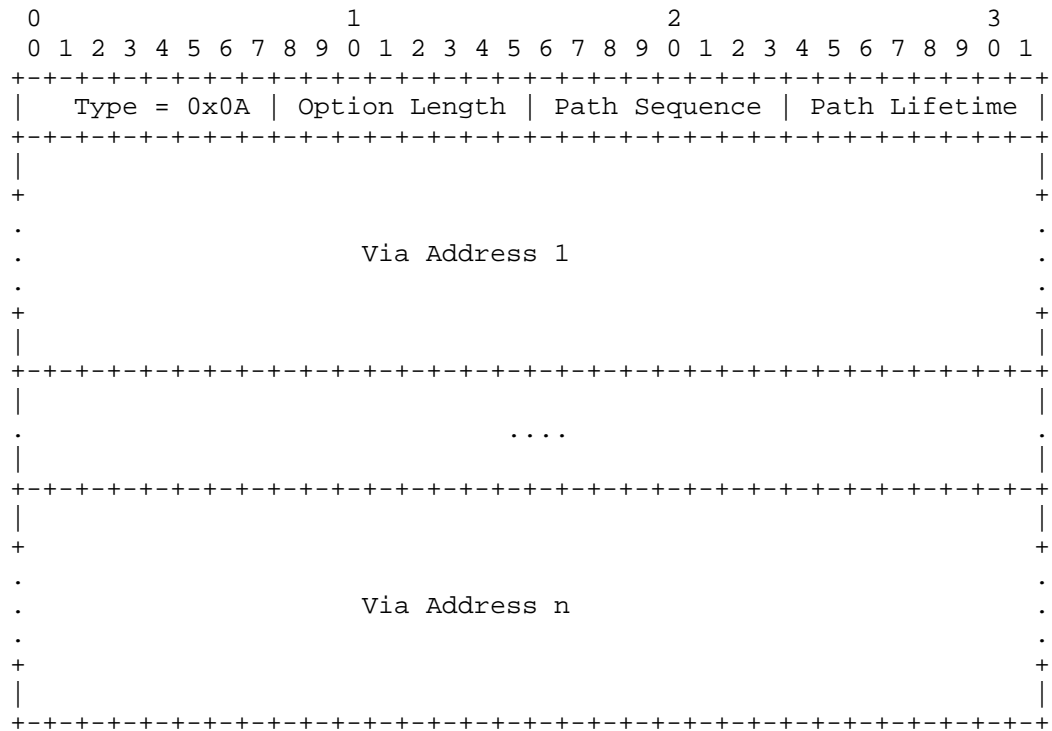


Figure 1: Via Information option format

Option Type: 0x0A (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 VIO. 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, otherwise it is expressed as 16 bytes, at least in storing mode.

4. Projected DAO

This draft adds a capability to RPL whereby the root projects a route through an extended DAO message called a Projected-DAO (P-DAO) to an arbitrary router down the DODAG, indicating a next hop or a sequence of routers via which a certain destination indicated in the Target Information option may be reached.

A P-DAO message MUST contain at least a Target Information option and at least one VIA Information option following it.

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 VIO as opposed to a TIO. Also, a Path Lifetime of 0 in a VIO indicates that a route is to be removed.

There are two kinds of P-DAO, the storing mode and the non-storing mode ones.

The non-storing mode P-DAO discussed in section Section 4.1 has a single VIO with one or more Via Addresses in it, the list of Via Addresses indicating the source-routed path to the target to be installed in the router that receives the message, which replies to the root directly with a DAO-ACK message.

The storing mode P-DAO discussed in section Section 4.2 has at least two Via Information options with one Via Address each, for the ingress and the egress of the path, and more if there are intermediate routers. 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 sequence of 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.

The root is expected to use these mechanisms optimally and with required parsimony to limit the state installed in the devices to fit within their resources, but how the root figures the amount of resources that is available in each device is out of scope for this document.

In particular, the draft expects that the root has enough information about the capability for each node to store a number of routes, which can be discovered for instance using a Network Management System (NMS) and/or the RPL routing extensions specified in "Routing for Path Calculation in LLNs" [RFC6551].

A route that is installed by a P-DAO is not necessarily installed along the DODAG, though how the root and the optional PCE obtain the additional topological information to compute other routes is out of scope for this document

4.1. Non-storing Mode Projected DAO

As illustrated in Figure 2, the non-storing mode P-DAO 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 path to any packet for which the current destination either is the said target or can be reached via the 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.

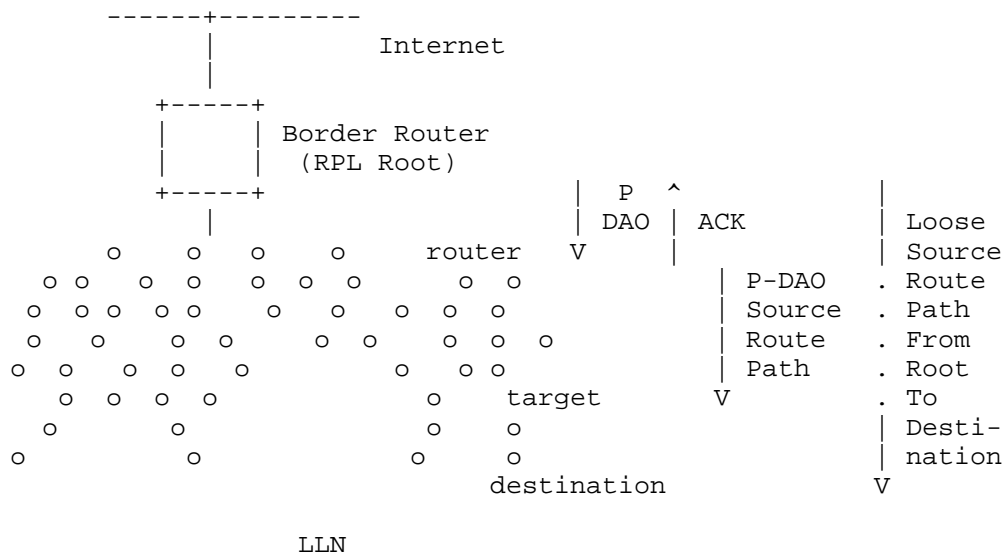


Figure 2: Projecting a Non-Storing route

A router that receives a non-storing P-DAO installs a source routed path towards each of the consecutive targets via a source route path indicated in the following VIO.

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 order to do so, 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 VIO, 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.

4.2. Storing-Mode Projected DAO

As illustrated in Figure 3, the storing mode P-DAO enables 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.

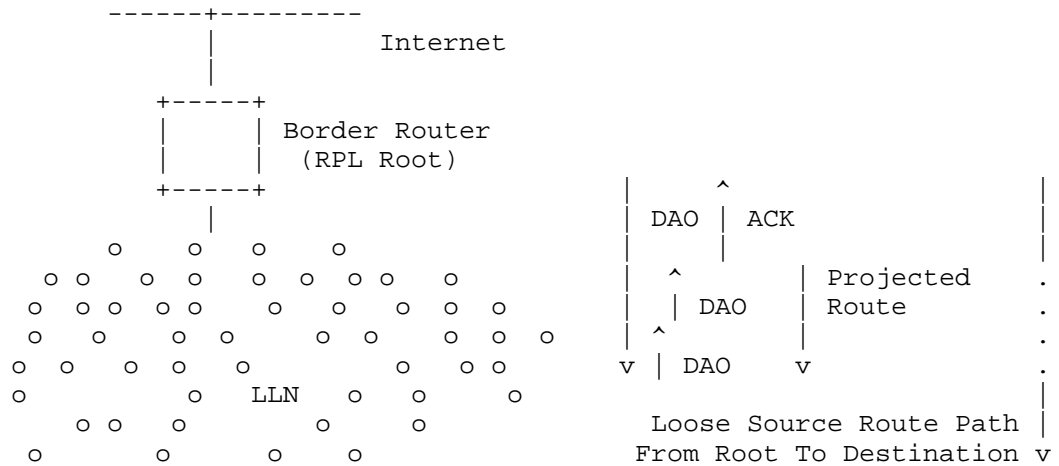


Figure 3: Projecting a route

Based on available topological, usage and capabilities node information, the root or an associated PCE computes which segment should be optimized and which relevant state should be installed in which nodes. The algorithm is out of scope but it is envisaged that the root could compute the ratio between the optimal path (existing path not traversing the root, and the current path), the application service level agreement (SLA) for specific flows that could benefit from shorter paths, the energy wasted in the network, local congestion on various links that would benefit from having flows routed along alternate paths.

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, which 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 that precedes the one that contains 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.

5. Applications

5.1. 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.

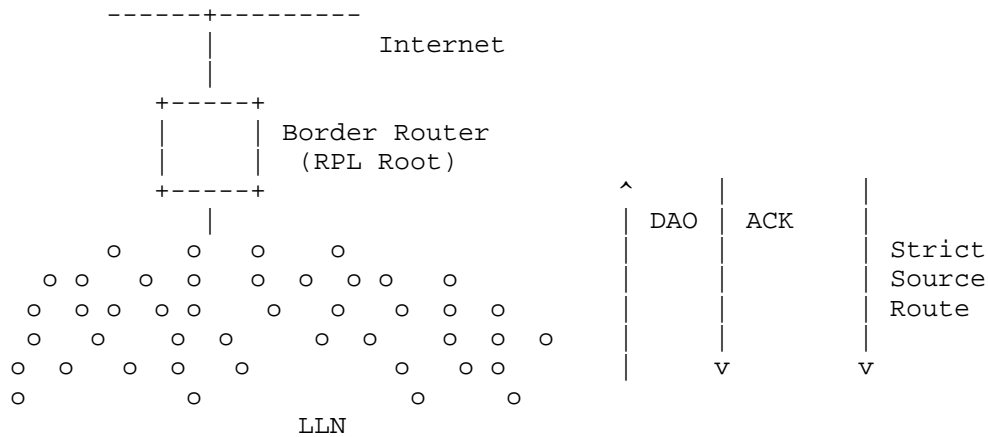


Figure 4: RPL non-storing mode of operation

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.

5.2. 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.

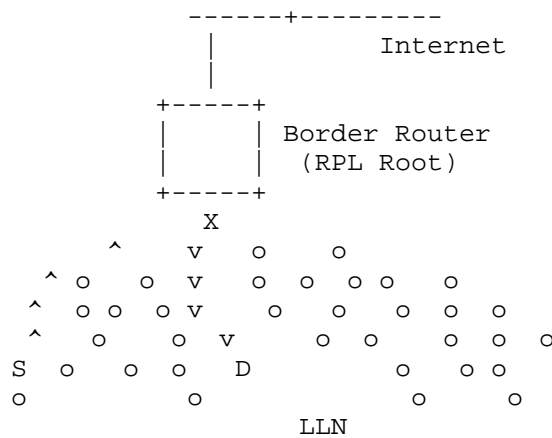


Figure 5: Routing Stretch between S and D via common parent X

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.

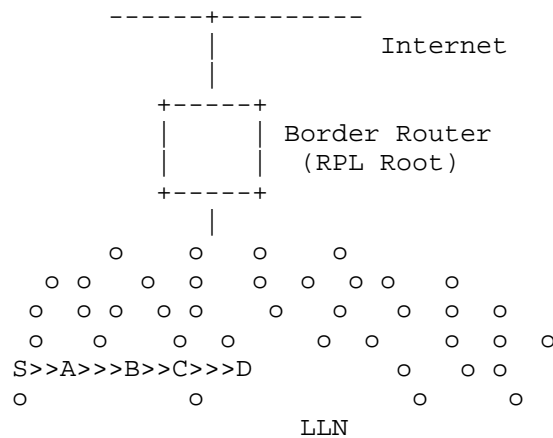


Figure 6: Projected Transversal Route

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.

6. 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.

7. 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.

8. 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

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

9. Acknowledgments

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

10. References

10.1. 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>>.

10.2. 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.

[I-D.ietf-detnet-architecture]

Finn, N., Thubert, P., Varga, B., and J. Farkas,
"Deterministic Networking Architecture", draft-ietf-
detnet-architecture-04 (work in progress), October 2017.

[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.

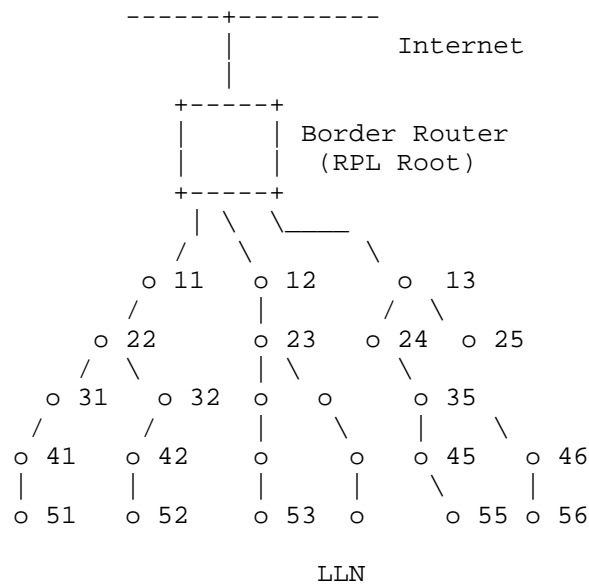


Figure 7: Example DODAG forming a logical tree topology

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 indicating target 55 and a Via segment (35, 45), as well as another DAO to node 46 indicating target 56 and a Via segment (35, 46). This will save one entry in the routing header on both sides. The root may then send a DAO to node 35 indicating targets 55 and 56 a Via segment (13, 24, 35) to fully optimize that path.

Alternatively, the root may send a DAO to node 45 indicating target 55 and a Via segment (13, 24, 35, 45) and then a DAO to node 46 indicating target 56 and a Via segment (13, 24, 35, 46), indicating the same DAO Sequence.

A.2. Projecting a storing-mode transversal route

In this example, say that a PCE determines that a path must be installed between node S and node D via routers A, B and C, in order to serve the needs of a particular application.

The root sends a P-DAO with a target option indicating the destination D and a sequence Via Information option, one for S, which is the ingress router of the segment, one for A and then for B, which are an intermediate routers, and one for C, which is the egress router.

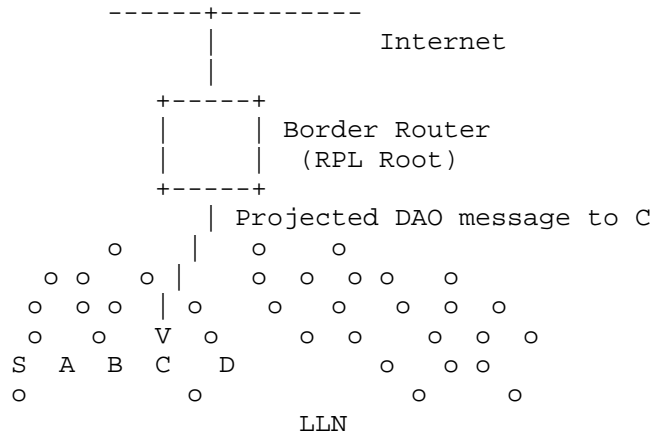


Figure 8: Projected DAO from root

Upon reception of the P-DAO, C validates that it can reach D, e.g. using IPv6 Neighbor Discovery, and if so, propagates the P-DAO unchanged to B.

B checks that it can reach C and of so, installs a route towards D via C. Then it propagates the P-DAO to A.

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.

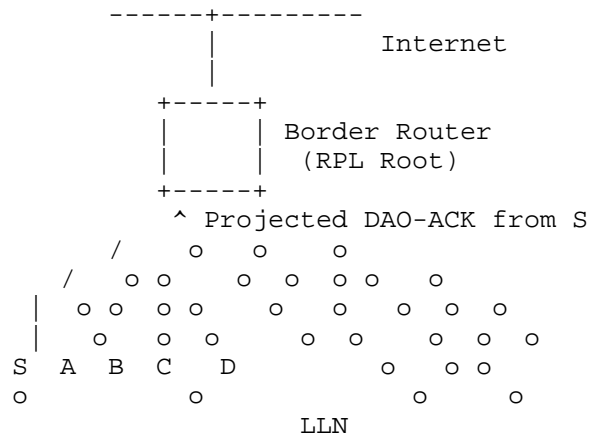


Figure 9: Projected DAO-ACK to root

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

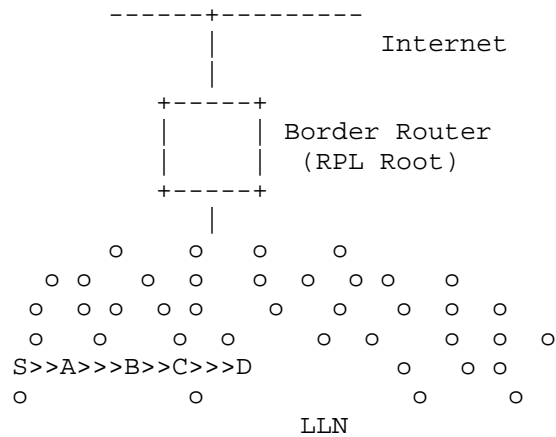


Figure 10: Projected Transversal Route

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 (editor)
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: 31 May 2021

P. Thubert, Ed.
Cisco Systems
R.A. Jadhav
Huawei Tech
M. Gillmore
Itron
27 November 2020

Root initiated routing state in RPL
draft-ietf-roll-dao-projection-15

Abstract

This document extends RFC 6550 and RFC 6553 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 31 May 2021.

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
2. Terminology	5
2.1. Requirements Language	5
2.2. Glossary	5
2.3. Other Terms	6
2.4. References	6
3. Extending RFC 6550	6
3.1. Projected DAO	6
3.2. Sibling Information Option	8
3.3. P-DAO Request	8
3.4. Extending the RPI	8
4. Extending RFC 6553	8
5. Extending RFC 8138	9
6. New RPL Control Messages and Options	10
6.1. New P-DAO Request Control Message	10
6.2. New PDR-ACK Control Message	11
6.3. Via Information Options	12
6.4. Sibling Information Option	15
7. Projected DAO	16
7.1. Requesting a Track	18
7.2. Identifying a Track	18
7.3. Installing a Track	19
7.4. Forwarding Along a Track	20
7.5. Non-Storing Mode Projected Route	21
7.6. Storing Mode Projected Route	23
8. Security Considerations	25
9. IANA Considerations	25
9.1. New Elective 6LoWPAN Routing Header Type	25
9.2. New Critical 6LoWPAN Routing Header Type	25
9.3. New Subregistry For The RPL Option Flags	26
9.4. New RPL Control Codes	26
9.5. New RPL Control Message Options	27
9.6. SubRegistry for the Projected DAO Request Flags	27
9.7. SubRegistry for the PDR-ACK Flags	28
9.8. Subregistry for the PDR-ACK Acceptance Status Values	28
9.9. Subregistry for the PDR-ACK Rejection Status Values	28
9.10. SubRegistry for the Via Information Options Flags	29
9.11. SubRegistry for the Sibling Information Option Flags	29
9.12. Error in Projected Route ICMPv6 Code	30
10. Acknowledgments	30

11. Normative References	30
12. Informative References	31
Appendix A. Applications	32
A.1. Loose Source Routing	32
A.2. Transversal Routes	34
Authors' Addresses	36

1. Introduction

RPL, the "Routing Protocol for Low Power and Lossy Networks" [RPL] (LLNs), 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. 6TiSCH uses RPL for its routing operations.

The "6TiSCH Architecture" [6TiSCH-ARCHI] also leverages the "Deterministic Networking Architecture" [RFC8655] centralized 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. With DetNet and 6TiSCH, the component of the controller that is responsible of computing routes is called a Path Computation Element ([PCE]).

Based on heuristics of usage, path length, and knowledge of device capacity and available resources such as battery levels and reservable buffers, the PCE with a global visibility on the system can compute direct Peer to Peer (P2P) routes that are optimized for the needs expressed by an objective function. This document specifies protocol extensions to RPL [RPL] that enable the Root of a main DODAG to install centrally-computed routes inside the DODAG on behalf of a PCE.

This specification expects that the main RPL Instance is operated in RPL Non-Storing Mode of Operation (MOP) to sustain the exchanges with the Root. In that Mode, the Root has enough information to build a basic DODAG topology based on parents and children, but lacks the knowledge of siblings. This document adds the capability for nodes to advertise sibling information in order to improve the topological awareness of the Root.

As opposed to the classical RPL operations where routes are injected by the Target nodes, the protocol extensions enable the Root of a DODAG to project the routes that are needed onto the nodes where they

should be installed. This specification uses the term Projected Route to refer to those routes. Projected Routes can be used to reduce the size of the source routing headers with loose source routing operations down the main RPL DODAG. Projected Routes can also be used to build transversal routes for route optimization and Traffic Engineering purposes, between nodes of the DODAG.

A Projected Route may be installed in either Storing and Non-Storing Mode, potentially resulting in hybrid situations where the Mode of the Projected Route is different from that of the main RPL Instance. A Projected Route may be a stand-alone end-to-end path or a Segment in a more complex forwarding graph called a Track.

The concept of a Track was introduced in the 6TiSCH architecture, as a potentially complex path with redundant forwarding solutions along the way. With this specification, a Track is a DODAG formed by a RPL local Instance that is rooted at the Track Ingress. 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.

The "Reliable and Available Wireless (RAW) Architecture/Framework" [RAW-ARCHI] defines 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 is a dataplane extension of the PCE; it controls the forwarding operation of the packets within a Track, using Packet ARQ, Replication, Elimination, and Overhearing (PAREO) functions over the Track segments, to provide a dynamic balance between the reliability and availability requirements of the flows and the need to conserve energy and spectrum.

The time scale at which the PCE (re)computes the Track can be long, using long-term statistical metrics to perform global optimizations at the scale of the whole network. Conversely, the PSE makes forwarding decisions at the time scale of one or a small collection of packets, based on a knowledge that is limited in scope to the Track itself, so it can be refreshed at a fast pace.

Projected Routes must be used with the parsimony to limit the amount of state that is installed in each device to fit within the device resources, and to maintain the amount of rerouted traffic 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 document.

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 PCE exchanges control messages with the Root over a Southbound API that is out of scope for this specification. The algorithm to compute the paths and the protocol used by an external PCE to obtain the topology of the network from the Root are also out of scope.

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 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. Glossary

This document 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)
LLN: Low-Power and Lossy Network
NMPR: Non-Storing Mode Projected Route
MOP: RPL Mode of Operation
P-DAO: Projected DAO
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
SR-VIO: A Source-Routed Via Information Option, used in Non-Storing Mode P-DAO messages.
SMPR: Storing Mode Projected Route
TIO: RPL Transit Information Option
SF-VIO: A Via Information Option, used in Storing Mode P-DAO messages.
VIO: A Via Information Option; it can be a SF-VIO or an SR-VIO.

2.3. Other Terms

Projected Route: A RPL Projected 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.

Projected DAO: A DAO message used to install a Projected Route.

Track: A DODAG that provides a complex path from or to a Root that is the destination of the DODAG. The Root is the Track Ingress, and the forward direction for packets is down the DODAG, from the Track Ingress to one of the possibly multiple Track Egress Nodes.

TrackID: A RPL Local InstanceID with the 'D' bit set to 0. The TrackID is associated with the IPv6 Address of the Track Ingress that is used to signal the DODAG Root.

2.4. References

In this document, readers will encounter terms and concepts that are discussed in the "Routing Protocol for Low Power and Lossy Networks" [RPL] and "Terminology in Low power And Lossy Networks" [RFC7102].

3. Extending RFC 6550

3.1. 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). This specification extends the DAO message with the Projected DAO (P-DAO); a P-DAO message signals a Projected Route to one or more Targets using the new CMOs presented therein. This specification enables to combine one or more Projected Routes into a DODAG called a Track, that is traversed to reach the Targets.

The Track is assimilated with the DODAG formed for a Local RPL Instance. The local RPLInstanceID of the Track is called the TrackID, more in Section 7.2. A P-DAO message for a Track signals the TrackID in the RPLInstanceID field. The Track Ingress is signaled in the DODAGID field of the Projected DAO Base Object; that field is elided in the case of the main RPL Instance. The Track Ingress is the Root of the Track, as shown in Figure 1. .

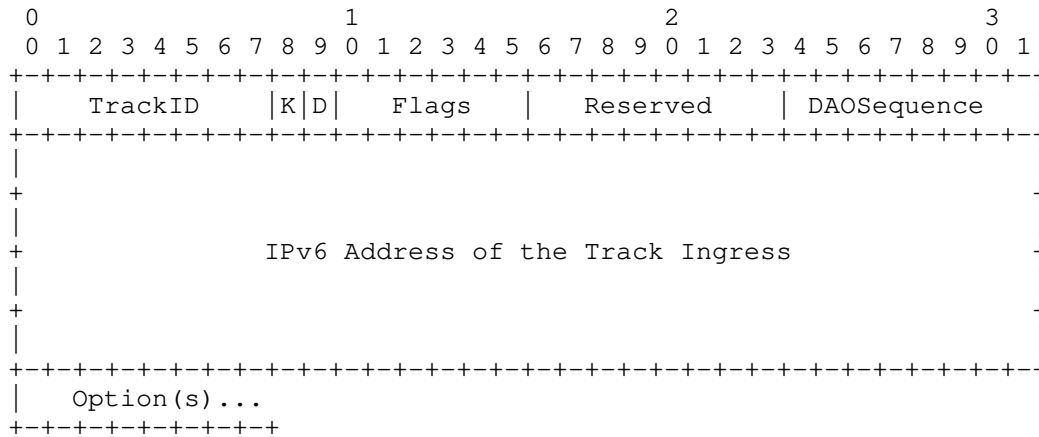


Figure 1: Projected DAO Format for a Track

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. Options may be factorized; multiple RTOs may be present to signal a collection of children that can be reached via the parent(s) indicated in the TIO(s) that follows the RTOs. This specification generalizes the case of a parent that can be used to reach a child with that of a whole Track through which both children and siblings of the Track Egress are reachable.

New CMOs called the Via Information Options (VIO) are introduced for use in P-DAO messages as a multihop alternative to the TIO. One VIO is the Stateful Via Information Option (SF-VIO); the SF-VIO installs Storing Mode Projected Route (SMPR) along a strict segment. The other is the Source-Routed SF-VIO (SR-VIO); the SR-VIO installs a Non-Storing Mode Projected Route (NMPR) at the Track Ingress, which uses that state to encapsulate a packet with a Routing Header (RH) to the Track Egress.

Like in a DAO message, the RTOs can be factorized in a P-DAO, but the Via Options cannot. A P-DAO contains one or more RTOs that indicate the destinations that can be reached via the Track, and exactly one Via Option that signals a sequence of nodes. In Non-Storing Mode, the Root sends the P-DAO to the Track Ingress where the source-routing state is stored. In Storing Mode, the P-DAO is sent to the Track Egress and forwarded along the Segment in the reverse direction, installing a Storing Mode state to the Track Egress at each hop. In both cases the Track Ingress is the owner of the Track, and it generates the P-DAO-ACK when the installation is successful.

3.2. Sibling Information Option

This specification adds another CMO called the Sibling Information Option (SIO) that is used by a RPL Aware Node (RAN) to advertise a selection of its candidate neighbors as siblings to the Root, more in Section 6.4. The sibling selection process is out of scope.

3.3. P-DAO Request

Two new RPL Control Messages are also introduced, to enable a RAN to request the establishment of a Track between self as the Track Ingress Node and a Track Egress. The RAN 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 6.1 for more. A positive PDR-ACK indicates that the Track was built and that the Roots commits to maintain the Track for the negotiated lifetime. 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. The Root may use an asynchronous PDR-ACK with a negative status to indicate that the Track was terminated before its time.

3.4. Extending 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 [USEofRPLInfo]). 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 extends [RPL] to create a new flag that signals that a packet is forwarded along a projected route.

Projected-Route 'P': 1-bit flag. It is set to 1 if this packet is sent over a projected route and set to 0 otherwise.

4. 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. [USEofRPLInfo] updated the Option Type from 0x63 to 0x23.

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

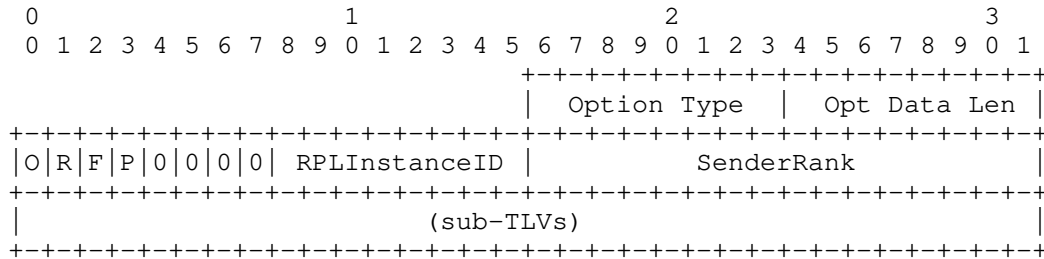


Figure 2: Extended RPL Option Format

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

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 3.4.

RPLInstanceID: See [RFC6553]. Indicates the TrackId if the 'P' flag is set.

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

5. Extending RFC 8138

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 Projected routes since the O,R,F flags are not used and the Rank is unknown and ignored.

This specification introduces a new 6LoRH, the P-RPI-6LoRH, with a type of 7. The P-RPI-6LoRH header is usually a a Critical 6LoWPAN Routing Header, but it can be elective as well if an SRH-6LoRH is present and controls the routing decision.

The P-RPI-6LoRH is designed to compress the RPI along RPL Projected Routes. It sformat is as follows:

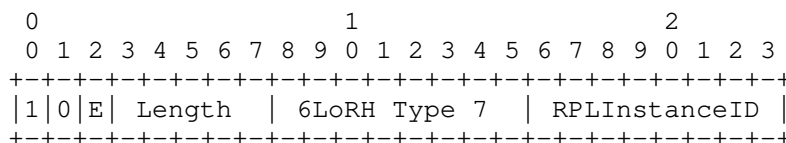


Figure 3: P-RPI-6LoRH Format

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

6. New RPL Control Messages and Options

6.1. 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. Exactly one RTO MUST be present in a PDR. The RTO signals the Track Egress, more in Section 7.1.

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

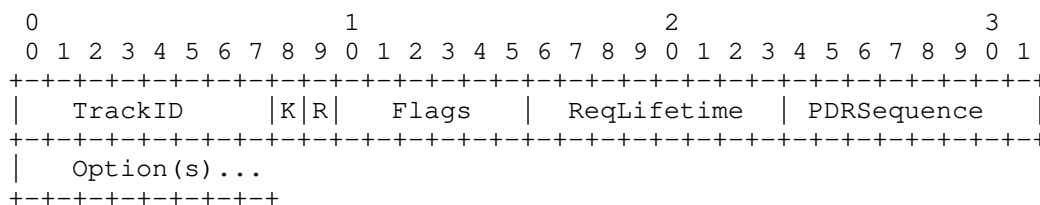


Figure 4: New P-DAO Request Format

TrackID: 8-bit field indicating the RPLInstanceID associated with the Track. It is set to zero upon the first request for a new Track and then to the TrackID once the Track was created, to either renew it or destroy it.

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 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.

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.

6.2. 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:

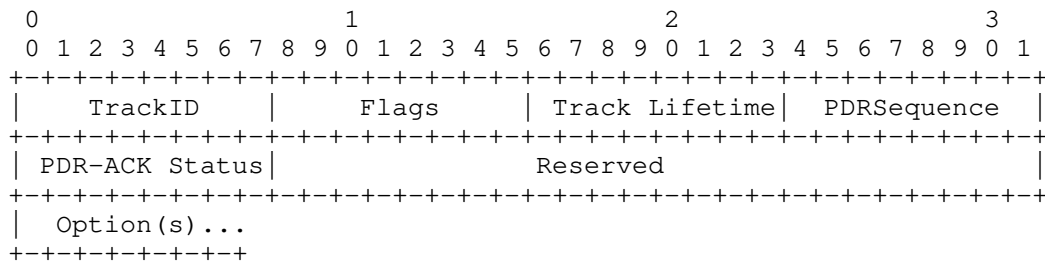


Figure 5: New PDR-ACK Control Message Format

TrackID: The RPLInstanceID of the Track that was created. The value of 0x00 is used to when no Track was created.

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 6:

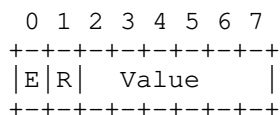


Figure 6: PDR-ACK status Format

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 7 and Table 8.

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

6.3. Via Information Options

An Via Option signals the ordered list of IPv6 Via Addresses that constitutes the hops of either a Serial Track or a Segment of a more Complex Track. An Via Option MUST contain at least one Via Address, and a Via Address MUST NOT be present more than once, otherwise the Via Option MUST be ignored. The format of the Via Options is as follows:

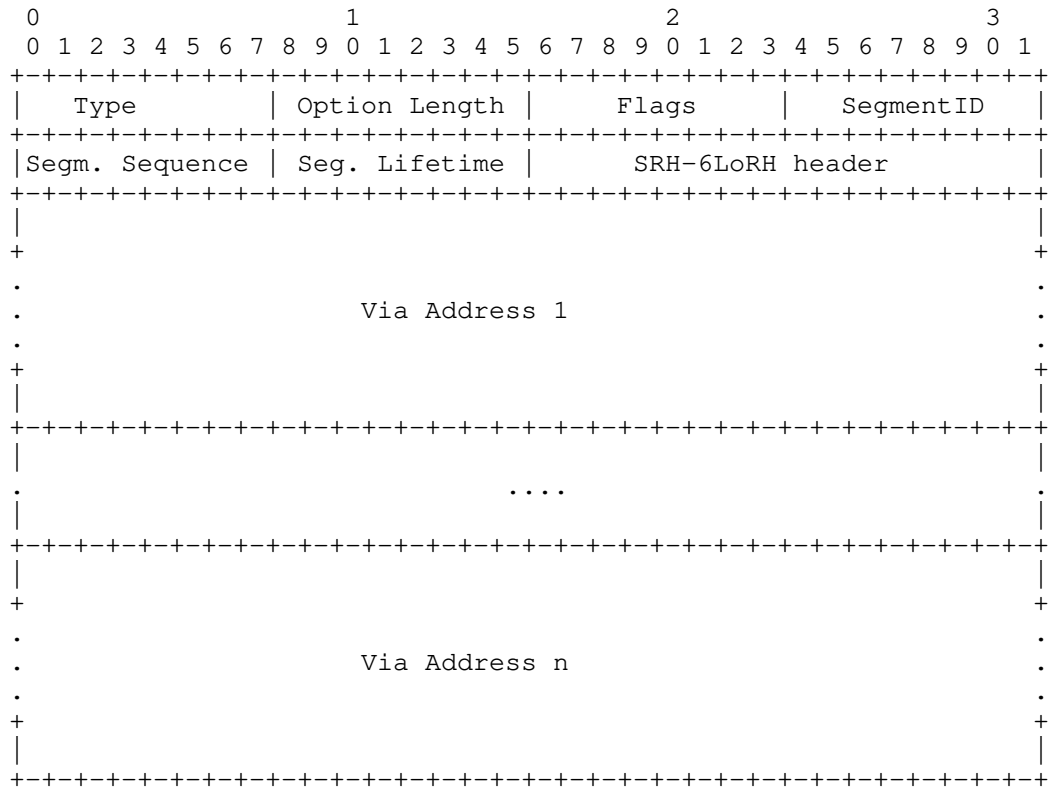


Figure 7: Via Information Option format (uncompressed form)

Option Type: 0x0B for SF-VIO, 0x0C for SR-VIO (to be confirmed by IANA)

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

SegmentID: 8-bit field that identifies a Segment within 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.

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 Via Option deprecates any state for the Segment indicated by the SegmentID within the indicated Track and sets up the new information.

An Via Option 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, SegmentID, 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.

A P-DAO message that contains a Via Information option with a Segment Lifetime of zero is referred as a No-Path P-DAO in this document.

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

Via Address: An IPv6 address along the Segment.

In a SF-VIO, the list is a strict path between direct neighbors, from the segment ingress to egress, both included. In an SR-VIO, the list starts at the first hop and ends at a Track Egress. The list in an SR-VIO may be loose, provided that each listed node has a path to the next listed node, e.g., via a segment or another Track.

In the case of a SF-VIO, or if [RFC8138] is not used in the data packets, then the Root MUST use only one SRH-6LoRH per Via Option, and the compression is the same for all the addresses, as shown in Figure 7.

In case of an SR-VIO, and if [RFC8138] is in use in the main DODAG, then the Root SHOULD optimize the size of the SR-VIO; more than one SRH-6LoRH may be present, e.g., if the compression level changes inside the Segment and different SRH-6LoRH Types are required. The content of the SR-VIO starting at the first SRH-6LoRH header is thus verbatim the one that the Track Ingress places in the packet encapsulation to reach the Track Ingress.

6.4. Sibling Information Option

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

The format of the SIO is as follows:

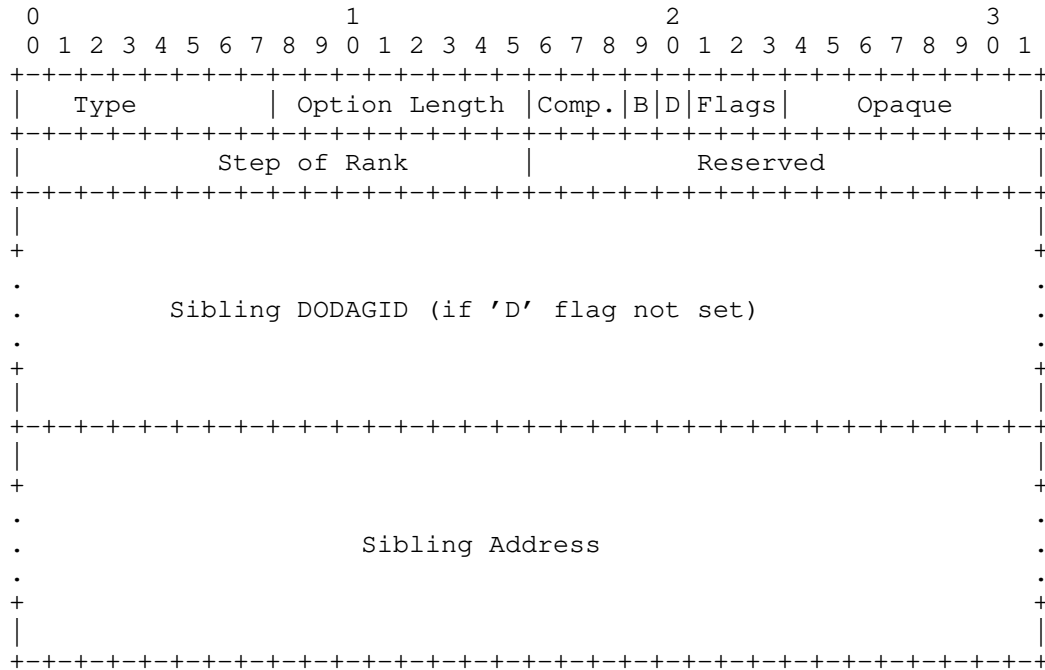


Figure 8: Sibling Information Option Format

Option Type: 0x0D (to be confirmed by IANA)

Option Length: In bytes, the size of the option.

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.

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.

D: 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.

Step of Rank: 16-bit unsigned integer. This is the Step of Rank [RPL] as computed by the Objective Function between this node and the sibling.

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 when the 'D' flag is not set.

Sibling Address: 2 to 16 bytes, the IPv6 Address of the sibling in a [RFC8138] compressed form as 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.

7. Projected DAO

This draft adds a capability to RPL whereby the Root of a main DODAG installs a Track as a collection of Projected Routes, using a Projected-DAO (P-DAO) message to maintain each individual route. 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 Via Information Option (VIO). A P-DAO message MUST contain exactly one VIO, which is either a SF-VIO or an SR-VIO, and

MUST follow one or more RTOs. There can be at most one such sequence of RTO(s) and an Via Option. A track is indentified by a tuple DODAGID, TrackID and each route within a Track is indexed by a SegmentID.

A P-DAO MUST be sent from the address of the Root that serves as DODAGID for the main DODAG. It MUST be sent to a GUA or a ULA of either the ingress or the egress of the Segment, more below. If the 'K' Flag is present in the P-DAO, and unless the P-DAO does not reach it, the ingress of the Segment is the node that acknowledges the message, using a DAO-ACK that MUST be sent back to the address that serves as DODAGID for the main DODAG.

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 Via Option as opposed to the Path Sequence from a TIO. Also, a Segment Lifetime of 0 in an Via Option indicates that the projected route associated to the Segment 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 7.5. A Non-Storing Mode P-DAO carries an SR-VIO with the loose list of Via Addresses that forms a source-routed Segment to the Track Egress. The recipient of the P-DAO is the Track Ingress; it MUST install a source-routed state to the Track Egress and reply to the Root directly using a DAO-ACK message if requested to.
- * The Storing Mode is discussed in Section 7.6. A Storing Mode P-DAO carries a SF-VIO with the strict list of Via Addresses from the ingress to the egress of the Segment in the data path order. The routers listed in the Via Addresses, except the egress, MUST install a routing state to the Target(s) via the next Via Address in the SF-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.

In case of a forwarding error along a Projected Route, an ICMP error is sent to the Root with a new Code "Error in Projected Route" (See Section 9.12). The Root can then modify or remove the Projected Route. The "Error in Projected Route" message has the same format as the "Destination Unreachable Message", as specified in RFC 4443 [RFC4443].

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.

The sender and exact operation depend on the Mode and is described in Section 7.5 and Section 7.6 respectively.

7.1. Requesting a Track

A Node is free to ask the Root for a new Track at any time. This is done with a PDR message, that indicates in the Requested Lifetime field the duration for which the Track should be established. Upon a PDR, the Root MAY install the necessary Segments, 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 is free to design the Track as it wishes, and to change the Segments overtime to serve the Track as needed, without notifying the requesting Node. 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, the requesting Node needs to 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.

7.2. Identifying a Track

RPL defines the concept of an Instance to signal an individual routing topology 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 within one routing topology.

This draft leverages the RPL Instance model as follows:

- * The Root MAY use P-DAO messages to add better routes in the main (Global) Instance in conformance with the routing objectives in that Instance. To achieve this, the Root MAY install an SMPR 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 Appendix A.1.

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

Once the Projected Route is installed, the intermediate nodes listed in the SF-VIO after first one (i.e. The ingress) can be elided from the RH in packets sent along the Segment signaled in the P-DAO. The resulting loose source routing header indicates (one of) the Target(s) as the next entry after the ingress.

- * The Root MAY also use P-DAO messages to install a specific (say, Traffic Engineered) path as a Serial or as a Complex Track, to a particular endpoint that is the Track Egress. In that case, the Root MUST install a Local RPL Instance (see section 5 of [RPL]).

In a that case, the TrackID MUST be unique for the Global Unique IPv6 Address (GUA) or Unique-Local Address (ULA) 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.

The Track Egress Address and the TrackID MUST be signaled in the P-DAO message as shown in Figure 1.

7.3. Installing a Track

A Storing Mode P-DAO contains an SF-VIO that signals the strict sequence of consecutive nodes to form a segment between a segment ingress and a segment egress (both included). It installs a route of a higher precedence along the segment towards the Targets indicated in the Target Options. The segment is included in a DODAG indicated by the P-DAO Base Object, that may be the one formed by the main RPL Instance, or a Track associated with a local RPL Instance. A Track Egress is signaled as a Target in the P-DAO, and as the last entry is an SF-VIO of a last segment towards that Egress.

A Non-Storing Mode P-DAO signals a strict or loose sequence of nodes between the Track Ingress (excluded) and a Track Egress (included). It installs a source-routed path of a higher precedence 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 encapsulation to add the SRH to an existing packet.

The next entry in the sequence must be either a neighbor of the previous entry, or reachable as a Target via another Projected Route, either Storing or Non-Storing. If it is reachable over a Storing Mode Projected Route, the next entry in the loose sequence is the Target of a previous segment and the ingress of a next segment; the segments are associated with the same Track, which avoids the need of an encapsulation. Conversely, if it is reachable over a Non-Storing Mode Projected Route, the next loose source routed hop of the inner Track is a Target of a previous Track and the ingress of a next Track, which requires a de- and a re-encapsulation.

A Serial Track is installed by a single Projected Routes that signals the sequence of consecutive nodes, either in Storing or Non-Storing Mode. It can be a loose Non-Storing Mode Projected 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 Projected Routes with the same DODAGID and Track ID. The Ingress of a Non-Storing Mode Projected Route must be the owner of the DODAGID. The Ingress of a Storing Mode Projected Route must be either the owner of the DODAGID, or the egress of a preceding Storing Mode Projected Route in the same Track. In the latter case, the Targets of the Projected Route must be Targets of the preceding Projected Route to ensure that they are visible from the track Ingress.

7.4. Forwarding Along a Track

This draft 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 7.4.

- * This draft conforms the principles of [USEofRPLInfo] with regards to packet forwarding and encapsulation along a Track.
 - In that case, the Track is the DODAG, the Track Ingress is the Root, and the Track Egress is a RAL, and neighbors of the Track Egress that can be reached via the Track are RULs. The encapsulation rules in [USEofRPLInfo] 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 in any fashion.

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. 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.

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.

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 main instance, e.g., by setting the "T" flag [TURN-ON_RFC8138] in the RPL configuration option.

7.5. Non-Storing Mode Projected Route

As illustrated in Figure 9, a P-DAO that carries an SR-VIO enables the Root to install a source-routed path towards a Track Egress in any particular router.

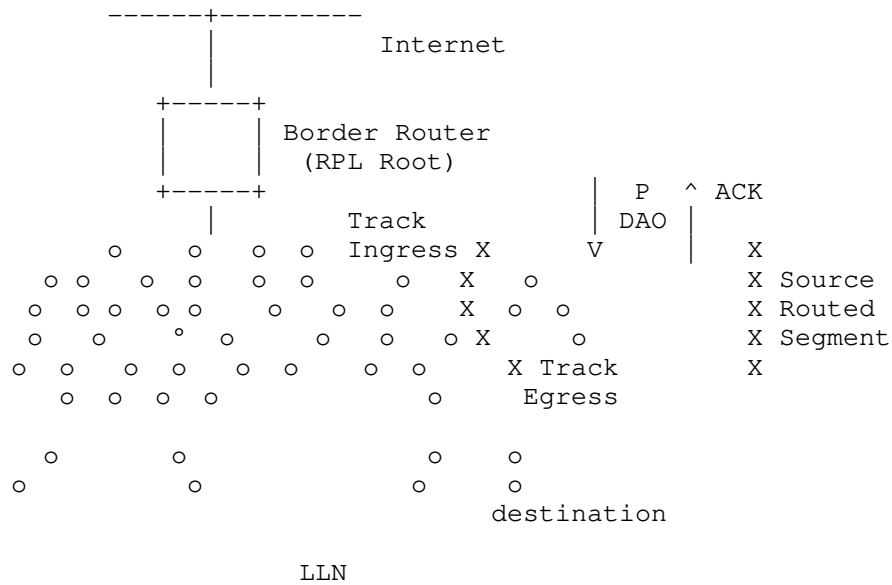


Figure 9: Projecting a Non-Storing Route

A route indicated by an SR-VIO 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.

When forwarding a packet to a destination for which the router determines that routing happens via the Track Egress, the router inserts the source routing header in the packet with the destination set to the Track Egress.

In order to signal the Segment, the router encapsulates the packet with an IP-in-IP header and a Routing Header as follows:

- * In the uncompressed form the source of the packet is this router, the destination is the first Via Address in the SR-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 SR-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 [TURN-ON_RFC8138].

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 another Track to the loose next hop for which this node is Ingress; in the latter case, another encapsulation takes place and the process possibly recurses; otherwise the packet is dropped.

In case of a 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 period of time and it SHOULD send an ICMP message with a Code "Error in Projected 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.

7.6. Storing Mode Projected Route

As illustrated in Figure 10, a P-DAO that carries a SF-VIO enables the Root to install a stateful route towards a collection of Targets along a Segment between a Track Ingress and a Track Egress.

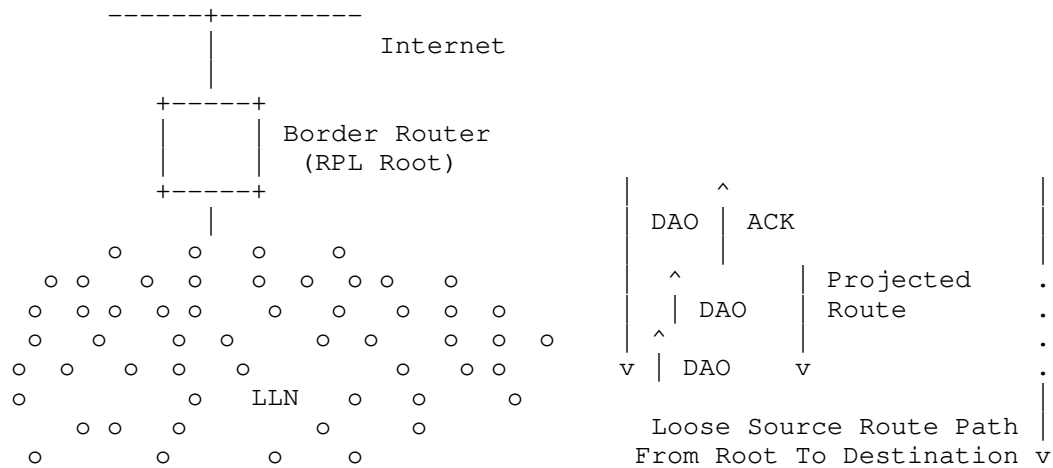


Figure 10: Projecting a route

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 SF-VIO with the direct sequence of Via Addresses. The SF-VIO follows one or more RTOs indicating the Targets to which the Track leads. The SF-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 SF-VIO. This is how the egress recognizes its role. In a similar fashion, the ingress node recognizes its role as it matches first Via Address in the SF-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. If one of the Targets is not known, 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 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 SF-VIO. The router MAY install additional routes towards the VIA Addresses that are the SF-VIO after the next one, if any, but in case of a conflict or a lack of resource, the route(s) to the Target(s) have precedence.

If a router cannot reach its predecessor in the SF-VIO, 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).

The process continues till the P-DAO is propagated to ingress router of the Segment, which answers with a DAO-ACK to the Root.

A Segment 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.

In case of a forwarding error along an SMPR, the node that fails to forward SHOULD send an ICMP error with a code "Error in Projected Route" to the Root. Failure to do so may result in packet loss and wasted resources along the Projected Route that is broken.

8. Security Considerations

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.

TODO: should probably consider how P-DAO messages could be abused by a) rogue nodes b) via replay of messages c) if use of P-DAO messages could in fact deal with any threats?

9. IANA Considerations

9.1. New Elective 6LoWPAN Routing Header Type

This document updates the IANA registry titled "Elective 6LoWPAN Routing Header Type" that was created for [RFC8138] and assigns the following value:

Value	Description	Reference
7	P-RPI-6LoRH	This document

Table 1: New Elective 6LoWPAN
Routing Header Type

9.2. New Critical 6LoWPAN Routing Header Type

This document updates the IANA registry titled "Critical 6LoWPAN Routing Header Type" that was created for [RFC8138] and assigns the following value:

Value	Description	Reference
7	P-RPI-6LoRH	This document

Table 2: New Critical 6LoWPAN
Routing Header Type

9.3. New 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 2, 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 6:

Bit number	Indication When Set	Reference
0	Down 'O'	[RFC6553]
1	Rank-Error (R)	[RFC6553]
2	Forwarding-Error (F)	[RFC6553]
3	Projected-Route (P)	This document

Table 3: Initial PDR Flags

9.4. New RPL Control Codes

This document extends the IANA Subregistry created by RFC 6550 for RPL Control Codes as indicated in Table 4:

Code	Description	Reference
0x09	Projected DAO Request (PDR)	This document
0x0A	PDR-ACK	This document

Table 4: New RPL Control Codes

9.5. New RPL Control Message Options

This document extends the IANA Subregistry created by RFC 6550 for RPL Control Message Options as indicated in Table 5:

Value	Meaning	Reference
0x0B	Stateful Via Information option (SF-VIO)	This document
0x0C	Source-Routed Via Information option (SR-VIO)	This document
0x0D	Sibling Information option	This document

Table 5: RPL Control Message Options

9.6. 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 6:

Bit number	Capability description	Reference
0	PDR-ACK request (K)	This document
1	Requested path should be redundant (R)	This document

Table 6: Initial PDR Flags

9.7. 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.

9.8. 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 7:

Value	Meaning	Reference
0	Unqualified acceptance	This document

Table 7: Acceptance values of the PDR-ACK Status

9.9. 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 8:

Value	Meaning	Reference
0	Unqualified rejection	This document

Table 8: Rejection values of the PDR-ACK Status

9.10. SubRegistry for the Via Information Options Flags

IANA is requested to create a Subregistry for the 5-bit Via Information Options (Via 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 Option) Flags.

9.11. 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 9:

Bit number	Capability description	Reference
0	Connectivity is bidirectional (B)	This document

Table 9: Initial SIO Flags

9.12. Error in Projected Route ICMPv6 Code

In some cases RPL will return an ICMPv6 error message when a message cannot be forwarded along a Projected Route. This ICMPv6 error message is "Error in Projected 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 Projected Route", with a suggested code value of 8, to be confirmed by IANA.

10. Acknowledgments

The authors wish to acknowledge JP Vasseur, Remy Liubing, James Pylakutty and Patrick Wetterwald for their contributions to the ideas developed here.

11. 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>>.
- [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>>.
- [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>>.

[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>>.

[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>>.

12. Informative References

[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>>.

[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>>.

[6TISCH-ARCHI]

Thubert, P., "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4", Work in Progress, Internet-Draft, draft-ietf-6tisch-architecture-29, 27 August 2020, <<https://tools.ietf.org/html/draft-ietf-6tisch-architecture-29>>.

[RAW-ARCHI]

Thubert, P., Papadopoulos, G., and R. Buddenberg, "Reliable and Available Wireless Architecture/Framework", Work in Progress, Internet-Draft, draft-pthubert-raw-

architecture-05, 15 November 2020,
<<https://tools.ietf.org/html/draft-pthubert-raw-architecture-05>>.

[TURN-ON_RFC8138]

Thubert, P. and L. Zhao, "A RPL DODAG Configuration Option for the 6LoWPAN Routing Header", Work in Progress, Internet-Draft, draft-ietf-roll-turnon-rfc8138-17, 30 September 2020, <<https://tools.ietf.org/html/draft-ietf-roll-turnon-rfc8138-17>>.

[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>>.

[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>>.

[USEofRPLInfo]

Robles, 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", Work in Progress, Internet-Draft, draft-ietf-roll-useofrplinfo-42, 12 November 2020, <<https://tools.ietf.org/html/draft-ietf-roll-useofrplinfo-42>>.

[PCE] IETF, "Path Computation Element", <<https://datatracker.ietf.org/doc/charter-ietf-pce/>>.

Appendix A. Applications

A.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 11. 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.

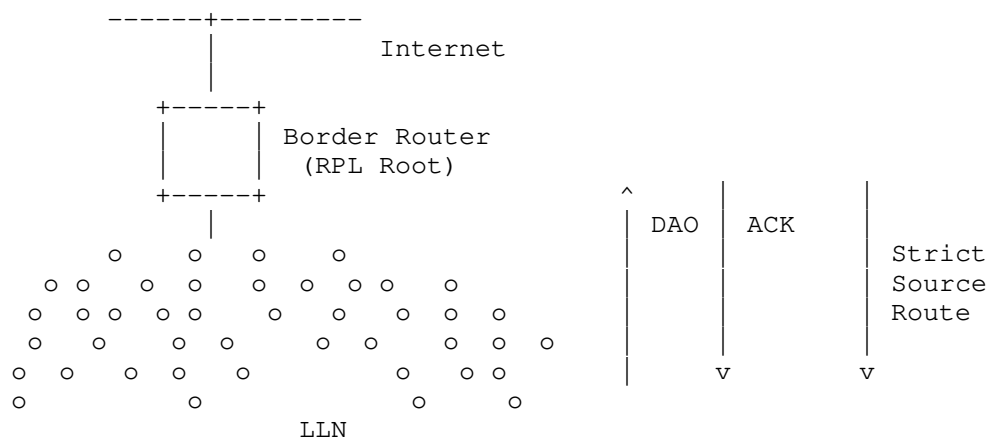


Figure 11: RPL Non-Storing Mode of operation

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 a 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.

A.2. Transversal Routes

RPL is optimized for Point-to-Multipoint (P2MP) and Multipoint-to-Point (MP2P), whereby routes are always installed along the RPL DODAG respectively from and towards the DODAG Root. Transversal Peer to Peer (P2P) routes in a RPL network will generally suffer from some elongated (stretched) path versus the best possible path, since routing between 2 nodes always happens via a common parent, as illustrated in Figure 12:

- * 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.
- * 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.

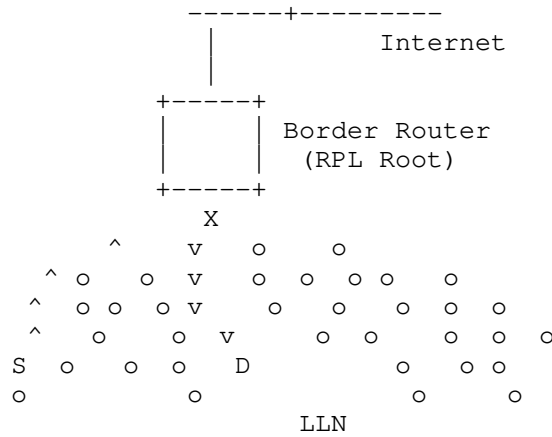


Figure 12: Routing Stretch between S and D via common parent X

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, 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.

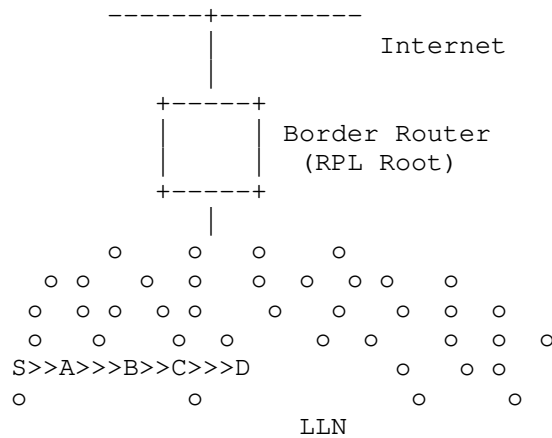


Figure 13: Projected Transversal Route

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.

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

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

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

Matthew Gillmore
Itron, Inc
Building D
2111 N Molter Road
Liberty Lake, 99019
United States

Phone: +1.800.635.5461
Email: matthew.gillmore@itron.com

ROLL
Internet-Draft
Intended status: Standards Track
Expires: April 20, 2018

R. Jadhav, Ed.
R. Sahoo
Z. Cao
Huawei Tech
October 17, 2017

No-Path DAO modifications
draft-ietf-roll-efficient-npdo-01

Abstract

This document describes the problems associated with the use of No-Path DAO messaging in RPL and a 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 April 20, 2018.

Copyright Notice

Copyright (c) 2017 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
1.2.	Current No-Path DAO messaging	3
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 NP-DAO due to link break to the previous parent	5
2.2.	Invalidate routes to dependent nodes of the switching node	5
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 the 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 NP-DAO operation in progress	6
4.	Proposed changes to RPL signaling	7
4.1.	Change in NPDAO semantics	7
4.2.	DAO message format changes	7
4.3.	Destination Cleanup Object (DCO)	8
4.3.1.	DCO Options	10
4.3.2.	Path Sequence number in the DCO	10
4.3.3.	Destination Cleanup Option Acknowledgement (DCO-ACK)	10
4.4.	Example messaging	11
4.5.	Other considerations	12
4.5.1.	Dependent Nodes invalidation	12
5.	Acknowledgements	13
6.	IANA Considerations	13
7.	Security Considerations	13
8.	References	13
8.1.	Normative References	13
8.2.	Informative References	14
Appendix A.	Additional Stuff	14
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: A new RPL control message type defined by this specification and stands for Destination Cleanup Object.

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:

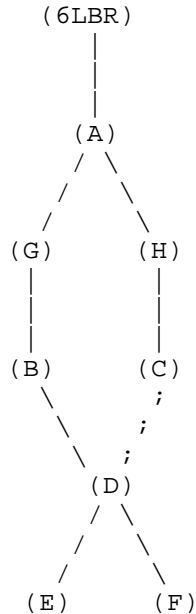


Figure 1: Sample topology

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 NP-DAO 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 long.

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.

An implementation technique to avoid this problem is to further delay the route invalidation by a fixed time interval after receiving an NPDAO, considering the time taken for the new path to be established. Coming up with such a time interval is tricky since the new route may also not be available and it may subsequently require more parent switches to establish a new path.

3. Requirements for the No-Path DAO Optimization

3.1. Req#1: Tolerant to the link failures to the previous parents

When the switching node send the NP-DAO 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 NP-DAO 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 NP-DAO 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 NP-DAO operation in progress

While sending the NP-DAO and DAO messages, it is possible that the NP-DAO 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 NP-DAO is synchronized with the DAO to avoid the risk of route downtime.

4. Proposed changes to RPL signaling

4.1. Change in NPDAO 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 proposes to add 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).

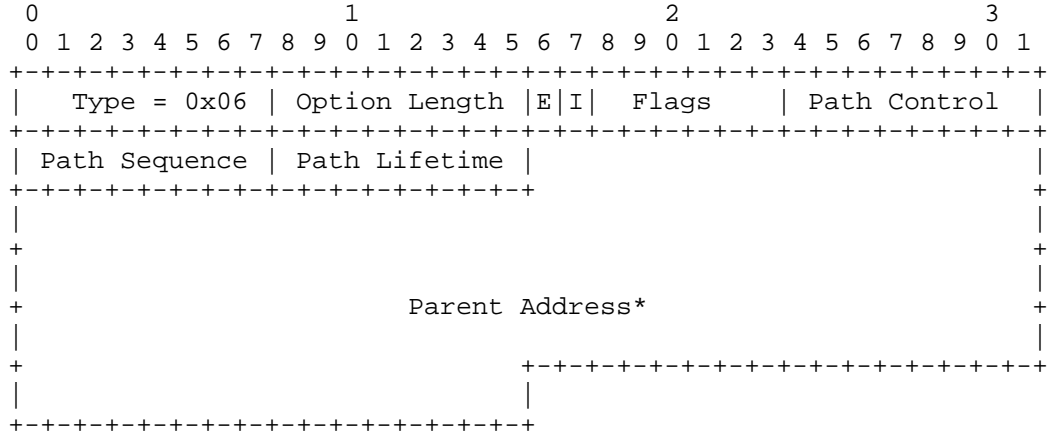


Figure 2: Updated Transit Information Option (New I flag added)

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.

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.

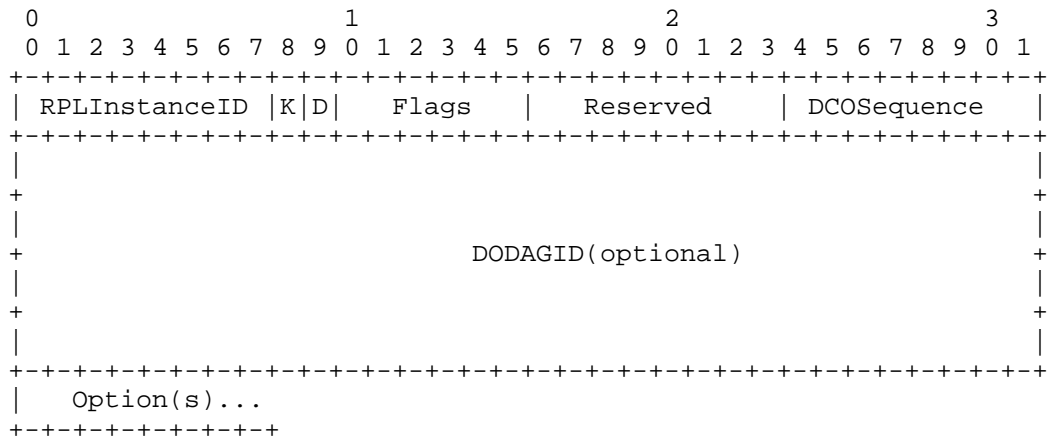


Figure 3: DCO base object

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 DAO Base are reserved. They MUST be set to zero on transmission and MUST be ignored on reception.

4.3.1. 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.

4.3.2. 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 and should use the same Path Sequence number present in the regular DAO message when the DCO is generated in response to DAO message.

4.3.3. Destination Cleanup Option Acknowledgement (DCO-ACK)

The DCO-ACK message is sent as a unicast packet by a DCO recipient in response to a unicast DCO message.

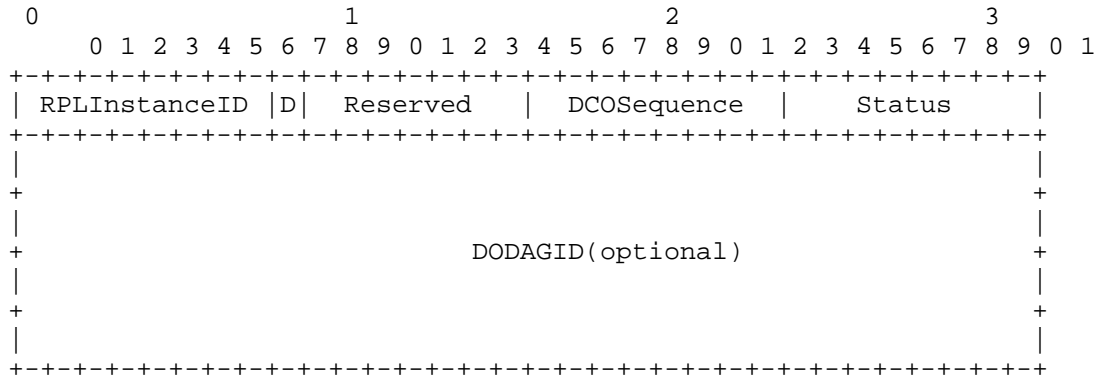


Figure 4: DCO-ACK base object

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 DAO Base are reserved. They MUST be set to zero on transmission and MUST be ignored on reception.

4.4. Example 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($\text{tgt}=\text{D}, \text{pathseq}=\text{x}+1, \text{I_flag}=1$) 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($\text{tgt}=\text{D}, \text{pathseq}=\text{pathseq}(\text{DAO})$) 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($\text{tgt}=\text{D}, \text{pathseq}=\text{x}+1, \text{I_flag}=\text{x}$) (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.

4.5. Other considerations

4.5.1. Dependent Nodes invalidation

Current RPL [RFC6550] does not provide a mechanism for route invalidation for dependent nodes.

This section describes approaches for invalidating routes of dependent nodes if the implementation chooses to solve this problem. The common ancestor node realizes that the paths for dependent nodes have changed (based on next hop change) when it receives a regular DAO on behalf of the dependent nodes. Thus dependent nodes route invalidation can be handled in the same way as the switching node. Note that there is no way that dependent nodes can set the I_flag in the DAO message selectively since they are unaware that their parent/grand parent node is switching paths. There are two ways to handle dependent node route invalidation:

1. One way to resolve is that the common ancestor does not depend upon the I_flag to generate the reverse NPDAO. The only factor it makes the decision will be based on next_hop change for an existing target to generate the NPDAO. Thus when the switching nodes and all the below dependent nodes advertise a regular DAO, the common ancestor node will detect a change in next hop and generate NPDAO for the same target as in the regular DAO.
2. Another way is that the nodes always set the I_flag whenever they send regular DAO. Thus common ancestor will first check whether I_flag is set and then check whether the next_hop has changed and subsequently trigger DCO if required.

This document recommends the approach in point 2. The advantage with I_flag is that the generation of downstream NPDAO is still controlled by the target node and thus is still in control of its own routing state.

5. Acknowledgements

We would like to thank Cenk Gundogan, Simon Duquennoy and Pascal Thubert 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
0x85	Destination Cleanup Object	This document
0x86	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

The secure versions of DCO and DCO-ACK also have to be considered in the future. The security considerations applicable to DAO, DAO-ACK messaging in RPL is also applicable here.

8. References

8.1. Normative 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-12 (work in progress), August 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>>.

8.2. Informative References

- [CONTIKI] Thingsquare, "Contiki: The Open Source OS for IoT", 2012, <<http://www.contiki-os.org>>.
- [RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <<https://www.rfc-editor.org/info/rfc3552>>.

Appendix A. Additional Stuff

This becomes an Appendix.

Authors' Addresses

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

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

Rabi Narayan Sahoo
Huawei Tech
Kundalahalli Village, Whitefield,
Bangalore, Karnataka 560037
India

Phone: +91-080-49160700
Email: rabinarayans@huawei.com

Zhen Cao
Huawei Tech
W Chang'an Ave
Beijing 560037
China

Email: 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:

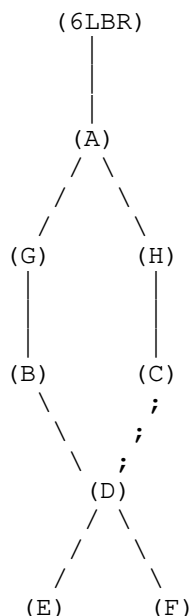


Figure 1: Sample topology

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].

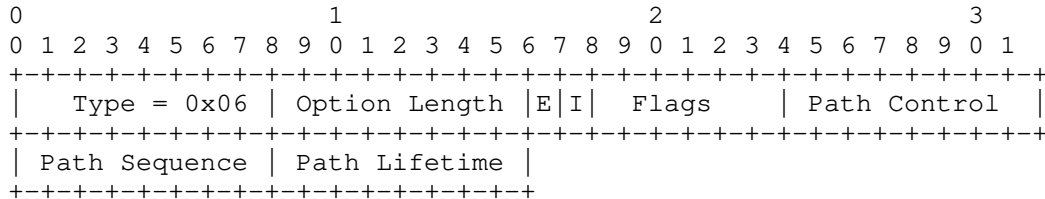


Figure 2: Updated Transit Information Option (New I flag added)

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.

The common ancestor node SHOULD generate a DCO message in response to this 'I' flag when it sees that the routing adjacencies have changed for the target. The 'I' flag is intended to give the target node control over its own route invalidation, serving as a signal to request DCO generation.

4.3. Destination Cleanup Object (DCO)

A new ICMPv6 RPL control message code is defined by this specification and is referred to 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.

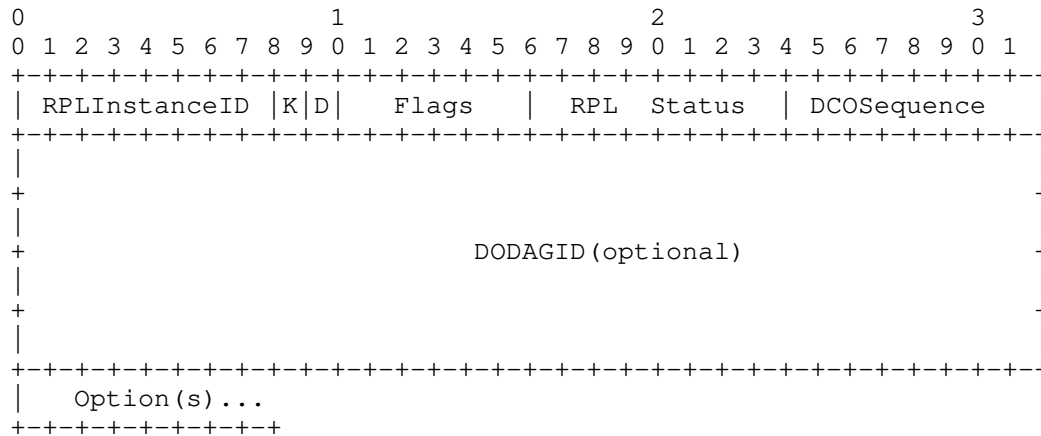


Figure 3: DCO base object

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 of DCO message is expected to send a DCO-ACK back. If the DCO-ACK is not received even after setting the 'K' flag, an implementation may retry the DCO at a later time. The number of retries are implementation and deployment dependent and are expected to be kept similar with those used in DAO retries in [RFC6550]. Section 4.6.3 specifies the considerations for DCO retry. A node receiving a DCO message without the 'K' flag set MAY respond with a DCO-ACK, especially to report an error condition. An example error condition could be that the node sending the DCO-ACK does not find the routing entry for the indicated target. When the sender does not set the 'K' flag it is an indication that the sender does not expect a response, and the sender SHOULD NOT retry the DCO.

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.

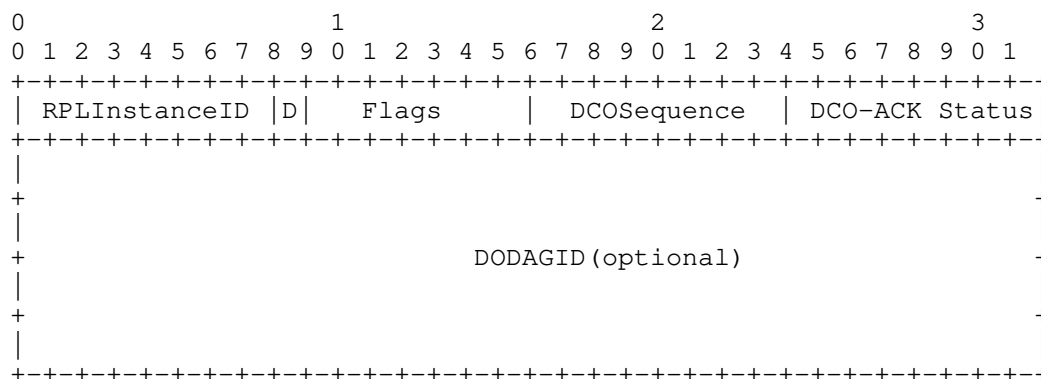


Figure 4: DCO-ACK base object

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

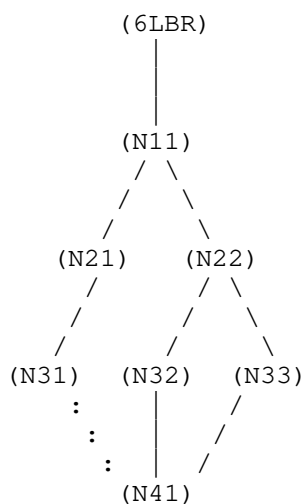


Figure 5: Sample topology 2

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: June 9, 2018

C. Ji, Ed.
R. Koutsiamanis
G. Papadopoulos
IMT Atlantique
D. Dujovne
Universidad Diego Portales
N. Montavont
IMT Atlantique
December 6, 2017

Traffic-aware Objective Function
draft-ji-roll-traffic-aware-objective-function-00

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 June 9, 2018.

Copyright Notice

Copyright (c) 2017 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 7
 8. IANA Considerations 7
 9. Informative references 8
 Authors' Addresses 8

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.gasem-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).

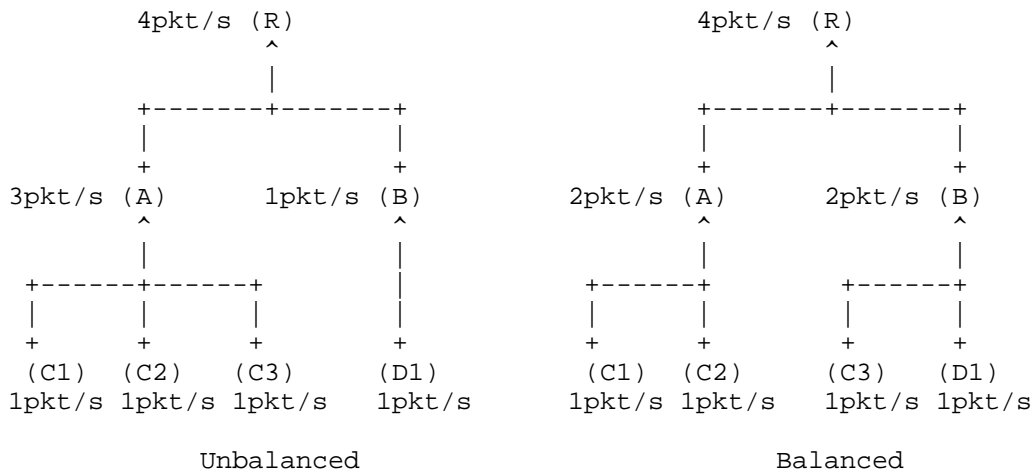


Figure 1: Packet Transmission Rates of nodes with the same requirements

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.

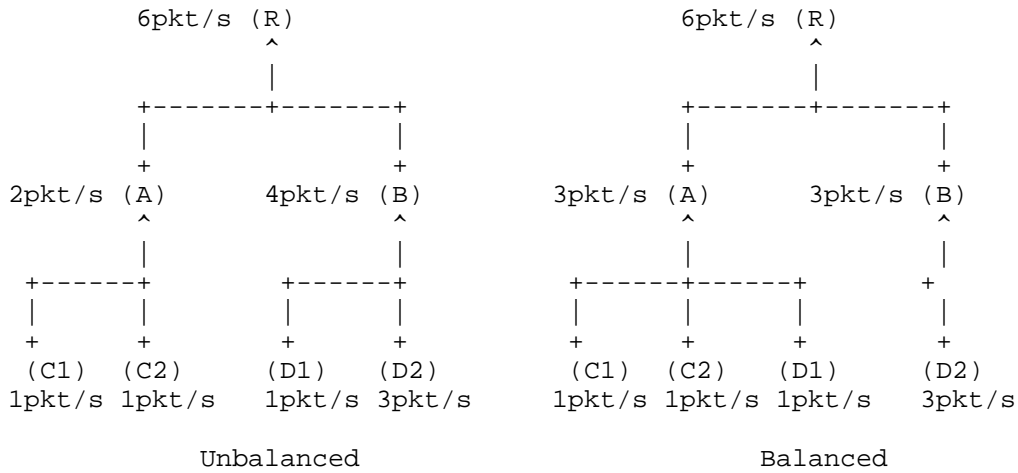


Figure 2: Packet Transmission Rates of nodes with different requirements

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

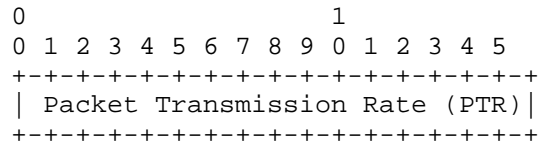


Figure 3: DAG metric container type format.

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.

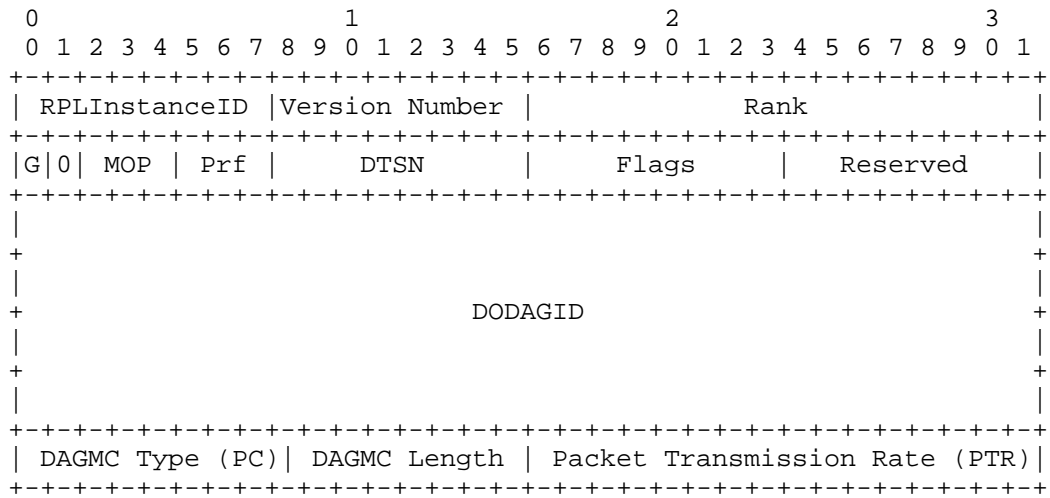


Figure 4: Example DIO Message with a DAG Metric Container option

An example DIO Message containing the proposed DAG Metric Container type is shown in Figure 4. The explicit definition of the fields is:

- DAGMC type: The type of the proposed DAGMC extension. To be assigned by IANA.
- DAGMC Length: The total length of the proposed DAGMC extension in bytes. MUST be 2.
- Packet Transmission Rate (PTR): The DAG Metric Container data, containing 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 for the metric type "PTR" in the Routing Metric/Constraint Type 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

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

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

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: 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

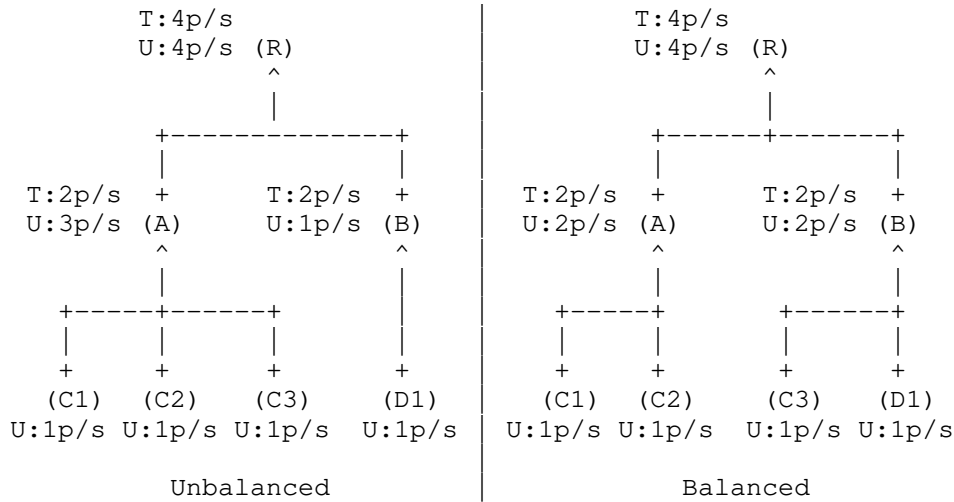


Figure 1: Use of Remaining Throughput with nodes with the same requirements

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 $2-1=1\text{p/s}$. 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.

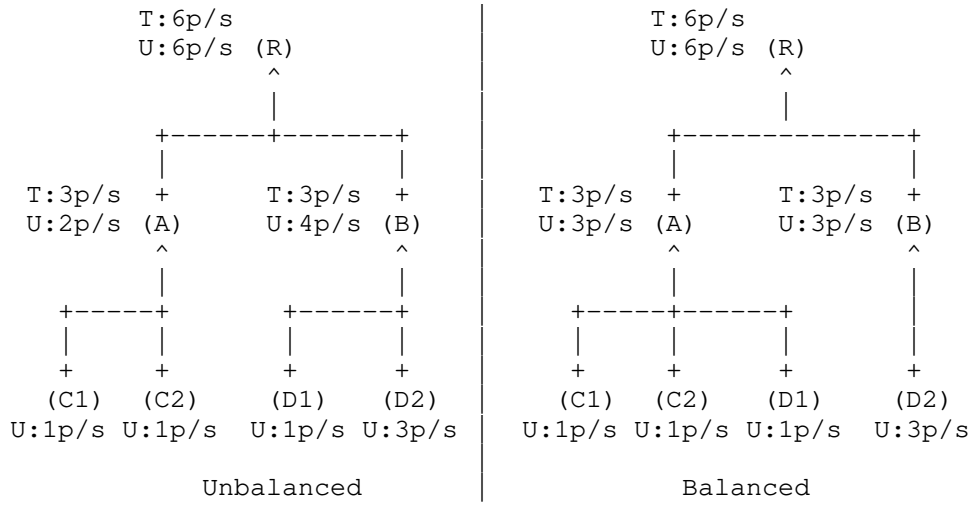


Figure 2: Use of Remaining Throughput with nodes with different requirements

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.

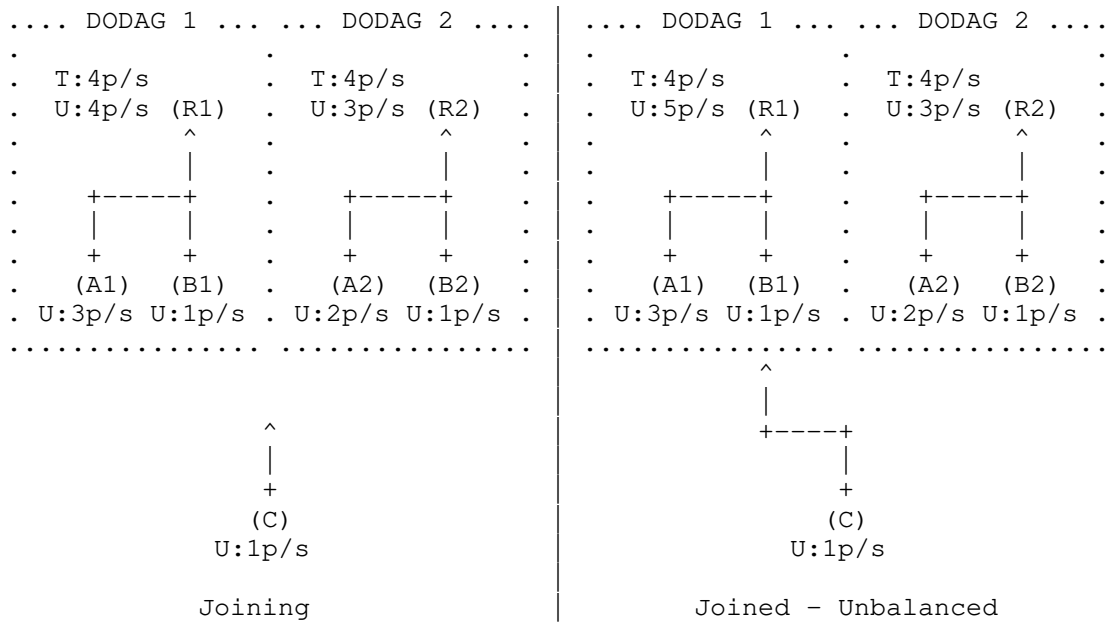


Figure 3: DODAG selection example leading to unbalanced traffic with RT metric

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.

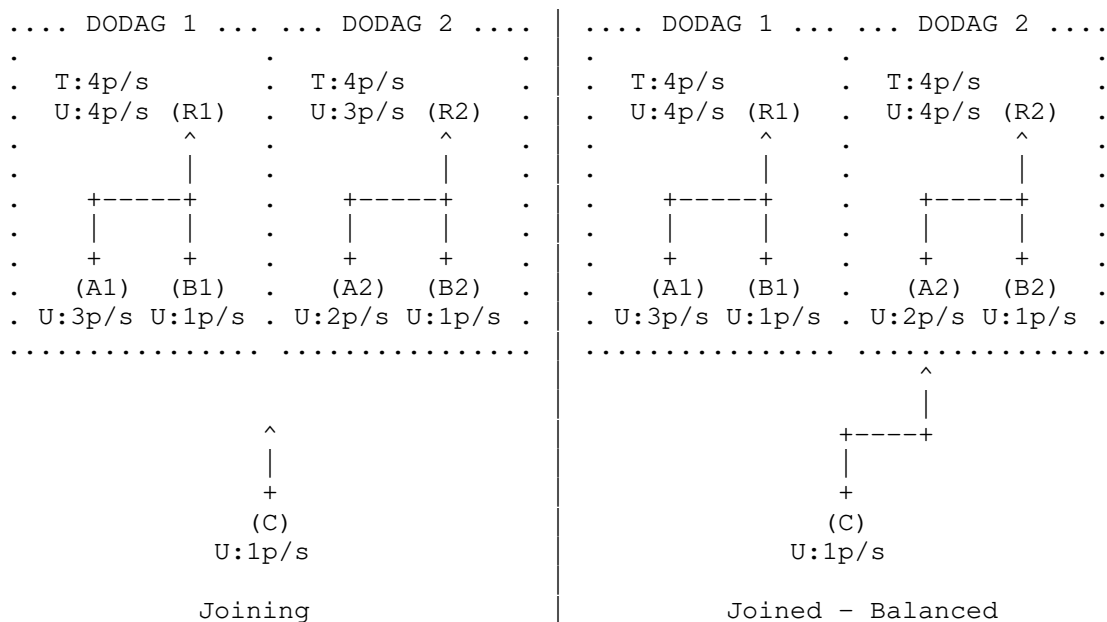


Figure 4: DODAG selection example leading to balanced traffic with RT metric

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

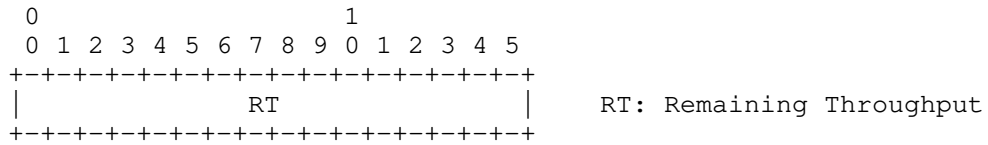


Figure 5: DAG metric container type format.

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.

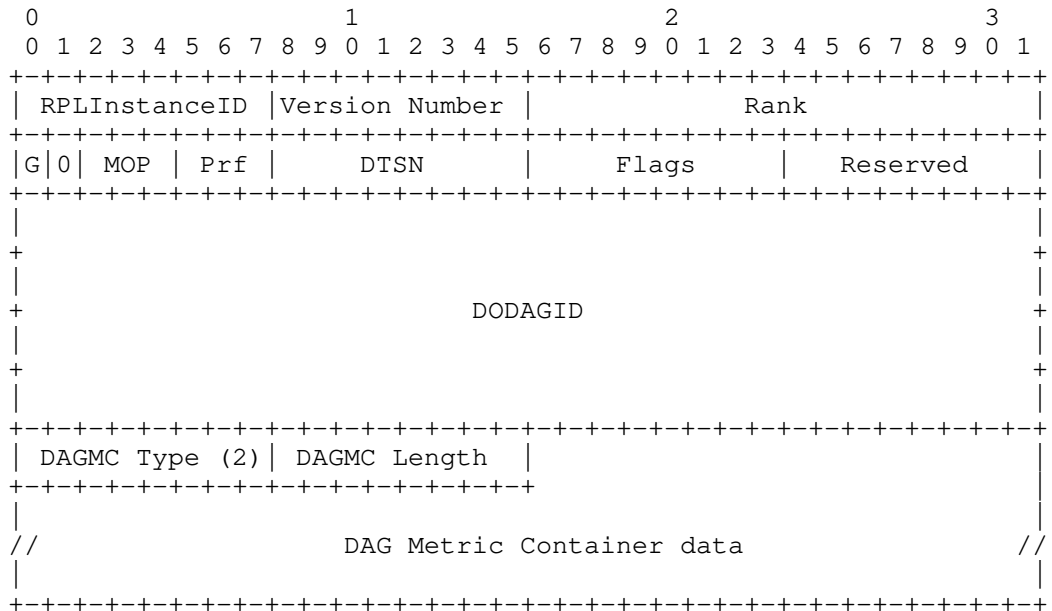


Figure 6: Example DIO Message with a DAG Metric Container option

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.

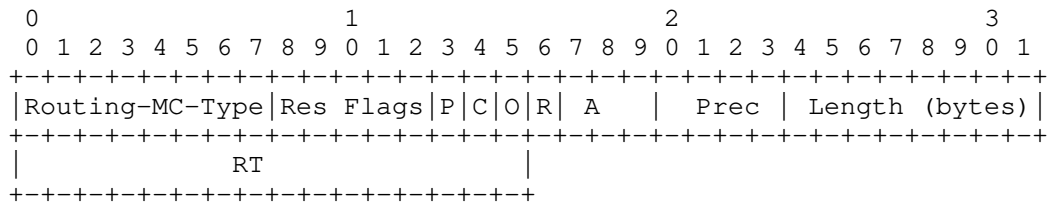


Figure 7: DAG Metric Container (MC) data with Remaining Throughput (RT) object body

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

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

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

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: July 21, 2018

R. Koutsiamanis, Ed.
G. Papadopoulos
N. Montavont
IMT Atlantique
P. Thubert
Cisco
January 17, 2018

RPL DAG Metric Container (MC) Node State and Attribute (NSA) object type
extension
draft-koutsiamanis-roll-nsa-extension-01

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 July 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
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	9
9. References	9
9.1. Informative references	9
9.2. Other Informative References	10
Authors' Addresses	10

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 node 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 node 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.

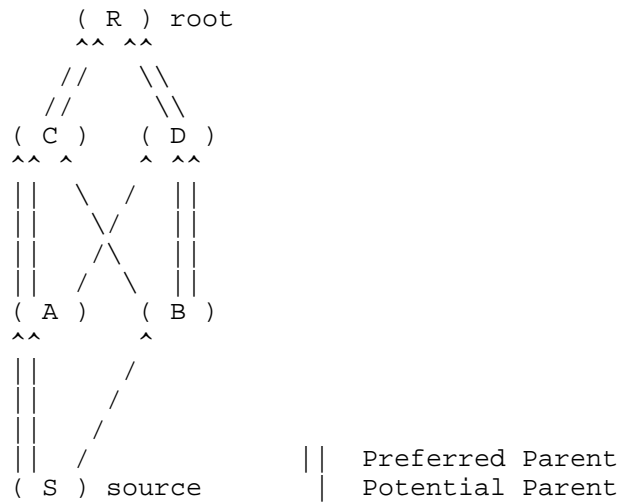


Figure 1: Packet Replication: S transmits the same data packet twice: to its DP (A) and to its AP (B).

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.

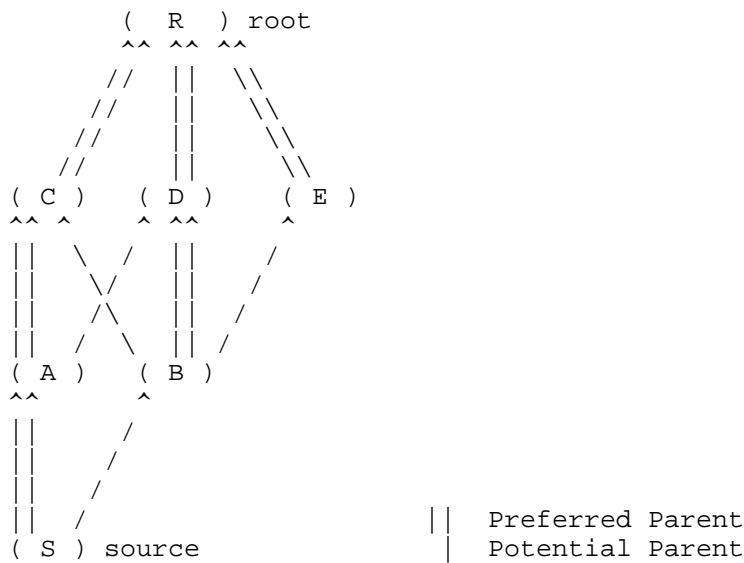


Figure 2: Example Parent Selection mechanism

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 node 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 node 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 node set.

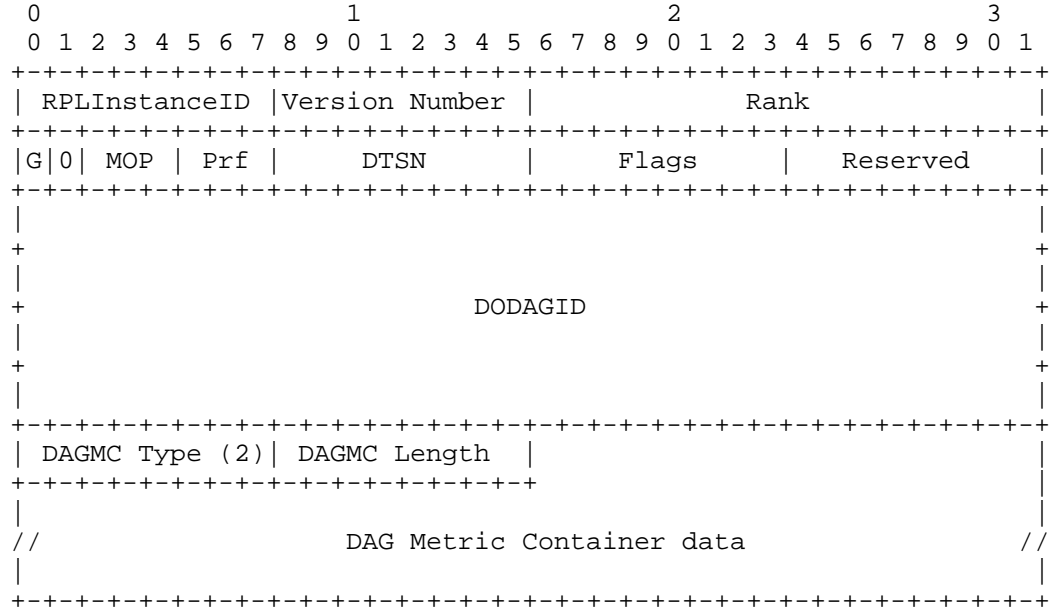


Figure 3: Example DIO Message with a DAG Metric Container option

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.

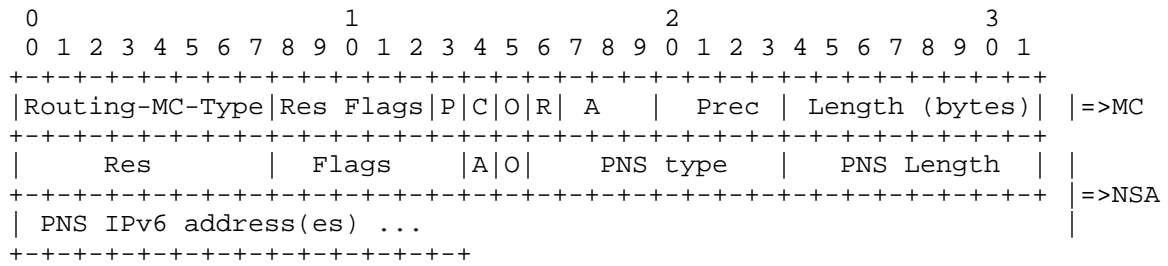


Figure 4: DAG Metric Container (MC) data with Node State and Attribute (NSA) object body and a TLV

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 Node Set and is abbreviated as PNS in Figure 4.

PNS type: The type of the Parent Node Set TLV. The value is TBD1.

PNS Length: The total length of the TLV value field (PNS IPv6 address(es)) in bytes.

PNS 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 PNS 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 Node 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 PNS MUST NOT be aggregated.

6.1.1.1. DAG Metric Container fields

Given the intended usage, when using the PNS, the NSA object it is contained in MUST be used as a constraint in the DAG Metric Container. More specifically, using the PNS places the following requirements on the DAG Metric Container header fields:

- o 'P' flag: MUST be cleared, since PNS is used only with constraints.
- o 'C' flag: MUST be set, since PNS is used only with constraints.
- o 'O' flag: Used as per [RFC6550], to indicated optionality.
- o 'R' flag: MUST be cleared, since PNS is used only with constraints.
- o 'A' Field: MUST be set to 0 and ignored, since PNS 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 PNS places no additional restrictions on the NSA flags ('A' and 'O'), which can be used as normally defined in [RFC6550].

6.2. Compression

The PNS IPv6 address(es) field in the Parent Node Set TLV MAY be compressed using any compression method available to conserve space.

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-13 (work in progress), November 2017.
- [I-D.ietf-detnet-architecture]
Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-04 (work in progress), October 2017.
- [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>>.

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.

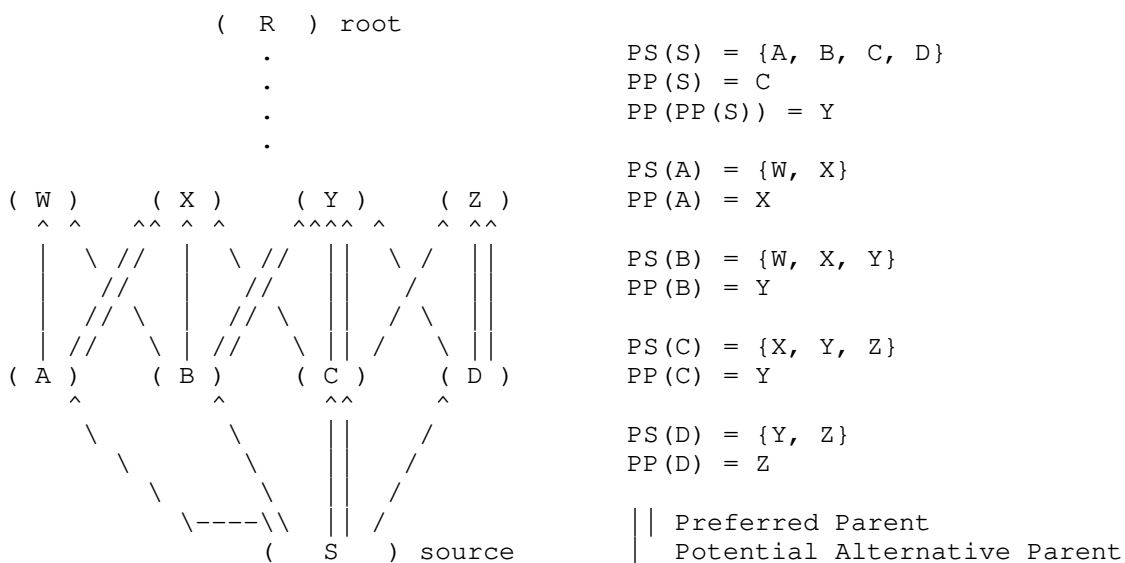


Figure 1: Example Common Ancestor Strict Alternative Parent Selection method

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 (PP) 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.

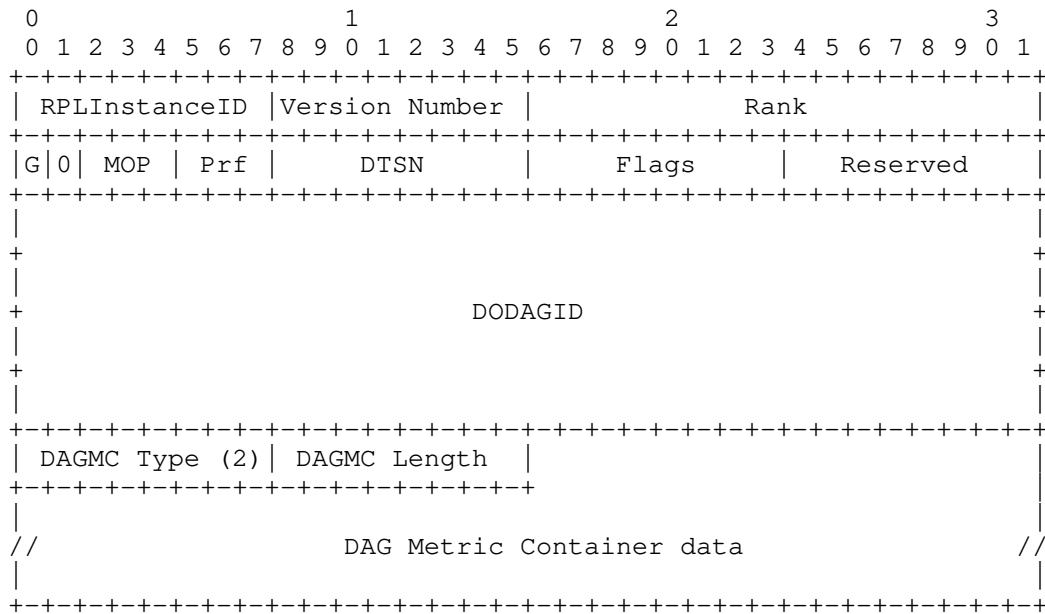


Figure 2: Example DIO Message with a DAG Metric Container option

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.

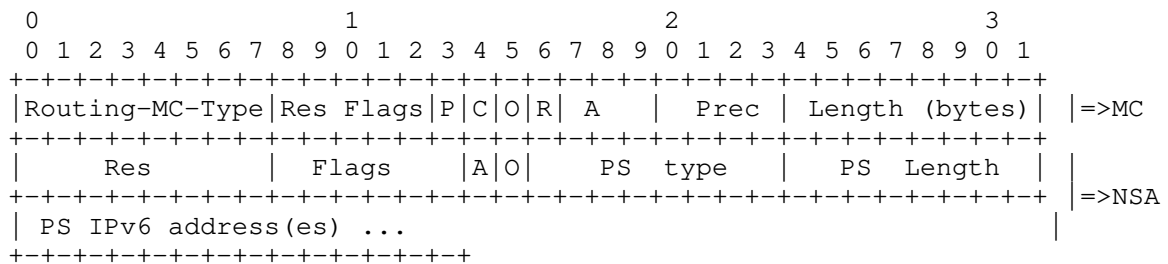


Figure 3: DAG Metric Container (MC) data with Node State and Attribute (NSA) object body and a TLV

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: August 8, 2018

R. Jadhav, Ed.
R. Sahoo
Y. Wu
Huawei
February 4, 2018

RPL Observations
draft-rahul-roll-rpl-observations-00

Abstract

This document describes RPL protocol design issues, various observations and possible consequences of the design and implementation choices. Also mentioned are implementation notes for the developers to be used in specific contexts.

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 August 8, 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.	Managing persistent variables across node reboots	3
2.1.	Persistent storage and RPL state information	3
2.2.	Lollipop Counters	3
2.3.	RPL State variables	4
2.3.1.	DODAG Version	5
2.3.2.	DTSN field in DIO	5
2.3.3.	PathSequence	5
2.4.	State variables update frequency	5
2.5.	Recommendations	6
2.6.	Implementation Notes	6
3.	DTSN increment in storing MOP	6
4.	DAO retransmission and use of DAO-ACK	7
5.	Handling resource unavailability	8
6.	Traffic Types observations	9
7.	RPL under-specification	9
8.	Acknowledgements	10
9.	IANA Considerations	10
10.	Security Considerations	10
11.	References	10
11.1.	Normative References	10
11.2.	Informative References	11
Appendix A.	Additional Stuff	11
Authors' Addresses	11

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. Managing persistent variables across node reboots

2.1. 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.

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.

2.2. 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.

2.3. 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.

2.3.1. 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.

2.3.2. 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).

2.3.3. 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.

2.4. 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

2.5. Recommendations

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.

2.6. 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. 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.

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 mechansim on how the DTSN update should happen in case of storing mode.

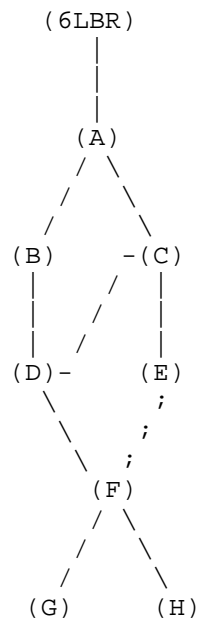


Figure 1: Sample topology

Consider example topology shown in figure 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.

4. DAO retransmission and use of DAO-ACK

[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) A parent responds with a DAO-ACK immediately after receiving the DAO.
- (2) A node waits for the upstream parent to send DAO-ACK to respond with a DAO-ACK downstream. This may not be feasible to use on constrained devices because it requires additional state information and timers to be handled on behalf of multiple downstream nodes whose DAO is in transit.

Following scenarios do not have clear handling in the specs:

- (1) What happens if the DAO-ACK for the target is lost at the ancestor node link?
- (2) What happens if the DAO-ACK with Status!=0 is responded by ancestor node?
- (3) Is there any way for the target node to know that the DAO it sent has reached the 6LBR successfully?

Note that any of these inefficiencies are not present in case of NSMOP in which the DAO is addressed directly to the 6LBR.

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.

Following needs to be handled by the specs:

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. Traffic Types observations

RPL is more suited towards MP2P (multi-point to point) traffic, the central point here usually is a grounded root/6LBR node. [RFC6997] allows establishing P2P paths within the DODAG. There are situations where a MP2P network needs to be established within the DODAG. For e.g. there could be multiple switches connecting the same light bulb. Currently to achieve this, every switch needs to establish a P2P path to the bulb. In cases where the cardinality of nodes connecting to the same node is high the cost of establishing P2P paths could be very high. RPL allows 'floating' DODAG to be created but the specification defines it to be used under other circumstances. To quote [RFC6550],

"A grounded DODAG offers connectivity to hosts that are required for satisfying the application-defined goal. ___A floating DODAG is not expected to satisfy the goal; in most cases, it only provides routes to nodes within the DODAG. Floating DODAGs may be used, for example, to preserve interconnectivity during repair.___"

Thus it is not clear whether floating DODAGs can be put to use for establishing MP2P paths within the DODAG.

7. 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.

8. Acknowledgements

9. IANA Considerations

This memo includes no request to IANA.

10. Security Considerations

This is an information draft and does not add any changes to the existing specifications.

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>>.
- [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>>.

11.2. 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-10 (work in progress), January 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
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.

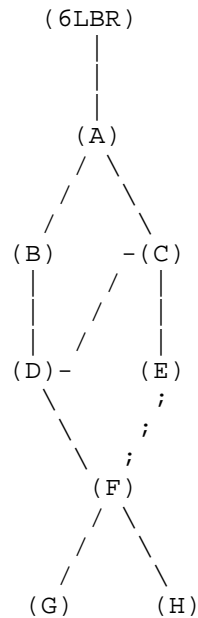


Figure 1: Sample topology

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.

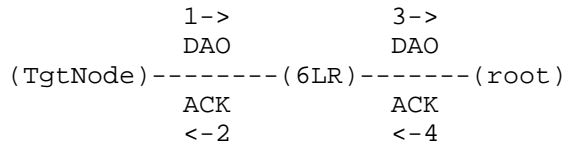


Figure 2: Hop-by-hop DAO-ACK

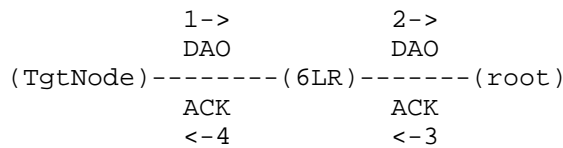


Figure 3: End-to-End DAO-ACK

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 bidirectional 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 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?

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 not 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

6lo Working Group
Internet-Draft
Intended status: Informational
Expires: August 6, 2018

M. Richardson
Sandelman Software Works
February 02, 2018

Enabling secure network enrollment in RPL networks
draft-richardson-6tisch-roll-enrollment-priority-00

Abstract

[I-D.richardson-6tisch-join-enhanced-beacon] defines a method by which a potential [I-D.ietf-6tisch-minimal-security] can announce itself as available for new Pledges to Join a network. The announcement includes a priority for join. This document provides a mechanism by which a RPL DODAG root can disable join announcements, or adjust the base priority for join operation.

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 August 6, 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	3
2. Protocol Definition	3
3. Security Considerations	4
4. Privacy Considerations	4
5. IANA Considerations	4
6. Acknowledgements	4
7. References	4
7.1. Normative References	4
7.2. Informative References	5
Appendix A. Change history	6
Author's Address	6

1. Introduction

[RFC7554] describes the use of the time-slotted channel hopping (TSCH) mode of [ieee802154]. [I-D.ietf-6tisch-minimal-security] and [I-D.ietf-6tisch-dtsecurity-secure-join] describe mechanisms by which a new node (the "pledge") can use a friendly router as a Join Proxy. [I-D.richardson-6tisch-join-enhanced-beacon] describes an extension to the 802.15.4 Enhanced Beacon that is used by a Join Proxy to announce its existence such that Pledges can find them.

It has become clear that not every routing member of the mesh ought to announce itself as a Join Proxy. There are a variety of local reasons by which a 6LR might not want to provide the Join Proxy function. They include available battery power, already committed network bandwidth, and also total available memory available for Join proxy neighbor cache slots.

There are other situations where the operator of the network would like to selective enable or disable the join process in a particular DODAG.

As the join process involves permitting unencrypted traffic into the best effort part of a (TSCH) network, it would be better to have the join process off when no new nodes are expected.

A network operator might also be able to recognize when certain parts of the network are overloaded and can not accomodate additional join traffic, and it would like to adjust the join priority among all nodes in the subtree of a congested link.

This document describes an RPL DIO option that can be used to announce a minimum join priority.

1.1. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in BCP 14, RFC 2119 [RFC2119] and indicate requirement levels for compliant STuPiD implementations.

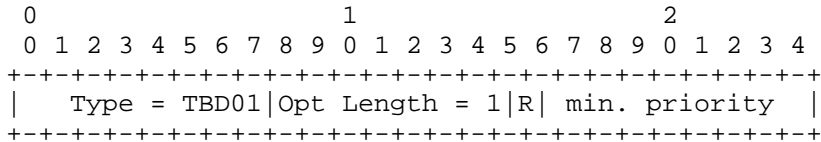
In addition, the terminology of [I-D.ietf-6tisch-terminology] and from [I-D.ietf-anima-voucher] are used.

2. Protocol Definition

The following option is defined to transmission in the DIO issued by the DODAG root. It may also be added by a router on part of the subtree as a result of some (out of scope for this document) management function.

6LRs that see this DIO Option SHOULD increment the minimum priority if they observe congestion on the channel used for join traffic. (TODO: how much? Do we need to standardize this?)

A 6LR which would otherwise be willing to act as a Join Proxy, will examine the minimum priority field, and to that number, add any additional local consideration (such as upstream congestion). The resulting priority, if less than 0x7f should enable the Join Proxy function.



min.priority a 7 bit field which provides a base value for the Enhanced Beacon Join priority. A value of 0x7f (127) disables the Join Proxy function entirely.

R a reserved bit that SHOULD be set to 0 by senders, and MUST be ignored by receivers. The reserved bit SHOULD be copied to options created.

3. Security Considerations

As per [RFC7416], RPL control frames either run over a secured layer 2, or use the [RFC6550] Secure DIO methods. This option can be placed into either a "clear" (layer-2 secured) DIO, or a layer-3 Secure DIO. As such this option will have both integrity and confidentiality mechanisms applied to it.

A malicious node (that was part of the RPL control plane) could see these options and could, based upon the observed minimal join priority signal a confederate that it was a good time to send malicious join traffic.

A malicious node (that was part of the RPL control plane) could also send DIOs with a different minimal join priority which would cause downstream mesh routers to change their Join Proxy behaviour. Lower minimal priorities would cause downstream nodes to accept more pledges than the network was expecting, and higher minimal priorities cause the join process to stall.

The use of layer-2 or layer-3 security for RPL control messages prevents the above two attacks.

4. Privacy Considerations

There are no new privacy issues caused by this extension.

5. IANA Considerations

Allocate a new number TBD01 from Registry RPL Control Message Options. This entry should be called Minimum Join Priority.

6. Acknowledgements

none so far.

7. References

7.1. Normative References

[I-D.ietf-6tisch-minimal-security]
Vucinic, M., Simon, J., Pister, K., and M. Richardson,
"Minimal Security Framework for 6TiSCH", draft-ietf-
6tisch-minimal-security-04 (work in progress), October
2017.

- [I-D.richardson-6tisch-join-enhanced-beacon]
Dujovne, D. and M. Richardson, "IEEE802.15.4 Informational Element encapsulation of 6tisch Join Information", draft-richardson-6tisch-join-enhanced-beacon-03 (work in progress), January 2018.
- [ieee802154]
IEEE Standard, ., "802.15.4-2015 - IEEE Standard for Low-Rate Wireless Personal Area Networks (WPANs)", 2015, <<http://standards.ieee.org/findstds/standard/802.15.4-2015.html>>.
- [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>>.
- [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>>.
- [RFC7554] Watteyne, T., Ed., Palattella, M., and L. Grieco, "Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement", RFC 7554, DOI 10.17487/RFC7554, May 2015, <<https://www.rfc-editor.org/info/rfc7554>>.

7.2. 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.
- [I-D.ietf-6tisch-dtsecurity-secure-join]
Richardson, M., "6tisch Secure Join protocol", draft-ietf-6tisch-dtsecurity-secure-join-01 (work in progress), February 2017.

[I-D.ietf-6tisch-terminology]

Palattella, M., Thubert, P., Watteyne, T., and Q. Wang,
"Terminology in IPv6 over the TSCH mode of IEEE
802.15.4e", draft-ietf-6tisch-terminology-09 (work in
progress), June 2017.

[I-D.ietf-anima-voucher]

Watson, K., Richardson, M., Pritikin, M., and T. Eckert,
"Voucher Profile for Bootstrapping Protocols", draft-ietf-
anima-voucher-07 (work in progress), January 2018.

[RFC8137] Kivinen, T. and P. Kinney, "IEEE 802.15.4 Information
Element for the IETF", RFC 8137, DOI 10.17487/RFC8137, May
2017, <<https://www.rfc-editor.org/info/rfc8137>>.

Appendix A. Change history

version 00.

Author's Address

Michael Richardson
Sandelman Software Works

Email: mcr+ietf@sandelman.ca

6lo Working Group
Internet-Draft
Intended status: Informational
Expires: February 9, 2020

M. Richardson
Sandelman Software Works
August 08, 2019

Enabling secure network enrollment in RPL networks
draft-richardson-6tisch-roll-enrollment-priority-03

Abstract

[I-D.ietf-6tisch-enrollment-enhanced-beacon] defines a method by which a potential [I-D.ietf-6tisch-minimal-security] join proxy can announce itself as a available for new Pledges to Join a network. The announcement includes a priority for join. This document provides a mechanism by which a RPL DODAG root can disable join announcements, or adjust the base priority for join operation.

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 9, 2020.

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
1.1. Terminology	3
2. Protocol Definition	3
3. Security Considerations	4
4. Privacy Considerations	4
5. IANA Considerations	4
6. Acknowledgements	4
7. References	4
7.1. Normative References	4
7.2. Informative References	5
Appendix A. Change history	6
Author's Address	6

1. Introduction

[RFC7554] describes the use of the time-slotted channel hopping (TSCH) mode of [ieee802154]. [I-D.ietf-6tisch-minimal-security] and [I-D.ietf-6tisch-dtsecurity-secure-join] describe mechanisms by which a new node (the "pledge") can use a friendly router as a Join Proxy. [I-D.ietf-6tisch-enrollment-enhanced-beacon] describes an extension to the 802.15.4 Enhanced Beacon that is used by a Join Proxy to announce its existence such that Pledges can find them.

It has become clear that not every routing member of the mesh ought to announce itself as a Join Proxy. There are a variety of local reasons by which a 6LR might not want to provide the Join Proxy function. They include available battery power, already committed network bandwidth, and also total available memory available for Join proxy neighbor cache slots.

There are other situations where the operator of the network would like to selective enable or disable the join process in a particular DODAG.

As the join process involves permitting unencrypted traffic into the best effort part of a (TSCH) network, it would be better to have the join process off when no new nodes are expected.

A network operator might also be able to recognize when certain parts of the network are overloaded and can not accomodate additional join traffic, and it would like to adjust the join priority among all nodes in the subtree of a congested link.

This document describes an RPL DIO option that can be used to announce a minimum join priority. Each potential Join Proxy would this value as a base on which to add (decreasing likely hood of attracting traffic) values relating to local conditions.

A network operator can set this value to the maximum value allowed, effectively disable all new join traffic.

1.1. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in BCP 14, RFC 2119 [RFC2119] and indicate requirement levels for compliant STuPiD implementations.

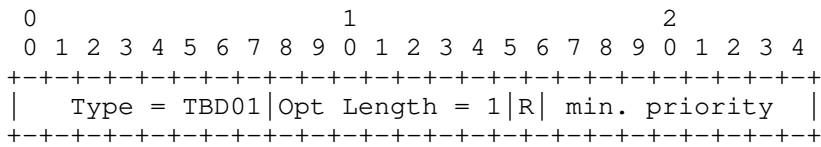
In addition, the terminology of [I-D.ietf-6tisch-terminology] and from [RFC8366] are used.

2. Protocol Definition

The following option is defined to transmission in the DIO issued by the DODAG root. It may also be added by a router on part of the sub-tree as a result of some (out of scope for this document) management function.

6LRs that see this DIO Option SHOULD increment the minimum priority if they observe congestion on the channel used for join traffic. (TODO: how much? Do we need to standardize this?)

A 6LR which would otherwise be willing to act as a Join Proxy, will examine the minimum priority field, and to that number, add any additional local consideration (such as upstream congestion). The resulting priority, if less than 0x7f should enable the Join Proxy function.



min.priority a 7 bit field which provides a base value for the Enhanced Beacon Join priority. A value of 0x7f (127) disables the Join Proxy function entirely.

R a reserved bit that SHOULD be set to 0 by senders, and MUST be ignored by receivers. The reserved bit SHOULD be copied to options created.

3. Security Considerations

As per [RFC7416], RPL control frames either run over a secured layer 2, or use the [RFC6550] Secure DIO methods. This option can be placed into either a "clear" (layer-2 secured) DIO, or a layer-3 Secure DIO. As such this option will have both integrity and confidentiality mechanisms applied to it.

A malicious node (that was part of the RPL control plane) could see these options and could, based upon the observed minimal join priority signal a confederate that it was a good time to send malicious join traffic.

A malicious node (that was part of the RPL control plane) could also send DIOs with a different minimal join priority which would cause downstream mesh routers to change their Join Proxy behaviour. Lower minimal priorities would cause downstream nodes to accept more pledges than the network was expecting, and higher minimal priorities cause the join process to stall.

The use of layer-2 or layer-3 security for RPL control messages prevents the above two attacks.

4. Privacy Considerations

There are no new privacy issues caused by this extension.

5. IANA Considerations

Allocate a new number TBD01 from Registry RPL Control Message Options. This entry should be called Minimum Join Priority.

6. Acknowledgements

This has been reviewed by Pascal Thubert and Thomas Wattenye.

7. References

7.1. Normative 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-02 (work in progress), March 2019.
- [I-D.ietf-6tisch-minimal-security]
Vucinic, M., Simon, J., Pister, K., and M. Richardson, "Minimal Security Framework for 6TiSCH", draft-ietf-6tisch-minimal-security-12 (work in progress), July 2019.
- [ieee802154]
IEEE Standard, ., "802.15.4-2015 - IEEE Standard for Low-Rate Wireless Personal Area Networks (WPANs)", 2015, <<http://standards.ieee.org/findstds/standard/802.15.4-2015.html>>.
- [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>>.
- [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>>.
- [RFC7554] Watteyne, T., Ed., Palattella, M., and L. Grieco, "Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement", RFC 7554, DOI 10.17487/RFC7554, May 2015, <<https://www.rfc-editor.org/info/rfc7554>>.

7.2. 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-24 (work in progress), July 2019.

- [I-D.ietf-6tisch-dtsecurity-secure-join]
Richardson, M., "6tisch Secure Join protocol", draft-ietf-6tisch-dtsecurity-secure-join-01 (work in progress), February 2017.
- [I-D.ietf-6tisch-terminology]
Palattella, M., Thubert, P., Watteyne, T., and Q. Wang, "Terms Used in IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-terminology-10 (work in progress), March 2018.
- [RFC8137] Kivinen, T. and P. Kinney, "IEEE 802.15.4 Information Element for the IETF", RFC 8137, DOI 10.17487/RFC8137, May 2017, <<https://www.rfc-editor.org/info/rfc8137>>.
- [RFC8366] Watsen, K., Richardson, M., Pritikin, M., and T. Eckert, "A Voucher Artifact for Bootstrapping Protocols", RFC 8366, DOI 10.17487/RFC8366, May 2018, <<https://www.rfc-editor.org/info/rfc8366>>.

Appendix A. Change history

version 00.

Author's Address

Michael Richardson
Sandelman Software Works

Email: mcr+ietf@sandelman.ca

ROLL
Internet-Draft
Updates: 6282,6550,6775 (if approved)
Intended status: Standards Track
Expires: July 26, 2018

P. Thubert, Ed.
Cisco
January 22, 2018

RPL-BIER
draft-thubert-roll-bier-01

Abstract

This specification extends RPL to provide unicast and multicast routing based on bitStrings such as used in Bit Index Explicit Replication and its source-routed Traffic Engineering variant, which correspond to RPL storing and Non-Storing Modes respectively.

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 July 26, 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	5
3. Applicability	5
4. Extensions to RFC 6550	5
4.1. RPL-BIER MOPs	6
4.1.1. RPL-BIER Non-Storing Mode	6
4.1.2. RPL-BIER Storing Mode	6
4.2. BitString Information	7
5. Updating the 6LoWPAN Framework	8
5.1. Extensions to RFC 6775	8
5.2. Extensions to RFC 6282	9
5.3. New Neighbor Discovery Options and Messages	9
5.3.1. 6LoWPAN ND Bit Position Option	9
5.3.2. Address Mapping Message	10
6. BitString formats	11
6.1. Bit-by-bit BitStrings	11
6.1.1. Allocating a Bit Position in a Bit-by-bit BitString	12
6.1.2. Aggregation of Bit-by-bit BitStrings	13
6.1.3. Forwarding Based on Bit-by-bit BitStrings	13
6.1.4. Reliable Multicast based on Bit-by-bit BitStrings	14
6.2. Bloom Filters	14
6.2.1. Computing and Saving Bloom Filters	15
6.2.2. Forwarding based on Bloom Filters	15
6.2.3. Hash Functions Distribution	15
7. Implementation Status	15
8. Security Considerations	15
9. IANA Considerations	15
10. Acknowledgments	15
11. References	15
11.1. Normative References	16
11.2. Informative References	16
Author's Address	18

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. In order to cope with lossy transmissions, RPL forms Direction-Oriented Directed Acyclic Graphs (DODAGs) that provide multiple forwarding solutions to most of the intermediate hops.

Because it is designed to adapt to fuzzy connectivity with lazy control, RPL can only provide a best effort routability, connecting most of the LLN nodes, most of the time.

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. RPL also leverages Routing Stretch to reduce further the amount of control traffic and routing state that is required to operate the 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.

RPL enables various trade-offs between routing stretch, amount of routing state and packet size, with the introduction of different modes of operation (MOPs):

- o With Reactive Discovery of Point-to-Point (P2P) Routes (aka on-demand) [RFC6997][I-D.ietf-roll-aodv-rpl], a limited number of routes are optimized on-demand, between select pairs of devices for which a service with some particular characteristics is needed.
- o In Storing Mode of operation (aka Storing Mode), Multipoint to Point (MP2P) routes from the LLN nodes to the root and Point to Multipoint (P2MP) routes from the root to the LLN nodes are optimized, but P2P routes between LLN nodes are stretched via a common parent. In Storing Mode, RPL requires that nodes maintain routing information for the maximum number of child nodes in their sub-DODAG. If a network is composed of similar nodes, this means that each node should be able to store a number of routes that is in the order of the total number of nodes in the network.
- o In Non-Storing Mode of operation (aka Non-Storing Mode), MP2P and P2MP routes are also optimized, but downwards routes can only be computed by the root and P2MP forwarding relies on strict source-routing. This increases the size of the packets and limits the depth of the DODAG. The Non-Storing Mode also results in additional stretch for P2P routes, whereby all packets must transit via the root following the default route all the way up, to be then source-routed down.

In order to alleviate the issues above and improve the existing multicast operations, this specification introduces the use of bitStrings in RPL LLNs. BitStrings are already used in the art as a datapath analog to one or more IPv6 [RFC8200] addresses:

- o With the Bit Index Explicit Replication (BIER), introduced in the "BIER Architecture" [RFC8279], each bit in a bitString represents a particular destination. This specification introduces a RPL-

- BIER Storing Mode that applies BIER operations to RPL Storing Mode.
- o "Traffic Engineering for Bit Index Explicit Replication" [I-D.eckert-bier-te-arch] (BIER-TE) adds support for Traffic Engineering by explicit hop-by-hop forwarding and loose hop forwarding of packets along a unicast route. With BIER-TE, each bit in a bitStrings represents a particular intermediate link. This specification introduces a RPL-BIER Non-Storing Mode that applies BIER-TE operations to RPL Non-Storing Mode.

This specification provides new signaling in RPL to enable RPL-BIER operations. In addition to classical bitStrings, this specification proposes a new technique that derives from Bloom Filters. This technique provides elasticity to the size of the bitString, which is not constrained to a fixed number of entries, and a trade-off between the number of bits that are needed for routing and the chances to waste energy forwarding down a path where no target exist. The Bloom Filters mechanism applies to RPL-BIER in both modes of operations.

In order to carry routing information in a concise fashion in a Low-Power Wireless Personal Area Network (6LoWPAN) for Route-Over use cases, the 6LoWPAN adaptation layer framework [RFC4944] [RFC6282] was extended with the 6LoWPAN Routing Header (6LoRH) specification [RFC8138], which uses the 6LoWPAN Paging Dispatch [RFC8025]. The original specification includes the formats necessary for RPL such as the Source Route Header (SRH) and is intended to be extended for additional routing artifacts. A companion document to this, the "6LoRH for BitStrings" [I-D.thubert-6lo-bier-dispatch], specification, proposes new 6LoRH formats to enable the concise encoding of the BIER bitStrings and of the Bloom Filters described therein.

In the current practice of LLN networks, the Non-Storing Mode is largely favored, because it does not place a restriction on how large a network formed of a particular device can become. One major benefit of introducing bitStrings is that the amount of state that is required for routing in Storing Mode is no more growing in the order of the total number of nodes in the network but linearly with the number of children attached to the RPL router. In other words, the maximum number of children that a router is willing to accept determines both the size of the Neighbor Cache for 6LoWPAN Neighbor Discovery (6LoWPAN ND) [RFC6775][I-D.ietf-6lo-rfc6775-update] and the size of the routing table that is required in RPL-BIER Storing Mode.

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].

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].

The term "byte" is used in its now customary sense as a synonym for "octet".

"RPL", "RPL Packet Information" (RPI) and "RPL Instance" are defined in the "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks" [RFC6550] specification.

The terms Bit-Forwarding Egress Routers (BFR), BFR-id and bitString are defined in [RFC8279]. A bitString indicates a continuous sequence of bits indexed by an offset in the sequence. The leftmost bit is bit 0 and corresponds to the value 0x80 of the leftmost byte in the BitString. With this specification, a bitString maybe used to encode one or more unsigned integer(s) as a bit position in the bitString (bit-by-bit), or a Bloom filter.

3. Applicability

BIER and other bit-indexed methods that would leverage bitStrings will generally require additional information in the packet to complement the bitString. For instance, BIER has the concept of a BFR-id and an Entropy value in the BIER header. Since those additional fields depend on the bit-indexed method, they are expected to be transported separately from the bitString. This specification concentrates on the bitString and a group identifier which enables a network to grow beyond the size of one bitString.

TBD Do we need entropy ?

4. Extensions to RFC 6550

This specification introduces two new modes of operation for RPL, the RPL-BIER Non-Storing Mode which is discussed in Section 4.1.1 and the RPL-BIER Storing Mode which is discussed in Section 4.1.2. A new

Control Message Options (CMO) is introduced to transport the bitStrings in Section 4.2.

4.1. RPL-BIER MOPs

In RPL-BIER modes of operation, one or more RPL Target Option are replaced by a new BitString Information Option which represent the advertised target(s) by a combination of a bitString and control information.

4.1.1. RPL-BIER Non-Storing Mode

In Non-Storing RPL-BIER, a bit in classical BIER bit-by-bit bitStrings (see Section 6.1) or a set of bits in a Bloom Filter (see Section 6.2) is associated to a next-hop from the perspective of an intermediate router. RPL Non-Storing Mode DAO messages are used to advertise the relation between a target and its parent (transit) directly to the root.

If multiple Targets Options were to be placed consecutively to factorize a Transit Information Option (TIO) in a classical RPL Non-Storing Mode DAO message, they are replaced by a single BIO with the aggregated bitString that represents all these targets.

4.1.2. RPL-BIER Storing Mode

In RPL-BIER Storing Mode, a bit in classical BIER Bit-by-bit bitStrings (see Section 6.1) or a set of bits in a Bloom Filter (see Section 6.2) is associated to a final destination. RPL Storing Mode DAO messages are used to advertise recursively the targets to the parent(s) all the way to the root.

The BitString Information Option(s) in the DAO message contain collectively an aggregated bitString that represents the advertising node and all of its descendants. Parents save the bitString per child, and use it to forward down the DODAG as discussed in Section 6.1.3.

The Transit Information Option is not used. The lack of transit information is compensated by a more frequent transmission of DAO messages and a full use of the RPL data plane loop avoidance and inconsistency detection mechanisms (section 11.2 of [RFC6550]), in collaboration with a periodic 6LoWPAN ND (re)registration that maintains the 6LBR and the root aware of which devices are actually present in the network with the associated lifetime and sequence information.

4.2. BitString Information

Section 6.7 of [RFC6550] specifies optional CMOs 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 optional fields; 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 a new CMO, the BitString Information option (BIO). A BIO is used as an alternate to one or more Target Option(s) in Storing and Non-Storing Mode DAOs using bit-by-bit bitStrings, and its format is as follows:

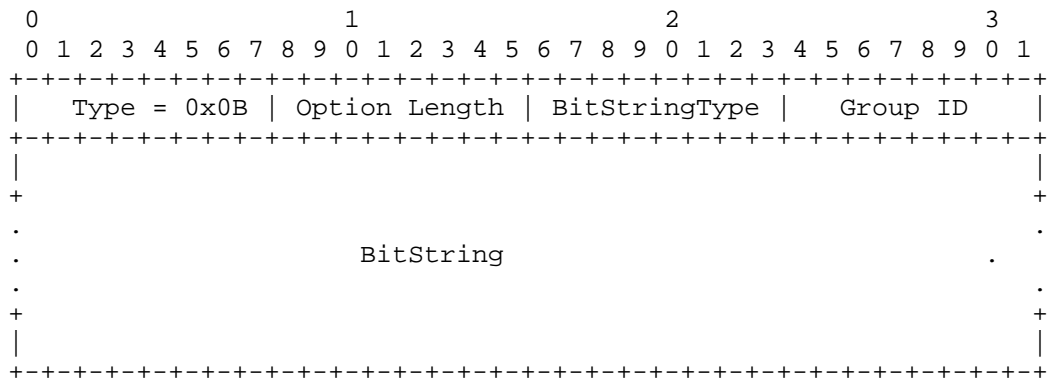


Figure 1: BitString Information option format

- Option Type: 0x0B (to be confirmed by IANA)
- Option Length: In bytes; variable, depending on the Type.
- BitString Type: Indicates the size of the bitString field as indicated in Table 1 below.

BitString Type	BitString Size
15	8 bits
16	16 bits
17	48 bits
18	96 bits
19	160 bits

Group ID : 8-bit unsigned integer. The Group ID for the bit-by-bit bitString.

BitString: 8 to 160 bits, depending on the Type.

5. Updating the 6LoWPAN Framework

5.1. Extensions to RFC 6775

It is noted that RPL does not provide a Duplicate Address Detection (DAD) and relies on 6LoWPAN ND to ensure that addresses are unique within the network. For that purpose, a 6LoWPAN Border Router (6LBR) maintains the list of addresses that are currently in use in the network that it serves. In the case of a RPL LLN, the 6LBR is typically collocated with the RPL root, and serves the RPL DODAG. With 6LoWPAN ND[RFC6775] [I-D.ietf-6lo-rfc6775-update], a Duplicate Address Request (DAR) / Duplicate Address Confirmation (DAC) exchange is used to perform the DAD operation. Scalability is achieved by federating multiple DODAGs with IPv6 Backbone Routers (6BBRs) [I-D.ietf-6lo-backbone-router].

In that context, it makes sense to also leverage the 6LBR to ensure that a tuple (groupID, bitString) is assigned unequivocally to an IPv6 address for the bit-by-bit operation. This specification extends the role of a 6LBR to 1) assign the tuple to the IPv6 address and 2) resolve an IPv6 address into a tuple. To achieve this, RFC 6775 is updated as follows:

- o A BIER Address Resolution (BAR) / BIER Address Confirmation (BAC) exchange is introduced for the purpose of the bitString lookup operation (see Section 6.1.1).
- o A new Bit Position Option (BPO) is introduced to carry the corresponding bit position bitString in 6LoWPAN ND exchanges. The BPO is transported in BAC, NA and DAC messages in response to BAR, NS and DAR messages, respectively (see Section 5.3.1).

5.2. Extensions to RFC 6282

This specification also extends the 6LoWPAN framework with the capability to transform an address into a tuple (Control field, bitString) as part of the 6LoWPAN Header Compression [RFC6282] (6LoWPAN HC). Since the 6LBR and the Header Compression functions are typically collocated, the latter may exploit local tables built by the former to map a destination IPv6 address into a bitString.

In Storing Mode, P2P stretched routing via a common parent can be obtained if the destination is expressed as a tuple (Control field, bitString). This can be achieved with a BAR/BAC exchange with the 6LBR.

5.3. New Neighbor Discovery Options and Messages

In order to allocate and lookup a bitString, this specification extends 6LoWPAN ND with the following new messages and formats.

5.3.1. 6LoWPAN ND Bit Position Option

The Bit Position Option (BPO) is intended to be used to return a bitString and related information in 6LoWPAN ND BA, DAC and BAC messages with a Status of 0 indicating success, the NA and DAC messages transporting an Address Registration Option (ARO) indicating the IPv6 address that is mapped with the bit position. Its format is as follows:

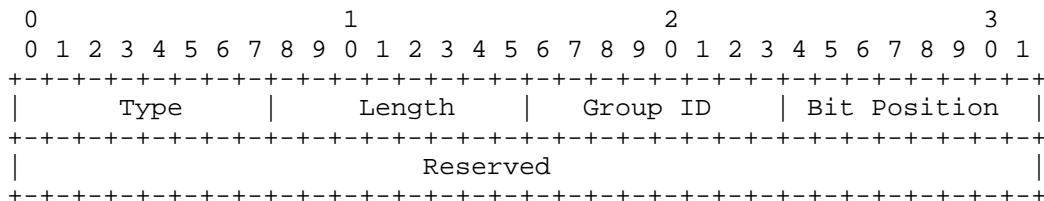


Figure 2: Bit Position Option format

Option Fields

- Type: 38 (to be confirmed by IANA)
- Length: 8-bit unsigned integer. The length of the option (including the type and length fields) in units of 8 bytes. The Length MUST be set to 1.
- Group ID: 8-bit unsigned integer. The GroupID for the bit-by-bit bitString.
- Bit Position: 8-bit unsigned integer. The offset in the the bit-by-bit bitString.

5.3.2. Address Mapping Message

For the multihop lookup exchanges between a 6LR that needs to perform a Header Compression including the bitString for the destination, and its 6LBR, which knows if the mapping exists and what it is, this specification introduces two new ICMPv6 message called the BIER Address Resolution (BAR) and the BIER Address Confirmation (BAC). We avoid reusing the NS and NA messages for this purpose, since these messages are not subject to the hop limit=255 check as they are forwarded by intermediate 6LRs.

The BAR and BAC use the same ICMPv6 type value which this specification, allocates for a generic Address Mapping service, but use different Codes. This is done to save addressable space in the ICMPv6 type values which is getting crowded, and because it is expected that in the future, other mapping techniques may be needed as well.

The Status field and Information Lifetime are not meaningful in the BAR message. When and only when the BAC message carries a status of 0, indicating success, the Information Lifetime must contain valid information and the message must carry a Bit Position Option (Section 5.3.1).

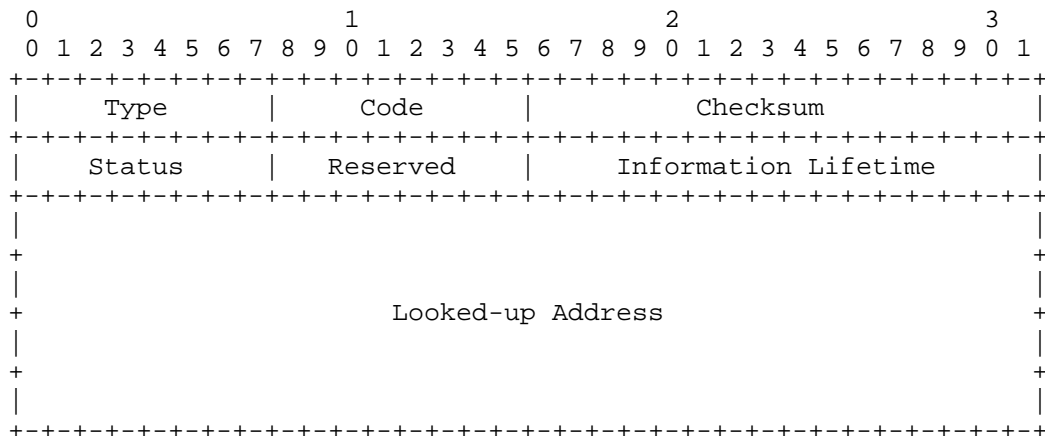


Figure 3: Address Mapping Message format

Message Fields

- Type: 160 (to be confirmed by IANA)
- Code: 8-bit unsigned integer. 1 for BAR and 2 for BAC. Other values are reserved.
- Checksum:: The ICMP checksum. See [RFC4443].

Status: 8-bit unsigned integer. Indicates the status of the lookup in the BAC. See Table 2 below.

Status	Description
0	Success
1	Looked-up Address not Found
2..255	Allocated using Standards Action [RFC5226]

Information Lifetime: 16-bit unsigned integer. The length of time in units of 60 seconds (relative to the time the packet is received) that this mapping information is valid. A value of all zero bits (0x0) assumes a default value of 10,000 (~one week).

Looked-up Address: 128-bit field. Carries the IPv6 address that is being mapped.

6. BitString formats

This specification introduces two BitString formats, the bit-by-bit and the Bloom Filter.

6.1. Bit-by-bit BitStrings

In the bit-by-bit case, each bit is mapped in an unequivocal fashion with a single addressable resource in the network. In RPL-BIER Storing Mode, this is an IPv6 address as advertised in RPL Storing Mode DAO messages, whereas in RPL-BIER Non-Storing Mode, this is a parent-child relationship as advertised in RPL Non-Storing Mode DAO messages.

if every node in a large network is given one or more bits in a bit-by-bit bitString, then the bitString may grow very large and lead to undesirably large overhead in the data plane. BIER allows to divide a potentially the very large abstract bitString into segments, aka groups, indexed by a groupID.

In the protocol elements that use a bitString, only the relevant group(s) are transported, and the advertised bitString is in fact the segment that corresponds to the groupID. It results that a bit position in the large abstract bitString is advertised either as the tuple (groupID, segment) or the tuple (groupID, bit position within segment).

For simplicity, when dealing with protocol elements, the specification uses the term `bitString` to refer to the tuple (`groupID`, `segment`) and a bitwise operation between `bitStrings` is really a bitwise operation between segments of a same `groupID`.

TBD: do we allow multiple (`groupID`, `bitString`) tuples in one packet?

6.1.1.1. Allocating a Bit Position in a Bit-by-bit BitString

Several methods may be used to associate a bit in a `bitString` to an IPv6 address. In order to guarantee interoperability, this specification RECOMMENDS that all implementations provide at least the method described therein.

With 6LoWPAN ND, a 6LoWPAN Node (6LN) registers with address(es) to one or more 6LoWPAN Router [RFC6775] to perform Duplicate Address Detection (DAD). As part of the DAD process, the 6LN validates that a Global Unicast Address (GUA) or a Unique Local Address (ULA) is effectively unique using a unicast DAR/DAC exchange with the 6LBR (this procedure is updated in [I-D.ietf-6lo-rfc6775-update]).

In a network that supports this specification, the 6LBR maintains a configurable number of groups (up to 32, indexed by `groupID`), and for each group, it maintains a pool of free bit positions. Upon a new registration, the 6LBR selects a `groupID` and a free bit position and associates it to the IPv6 address.

The `bitString Size` in any given group should be configurable. The policy for selecting the `groupID` for a new registration is left to implementation. It is noted that in large networks that require multiple groups, associating groups with immediate children of the root may be an option to limit the number of groups that the RPL routers must be aware of.

If the 6LBR accepts the registration, then it returns a DAC message with a status of 0 indicating success, adding a 6LoWPAN ND Bit Position Option (Section 5.3.1) to the DAC message to indicate the `groupID` and bit.

The 6LR maintains a binding cache entry (BCE) for the 6LN based on successful DAC messages. With this specification, the 6LR also stores the matching between the address and the `bitString` and uses it for searching its children when forwarding packets in Non-Storing Mode (see Section 6.1.3).

If the 6LN child does not support the BIER encoding (e.g.[I-D.thubert-6lo-bier-dispatch]), then the packet is converted in a format that the child supports (e.g.[RFC8138]).

6.1.2. Aggregation of Bit-by-bit BitStrings

BitStrings are aggregated by a 'OR' operation so that all the bits that are set in either bitString is set in the resulting bitString. In the concise form of a tuple (groupID, bitString), the aggregation is done on a group-by-group basis, between segments of a same group.

In RPL-BIER Storing Mode, the bit-by-bit BitStrings are passed from child to parent using DAO messages, in a fashion similar to RPL Storing Mode [RFC6550]. The BitString Information option (Figure 1) is used in replacement of the Target option. A DAO message contains one BIO per group, and the parent that receives the messages associates the BIO information to the advertising child. In order to build a DAO message, the parent regenerates its own BIO, one per group, by aggregating the bitStrings from all of its children with its own, and places the resulting BIOs in the DAO message.

6.1.3. Forwarding Based on Bit-by-bit BitStrings

Forwarding is based on matching a bitString in a packet with those of children. For unicast packets, only one matching child gets the packet. For multicast packets, all matching children get a copy. Matches are found by scanning all children and performing bitwise operations as follows.

In order to search for a match, a reference bitString is initialized with the destination bitString in the packet. A match is found with a child if the bitwise 'AND' between the reference bitString and the bitString stored for that child does not result in a NULL bitString of all zeroes.

In Non-Storing Mode, a packet is copied to all matching children, which are found by trying all children.

In Non-Storing Mode, if a child is selected for forwarding, then an 'XOR' operation is performed between the reference bitString and the bitString resulting from the 'AND' operation. If the 'XOR' operation does not result in a NULL bitString, denoting that more children should get the packet, then the result of the 'XOR' operation becomes the new reference bitString and the search continues. The 'XOR' operation allows to stop the search loop as soon as all matches are found; it also avoids forwarding twice to a same destination along different downwards path in the DODAG.

6.1.4. Reliable Multicast based on Bit-by-bit BitStrings

Multicast from the root to a collection of target 6LNs can be made reliable with the following operation:

A multicast packet is identified by a unique packetID which is also found in the acknowledgments. The root signals the set of targets with a destination bitString that has the bits set for each of them, and the message is forwarded as described on Section 6.1.3.

Listeners acknowledge with an acknowledgment packet that contains the same information, the packetID and the bitString representing the listener. The bitStrings in acknowledgment packets are aggregated recursively on the way back as indicated in Section 6.1.2.

The root aggregates the bitStrings from its children into one acknowledgment bitString. It then checks that the acknowledgment bitString is correct, by an 'AND' operation with the destination bitString that should result in the acknowledgment bitString. If this is not the case, bits that are set in the acknowledgment bitString and not in the destination bitString are in the acknowledgment bitString.

The root generates the bitString of the devices that did not acknowledge the multicast message by a bitwise 'XOR' operation between the destination bitString and the acknowledgment bitString, and may use it to selectively retry the multicast.

6.2. Bloom Filters

A Bloom Filter can be seen as an additional compression technique for the bitString representation. A Bloom Filter may generate false positives, which, in the case of BIER, result in undue forwarding of a packet down a path where no listener exists.

As an example, the Constrained-Cast [I-D.bergmann-bier-ccast] specification employs Bloom Filters as a compact representation of a match or non-match for elements in a set that may be larger than the number of bits in the BitString.

In the case of a Bloom Filter, a number of Hash functions must be run to obtain a multi-bit signature of an encoded element. This specification uses the 5-bits Control field to signal an Identifier of the set of Hash functions being used to generate a certain bitString, so as to enable the migration from a set of Hash functions to the next.

6.2.1. Computing and Saving Bloom Filters

6.2.2. Forwarding based on Bloom Filters

6.2.3. Hash Functions Distribution

7. Implementation Status

TBD

8. Security Considerations

TBD

9. IANA Considerations

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

Code	Description	Reference
0x0B	bitString	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
6	RPL-BIER Non-Storing Mode of operation	This document
7	RPL-BIER Storing Mode of operation	This document

DIO Mode of operation

10. Acknowledgments

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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.

11.2. Informative References

- [I-D.bergmann-bier-ccast] Bergmann, O., Bormann, C., Gerdes, S., and H. Chen, "Constrained-Cast: Source-Routed Multicast for RPL", draft-bergmann-bier-ccast-02 (work in progress), October 2016.

- [I-D.eckert-bier-te-arch]
Eckert, T., Cauchie, G., Braun, W., and M. Menth, "Traffic Engineering for Bit Index Explicit Replication BIER-TE", draft-eckert-bier-te-arch-06 (work in progress), November 2017.
- [I-D.ietf-6lo-backbone-router]
Thubert, P., "IPv6 Backbone Router", draft-ietf-6lo-backbone-router-05 (work in progress), January 2018.
- [I-D.ietf-6lo-rfc6775-update]
Thubert, P., Nordmark, E., Chakrabarti, S., and C. Perkins, "An Update to 6LoWPAN ND", draft-ietf-6lo-rfc6775-update-11 (work in progress), December 2017.
- [I-D.ietf-roll-aodv-rpl]
Anamalamudi, S., Zhang, M., Sangi, A., Perkins, C., and S. Anand, "Asymmetric AODV-P2P-RPL in Low-Power and Lossy Networks (LLNs)", draft-ietf-roll-aodv-rpl-02 (work in progress), September 2017.
- [I-D.thubert-6lo-bier-dispatch]
Thubert, P., Brodard, Z., Jiang, H., and G. Texier, "A 6LoRH for BitStrings", draft-thubert-6lo-bier-dispatch-04 (work in progress), January 2018.
- [RFC4944] 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>>.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <<https://www.rfc-editor.org/info/rfc5226>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.

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: 6282,6550,6775 (if approved)
Intended status: Standards Track
Expires: January 25, 2019

P. Thubert, Ed.
Cisco
July 24, 2018

RPL-BIER
draft-thubert-roll-bier-02

Abstract

This specification extends RPL to provide unicast and multicast routing based on bitStrings such as used in Bit Index Explicit Replication and its source-routed Traffic Engineering variant, which correspond to RPL storing and Non-Storing Modes respectively.

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 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.	Introduction	2
2.	Terminology	5
3.	Applicability	5
4.	Extensions to RFC 6550	5
4.1.	RPL-BIER MOPs	6
4.1.1.	RPL-BIER Non-Storing Mode	6
4.1.2.	RPL-BIER Storing Mode	6
4.2.	BitString Information	7
5.	Updating the 6LoWPAN Framework	8
5.1.	Extensions to RFC 6775	8
5.2.	Extensions to RFC 6282	9
5.3.	New Neighbor Discovery Options and Messages	9
5.3.1.	6LoWPAN ND Bit Position Option	9
5.3.2.	Address Mapping Message	10
6.	BitString formats	11
6.1.	Bit-by-bit BitStrings	11
6.1.1.	Allocating a Bit Position in a Bit-by-bit BitString	12
6.1.2.	Aggregation of Bit-by-bit BitStrings	13
6.1.3.	Forwarding Based on Bit-by-bit BitStrings	13
6.1.4.	Reliable Multicast based on Bit-by-bit BitStrings	14
6.2.	Bloom Filters	14
6.2.1.	Computing and Saving Bloom Filters	15
6.2.2.	Forwarding based on Bloom Filters	15
6.2.3.	Hash Functions Distribution	15
7.	Implementation Status	15
8.	Security Considerations	15
9.	IANA Considerations	15
10.	Acknowledgments	15
11.	References	15
11.1.	Normative References	16
11.2.	Informative References	16
	Author's Address	18

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. In order to cope with lossy transmissions, RPL forms Direction-Oriented Directed Acyclic Graphs (DODAGs) that provide multiple forwarding solutions to most of the intermediate hops.

Because it is designed to adapt to fuzzy connectivity with lazy control, RPL can only provide a best effort routability, connecting most of the LLN nodes, most of the time.

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. RPL also leverages Routing Stretch to reduce further the amount of control traffic and routing state that is required to operate the 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.

RPL enables various trade-offs between routing stretch, amount of routing state and packet size, with the introduction of different modes of operation (MOPs):

- o With Reactive Discovery of Point-to-Point (P2P) Routes (aka on-demand) [RFC6997][I-D.ietf-roll-aodv-rpl], a limited number of routes are optimized on-demand, between select pairs of devices for which a service with some particular characteristics is needed.
- o In Storing Mode of operation (aka Storing Mode), Multipoint to Point (MP2P) routes from the LLN nodes to the root and Point to Multipoint (P2MP) routes from the root to the LLN nodes are optimized, but P2P routes between LLN nodes are stretched via a common parent. In Storing Mode, RPL requires that nodes maintain routing information for the maximum number of child nodes in their sub-DODAG. If a network is composed of similar nodes, this means that each node should be able to store a number of routes that is in the order of the total number of nodes in the network.
- o In Non-Storing Mode of operation (aka Non-Storing Mode), MP2P and P2MP routes are also optimized, but downwards routes can only be computed by the root and P2MP forwarding relies on strict source-routing. This increases the size of the packets and limits the depth of the DODAG. The Non-Storing Mode also results in additional stretch for P2P routes, whereby all packets must transit via the root following the default route all the way up, to be then source-routed down.

In order to alleviate the issues above and improve the existing multicast operations, this specification introduces the use of bitStrings in RPL LLNs. BitStrings are already used in the art as a datapath analog to one or more IPv6 [RFC8200] addresses:

- o With the Bit Index Explicit Replication (BIER), introduced in the "BIER Architecture" [RFC8279], each bit in a bitString represents a particular destination. This specification introduces a RPL-

- BIER Storing Mode that applies BIER operations to RPL Storing Mode.
- o "Traffic Engineering for Bit Index Explicit Replication" [I-D.eckert-bier-te-arch] (BIER-TE) adds support for Traffic Engineering by explicit hop-by-hop forwarding and loose hop forwarding of packets along a unicast route. With BIER-TE, each bit in a bitStrings represents a particular intermediate link. This specification introduces a RPL-BIER Non-Storing Mode that applies BIER-TE operations to RPL Non-Storing Mode.

This specification provides new signaling in RPL to enable RPL-BIER operations. In addition to classical bitStrings, this specification proposes a new technique that derives from Bloom Filters. This technique provides elasticity to the size of the bitString, which is not constrained to a fixed number of entries, and a trade-off between the number of bits that are needed for routing and the chances to waste energy forwarding down a path where no target exist. The Bloom Filters mechanism applies to RPL-BIER in both modes of operations.

In order to carry routing information in a concise fashion in a Low-Power Wireless Personal Area Network (6LoWPAN) for Route-Over use cases, the 6LoWPAN adaptation layer framework [RFC4944] [RFC6282] was extended with the 6LoWPAN Routing Header (6LoRH) specification [RFC8138], which uses the 6LoWPAN Paging Dispatch [RFC8025]. The original specification includes the formats necessary for RPL such as the Source Route Header (SRH) and is intended to be extended for additional routing artifacts. A companion document to this, the "6LoRH for BitStrings" [I-D.thubert-6lo-bier-dispatch], specification, proposes new 6LoRH formats to enable the concise encoding of the BIER bitStrings and of the Bloom Filters described therein.

In the current practice of LLN networks, the Non-Storing Mode is largely favored, because it does not place a restriction on how large a network formed of a particular device can become. One major benefit of introducing bitStrings is that the amount of state that is required for routing in Storing Mode is no more growing in the order of the total number of nodes in the network but linearly with the number of children attached to the RPL router. In other words, the maximum number of children that a router is willing to accept determines both the size of the Neighbor Cache for 6LoWPAN Neighbor Discovery (6LoWPAN ND) [RFC6775][I-D.ietf-6lo-rfc6775-update] and the size of the routing table that is required in RPL-BIER Storing Mode.

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].

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].

The term "byte" is used in its now customary sense as a synonym for "octet".

"RPL", "RPL Packet Information" (RPI) and "RPL Instance" are defined in the "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks" [RFC6550] specification.

The terms Bit-Forwarding Egress Routers (BFR), BFR-id and bitString are defined in [RFC8279]. A bitString indicates a continuous sequence of bits indexed by an offset in the sequence. The leftmost bit is bit 0 and corresponds to the value 0x80 of the leftmost byte in the BitString. With this specification, a bitString maybe used to encode one or more unsigned integer(s) as a bit position in the bitString (bit-by-bit), or a Bloom filter.

3. Applicability

BIER and other bit-indexed methods that would leverage bitStrings will generally require additional information in the packet to complement the bitString. For instance, BIER has the concept of a BFR-id and an Entropy value in the BIER header. Since those additional fields depend on the bit-indexed method, they are expected to be transported separately from the bitString. This specification concentrates on the bitString and a group identifier which enables a network to grow beyond the size of one bitString.

TBD Do we need entropy ?

4. Extensions to RFC 6550

This specification introduces two new modes of operation for RPL, the RPL-BIER Non-Storing Mode which is discussed in Section 4.1.1 and the RPL-BIER Storing Mode which is discussed in Section 4.1.2. A new

Control Message Options (CMO) is introduced to transport the bitStrings in Section 4.2.

4.1. RPL-BIER MOPs

In RPL-BIER modes of operation, one or more RPL Target Option are replaced by a new BitString Information Option which represent the advertised target(s) by a combination of a bitString and control information.

4.1.1. RPL-BIER Non-Storing Mode

In Non-Storing RPL-BIER, a bit in classical BIER bit-by-bit bitStrings (see Section 6.1) or a set of bits in a Bloom Filter (see Section 6.2) is associated to a next-hop from the perspective of an intermediate router. RPL Non-Storing Mode DAO messages are used to advertise the relation between a target and its parent (transit) directly to the root.

If multiple Targets Options were to be placed consecutively to factorize a Transit Information Option (TIO) in a classical RPL Non-Storing Mode DAO message, they are replaced by a single BIO with the aggregated bitString that represents all these targets.

4.1.2. RPL-BIER Storing Mode

In RPL-BIER Storing Mode, a bit in classical BIER Bit-by-bit bitStrings (see Section 6.1) or a set of bits in a Bloom Filter (see Section 6.2) is associated to a final destination. RPL Storing Mode DAO messages are used to advertise recursively the targets to the parent(s) all the way to the root.

The BitString Information Option(s) in the DAO message contain collectively an aggregated bitString that represents the advertising node and all of its descendants. Parents save the bitString per child, and use it to forward down the DODAG as discussed in Section 6.1.3.

The Transit Information Option is not used. The lack of transit information is compensated by a more frequent transmission of DAO messages and a full use of the RPL data plane loop avoidance and inconsistency detection mechanisms (section 11.2 of [RFC6550]), in collaboration with a periodic 6LoWPAN ND (re)registration that maintains the 6LBR and the root aware of which devices are actually present in the network with the associated lifetime and sequence information.

4.2. BitString Information

Section 6.7 of [RFC6550] specifies optional CMOs 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 optional fields; 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 a new CMO, the BitString Information option (BIO). A BIO is used as an alternate to one or more Target Option(s) in Storing and Non-Storing Mode DAOs using bit-by-bit bitStrings, and its format is as follows:

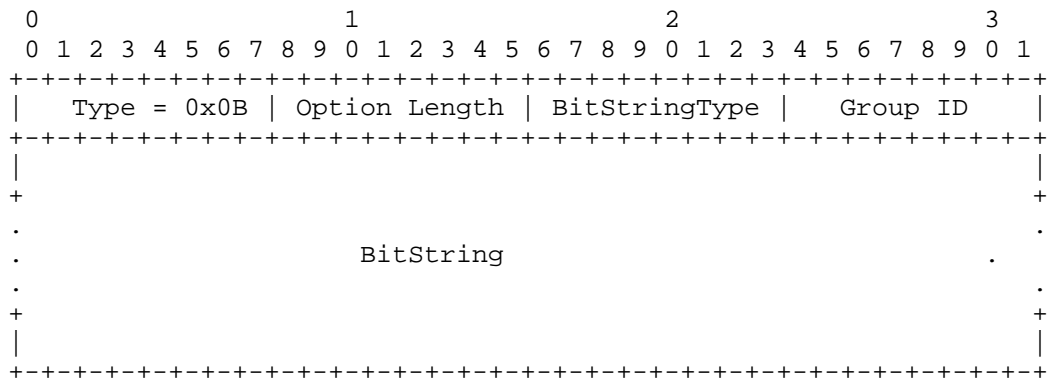


Figure 1: BitString Information option format

- Option Type: 0x0B (to be confirmed by IANA)
- Option Length: In bytes; variable, depending on the Type.
- BitString Type: Indicates the size of the bitString field as indicated in Table 1 below.

BitString Type	BitString Size
15	8 bits
16	16 bits
17	48 bits
18	96 bits
19	160 bits

Group ID : 8-bit unsigned integer. The Group ID for the bit-by-bit bitString.

BitString: 8 to 160 bits, depending on the Type.

5. Updating the 6LoWPAN Framework

5.1. Extensions to RFC 6775

It is noted that RPL does not provide a Duplicate Address Detection (DAD) and relies on 6LoWPAN ND to ensure that addresses are unique within the network. For that purpose, a 6LoWPAN Border Router (6LBR) maintains the list of addresses that are currently in use in the network that it serves. In the case of a RPL LLN, the 6LBR is typically collocated with the RPL root, and serves the RPL DODAG. With 6LoWPAN ND[RFC6775] [I-D.ietf-6lo-rfc6775-update], a Duplicate Address Request (DAR) / Duplicate Address Confirmation (DAC) exchange is used to perform the DAD operation. Scalability is achieved by federating multiple DODAGs with IPv6 Backbone Routers (6BBRs) [I-D.ietf-6lo-backbone-router].

In that context, it makes sense to also leverage the 6LBR to ensure that a tuple (groupID, bitString) is assigned unequivocally to an IPv6 address for the bit-by-bit operation. This specification extends the role of a 6LBR to 1) assign the tuple to the IPv6 address and 2) resolve an IPv6 address into a tuple. To achieve this, RFC 6775 is updated as follows:

- o A BIER Address Resolution (BAR) / BIER Address Confirmation (BAC) exchange is introduced for the purpose of the bitString lookup operation (see Section 6.1.1).
- o A new Bit Position Option (BPO) is introduced to carry the corresponding bit position bitString in 6LoWPAN ND exchanges. The BPO is transported in BAC, NA and DAC messages in response to BAR, NS and DAR messages, respectively (see Section 5.3.1).

5.2. Extensions to RFC 6282

This specification also extends the 6LoWPAN framework with the capability to transform an address into a tuple (Control field, bitString) as part of the 6LoWPAN Header Compression [RFC6282] (6LoWPAN HC). Since the 6LBR and the Header Compression functions are typically collocated, the latter may exploit local tables built by the former to map a destination IPv6 address into a bitString.

In Storing Mode, P2P stretched routing via a common parent can be obtained if the destination is expressed as a tuple (Control field, bitString). This can be achieved with a BAR/BAC exchange with the 6LBR.

5.3. New Neighbor Discovery Options and Messages

In order to allocate and lookup a bitString, this specification extends 6LoWPAN ND with the following new messages and formats.

5.3.1. 6LoWPAN ND Bit Position Option

The Bit Position Option (BPO) is intended to be used to return a bitString and related information in 6LoWPAN ND BA, DAC and BAC messages with a Status of 0 indicating success, the NA and DAC messages transporting an Address Registration Option (ARO) indicating the IPv6 address that is mapped with the bit position. Its format is as follows:

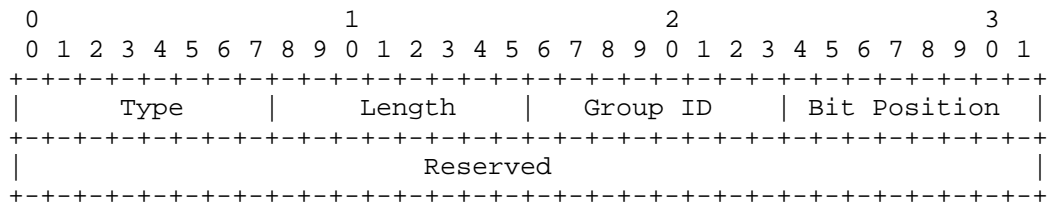


Figure 2: Bit Position Option format

Option Fields

- Type: 38 (to be confirmed by IANA)
- Length: 8-bit unsigned integer. The length of the option (including the type and length fields) in units of 8 bytes. The Length MUST be set to 1.
- Group ID: 8-bit unsigned integer. The GroupID for the bit-by-bit bitString.
- Bit Position: 8-bit unsigned integer. The offset in the the bit-by-bit bitString.

5.3.2. Address Mapping Message

For the multihop lookup exchanges between a 6LR that needs to perform a Header Compression including the bitString for the destination, and its 6LBR, which knows if the mapping exists and what it is, this specification introduces two new ICMPv6 message called the BIER Address Resolution (BAR) and the BIER Address Confirmation (BAC). We avoid reusing the NS and NA messages for this purpose, since these messages are not subject to the hop limit=255 check as they are forwarded by intermediate 6LRs.

The BAR and BAC use the same ICMPv6 type value which this specification, allocates for a generic Address Mapping service, but use different Codes. This is done to save addressable space in the ICMPv6 type values which is getting crowded, and because it is expected that in the future, other mapping techniques may be needed as well.

The Status field and Information Lifetime are not meaningful in the BAR message. When and only when the BAC message carries a status of 0, indicating success, the Information Lifetime must contain valid information and the message must carry a Bit Position Option (Section 5.3.1).

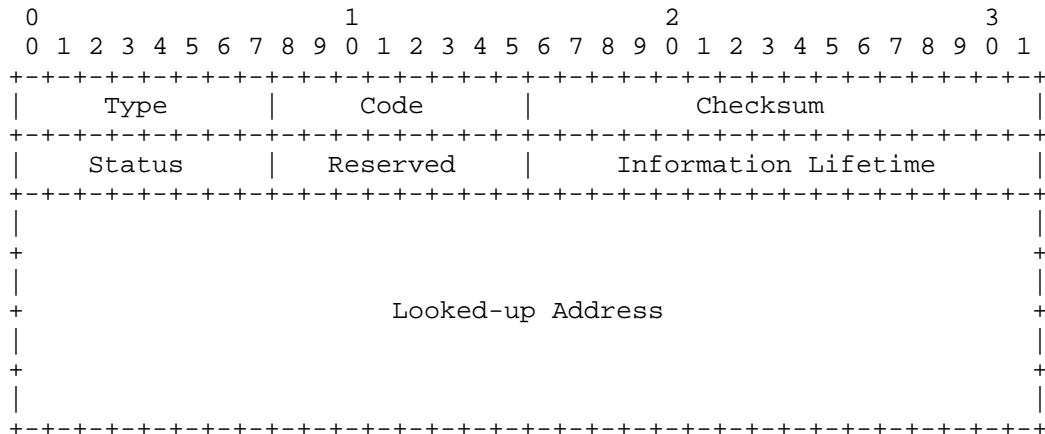


Figure 3: Address Mapping Message format

Message Fields

- Type: 160 (to be confirmed by IANA)
- Code: 8-bit unsigned integer. 1 for BAR and 2 for BAC. Other values are reserved.
- Checksum:: The ICMP checksum. See [RFC4443].

Status: 8-bit unsigned integer. Indicates the status of the lookup in the BAC. See Table 2 below.

Status	Description
0	Success
1	Looked-up Address not Found
2..255	Allocated using Standards Action [RFC8126]

Information Lifetime: 16-bit unsigned integer. The length of time in units of 60 seconds (relative to the time the packet is received) that this mapping information is valid. A value of all zero bits (0x0) assumes a default value of 10,000 (~one week).

Looked-up Address: 128-bit field. Carries the IPv6 address that is being mapped.

6. BitString formats

This specification introduces two BitString formats, the bit-by-bit and the Bloom Filter.

6.1. Bit-by-bit BitStrings

In the bit-by-bit case, each bit is mapped in an unequivocal fashion with a single addressable resource in the network. In RPL-BIER Storing Mode, this is an IPv6 address as advertised in RPL Storing Mode DAO messages, whereas in RPL-BIER Non-Storing Mode, this is a parent-child relationship as advertised in RPL Non-Storing Mode DAO messages.

if every node in a large network is given one or more bits in a bit-by-bit bitString, then the bitString may grow very large and lead to undesirably large overhead in the data plane. BIER allows to divide a potentially the very large abstract bitString into segments, aka groups, indexed by a groupID.

In the protocol elements that use a bitString, only the relevant group(s) are transported, and the advertised bitString is in fact the segment that corresponds to the groupID. It results that a bit position in the large abstract bitString is advertised either as the tuple (groupID, segment) or the tuple (groupID, bit position within segment).

For simplicity, when dealing with protocol elements, the specification uses the term `bitString` to refer to the tuple (`groupID`, `segment`) and a bitwise operation between `bitStrings` is really a bitwise operation between segments of a same `groupID`.

TBD: do we allow multiple (`groupID`, `bitString`) tuples in one packet?

6.1.1.1. Allocating a Bit Position in a Bit-by-bit BitString

Several methods may be used to associate a bit in a `bitString` to an IPv6 address. In order to guarantee interoperability, this specification RECOMMENDS that all implementations provide at least the method described therein.

With 6LoWPAN ND, a 6LoWPAN Node (6LN) registers with address(es) to one or more 6LoWPAN Router [RFC6775] to perform Duplicate Address Detection (DAD). As part of the DAD process, the 6LN validates that a Global Unicast Address (GUA) or a Unique Local Address (ULA) is effectively unique using a unicast DAR/DAC exchange with the 6LBR (this procedure is updated in [I-D.ietf-6lo-rfc6775-update]).

In a network that supports this specification, the 6LBR maintains a configurable number of groups (up to 32, indexed by `groupID`), and for each group, it maintains a pool of free bit positions. Upon a new registration, the 6LBR selects a `groupID` and a free bit position and associates it to the IPv6 address.

The `bitString Size` in any given group should be configurable. The policy for selecting the `groupID` for a new registration is left to implementation. It is noted that in large networks that require multiple groups, associating groups with immediate children of the root may be an option to limit the number of groups that the RPL routers must be aware of.

If the 6LBR accepts the registration, then it returns a DAC message with a status of 0 indicating success, adding a 6LoWPAN ND Bit Position Option (Section 5.3.1) to the DAC message to indicate the `groupID` and bit.

The 6LR maintains a binding cache entry (BCE) for the 6LN based on successful DAC messages. With this specification, the 6LR also stores the matching between the address and the `bitString` and uses it for searching its children when forwarding packets in Non-Storing Mode (see Section 6.1.3).

If the 6LN child does not support the BIER encoding (e.g.[I-D.thubert-6lo-bier-dispatch]), then the packet is converted in a format that the child supports (e.g.[RFC8138]).

6.1.2. Aggregation of Bit-by-bit BitStrings

BitStrings are aggregated by a 'OR' operation so that all the bits that are set in either bitString is set in the resulting bitString. In the concise form of a tuple (groupID, bitString), the aggregation is done on a group-by-group basis, between segments of a same group.

In RPL-BIER Storing Mode, the bit-by-bit BitStrings are passed from child to parent using DAO messages, in a fashion similar to RPL Storing Mode [RFC6550]. The BitString Information option (Figure 1) is used in replacement of the Target option. A DAO message contains one BIO per group, and the parent that receives the messages associates the BIO information to the advertising child. In order to build a DAO message, the parent regenerates its own BIO, one per group, by aggregating the bitStrings from all of its children with its own, and places the resulting BIOs in the DAO message.

6.1.3. Forwarding Based on Bit-by-bit BitStrings

Forwarding is based on matching a bitString in a packet with those of children. For unicast packets, only one matching child gets the packet. For multicast packets, all matching children get a copy. Matches are found by scanning all children and performing bitwise operations as follows.

In order to search for a match, a reference bitString is initialized with the destination bitString in the packet. A match is found with a child if the bitwise 'AND' between the reference bitString and the bitString stored for that child does not result in a NULL bitString of all zeroes.

In Non-Storing Mode, a packet is copied to all matching children, which are found by trying all children.

In Non-Storing Mode, if a child is selected for forwarding, then an 'XOR' operation is performed between the reference bitString and the bitString resulting from the 'AND' operation. If the 'XOR' operation does not result in a NULL bitString, denoting that more children should get the packet, then the result of the 'XOR' operation becomes the new reference bitString and the search continues. The 'XOR' operation allows to stop the search loop as soon as all matches are found; it also avoids forwarding twice to a same destination along different downwards path in the DODAG.

6.1.4. Reliable Multicast based on Bit-by-bit BitStrings

Multicast from the root to a collection of target 6LNs can be made reliable with the following operation:

A multicast packet is identified by a unique packetID which is also found in the acknowledgments. The root signals the set of targets with a destination bitString that has the bits set for each of them, and the message is forwarded as described on Section 6.1.3.

Listeners acknowledge with an acknowledgment packet that contains the same information, the packetID and the bitString representing the listener. The bitStrings in acknowledgment packets are aggregated recursively on the way back as indicated in Section 6.1.2.

The root aggregates the bitStrings from its children into one acknowledgment bitString. It then checks that the acknowledgment bitString is correct, by an 'AND' operation with the destination bitString that should result in the acknowledgment bitString. If this is not the case, bits that are set in the acknowledgment bitString and not in the destination bitString are in the acknowledgment bitString.

The root generates the bitString of the devices that did not acknowledge the multicast message by a bitwise 'XOR' operation between the destination bitString and the acknowledgment bitString, and may use it to selectively retry the multicast.

6.2. Bloom Filters

A Bloom Filter can be seen as an additional compression technique for the bitString representation. A Bloom Filter may generate false positives, which, in the case of BIER, result in undue forwarding of a packet down a path where no listener exists.

As an example, the Constrained-Cast [I-D.bergmann-bier-ccast] specification employs Bloom Filters as a compact representation of a match or non-match for elements in a set that may be larger than the number of bits in the BitString.

In the case of a Bloom Filter, a number of Hash functions must be run to obtain a multi-bit signature of an encoded element. This specification uses the 5-bits Control field to signal an Identifier of the set of Hash functions being used to generate a certain bitString, so as to enable the migration from a set of Hash functions to the next.

6.2.1. Computing and Saving Bloom Filters

6.2.2. Forwarding based on Bloom Filters

6.2.3. Hash Functions Distribution

7. Implementation Status

TBD

8. Security Considerations

TBD

9. IANA Considerations

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

Code	Description	Reference
0x0B	bitString	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
6	RPL-BIER Non-Storing Mode of operation	This document
7	RPL-BIER Storing Mode of operation	This document

DIO Mode of operation

10. Acknowledgments

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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.

11.2. Informative References

- [I-D.bergmann-bier-ccast] Bergmann, O., Bormann, C., Gerdes, S., and H. Chen, "Constrained-Cast: Source-Routed Multicast for RPL", draft-bergmann-bier-ccast-02 (work in progress), October 2016.

- [I-D.eckert-bier-te-arch]
Eckert, T., Cauchie, G., Braun, W., and M. Menth, "Traffic Engineering for Bit Index Explicit Replication BIER-TE", draft-eckert-bier-te-arch-06 (work in progress), November 2017.
- [I-D.ietf-6lo-backbone-router]
Thubert, P., "IPv6 Backbone Router", draft-ietf-6lo-backbone-router-06 (work in progress), February 2018.
- [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-21 (work in progress), June 2018.
- [I-D.ietf-roll-aodv-rpl]
Anamalamudi, S., Zhang, M., Sangi, A., Perkins, C., Anand, S., and B. Liu, "Asymmetric AODV-P2P-RPL in Low-Power and Lossy Networks (LLNs)", draft-ietf-roll-aodv-rpl-04 (work in progress), July 2018.
- [I-D.thubert-6lo-bier-dispatch]
Thubert, P., Brodard, Z., Jiang, H., and G. Texier, "A 6LoRH for BitStrings", draft-thubert-6lo-bier-dispatch-04 (work in progress), January 2018.
- [RFC4944] 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>>.
- [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>>.
- [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>>.

- [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>>.
- [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>>.
- [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>>.

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, 6775 (if approved)
Intended status: Standards Track
Expires: September 19, 2018

P. Thubert, Ed.
Cisco
March 18, 2018

Routing for RPL Leaves
draft-thubert-roll-unaware-leaves-04

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 September 19, 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	4
3. Updating RFC 6550	5
4. Updating RFC 6775 Update	5
5. Dependencies on 6LN	5
6. Protocol Operations	6
6.1. General Flow	6
6.2. 6LN Operation	8
6.3. 6LR Operation	9
6.4. RPL Root Operation	10
6.5. 6LBR Operation	11
7. Implementation Status	11
8. Security Considerations	11
9. IANA Considerations	11
10. Acknowledgments	12
11. References	12
11.1. Normative References	12
11.2. Informative References	13
Appendix A. Subset of a 6LoWPAN Glossary	13
Author's Address	14

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. A RPL-Aware Node 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.

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.

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.

"Registration Extensions for 6LoWPAN Neighbor Discovery" [I-D.ietf-6lo-rfc6775-update] defines an Extended ARO (EARO) with a '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. The DAR and DAC messages are also extended as EDAR and EDAC messages respectively.

A RPL-Unaware Leaf (RUL) sets the 'R' flag in the EARO to declare itself as a host with the expectation that the 6LR that accepts the registration injects routing information for the Registered Address in the RPL domain. 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

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].

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 Appendix A.

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.

3. 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.

4. 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 RPL 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.

5. 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.

6. Protocol Operations

6.1. 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 1, 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.

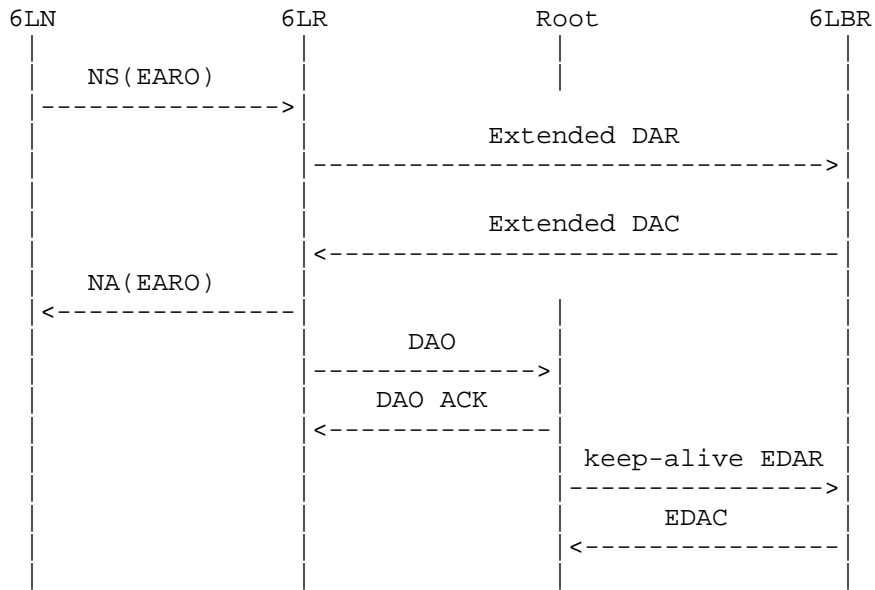


Figure 1: First Registration Flow

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 1, 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.

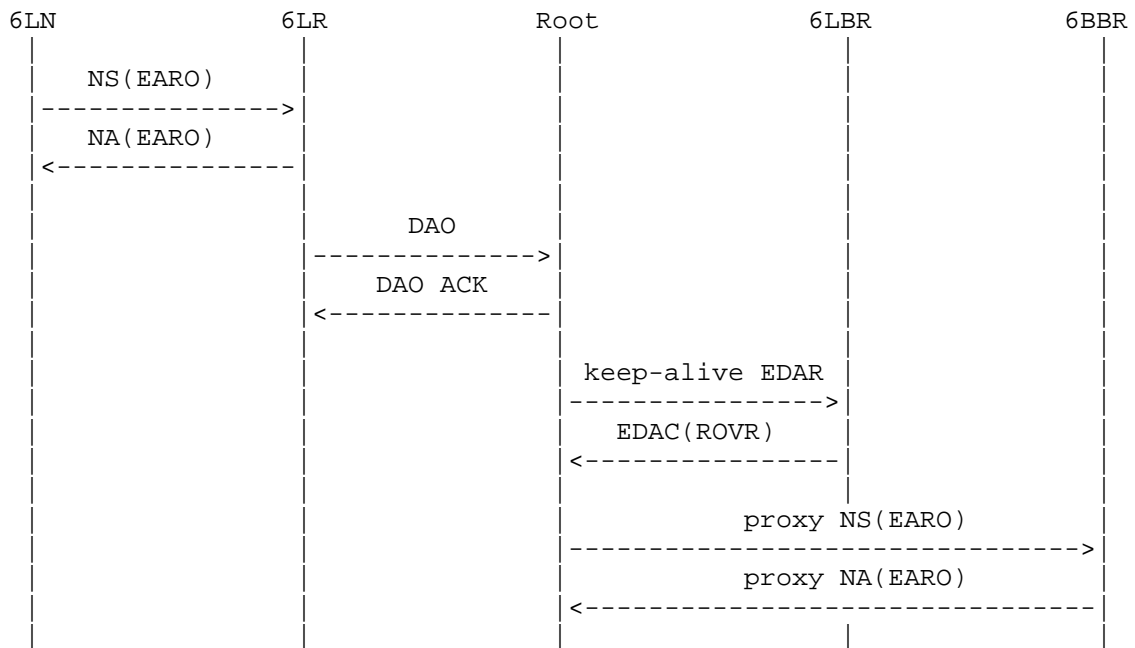


Figure 2: Next Registration Flow

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.

6.2. 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 increases the TID field.
- 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.

6.3. 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.

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.

6.4. 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-npdoa].

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.

6.5. 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.

7. Implementation Status

8. 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.

9. IANA Considerations

This specification has no requirement on IANA.

10. Acknowledgments

The author wishes to thank Michael Richardson and Georgios Papadopoulos for their early reviews of and contributions to this document

11. References

11.1. 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-15 (work in progress), March 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>>.
- [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>>.
- [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>>.
- [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>>.

11.2. 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., Sahoo, R., and Z. Cao, "No-Path DAO modifications", draft-ietf-roll-efficient-npdao-01 (work in progress), October 2017.
- [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-22 (work in progress), March 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>>.

Appendix A. 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
DAD: Duplicate Address Detection
LLN: Low Power Lossy Network (a typical IoT network)
NA: Neighbor Advertisement
NCE: Neighbor Cache Entry
ND: Neighbor Discovery
NDP: Neighbor Discovery Protocol
NS: Neighbor Solicitation
RUID: Registration Unique ID
TSCH: TimeSlotted Channel Hopping
TID: Transaction ID (a sequence counter in the EARO)

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:

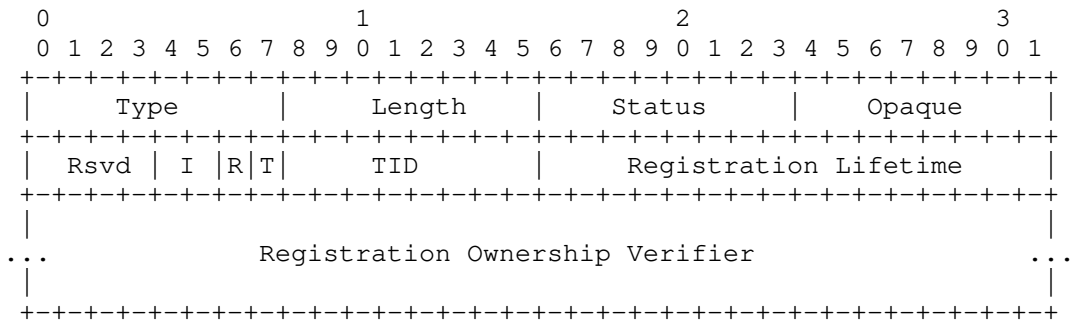


Figure 1: EARO Option Format

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.

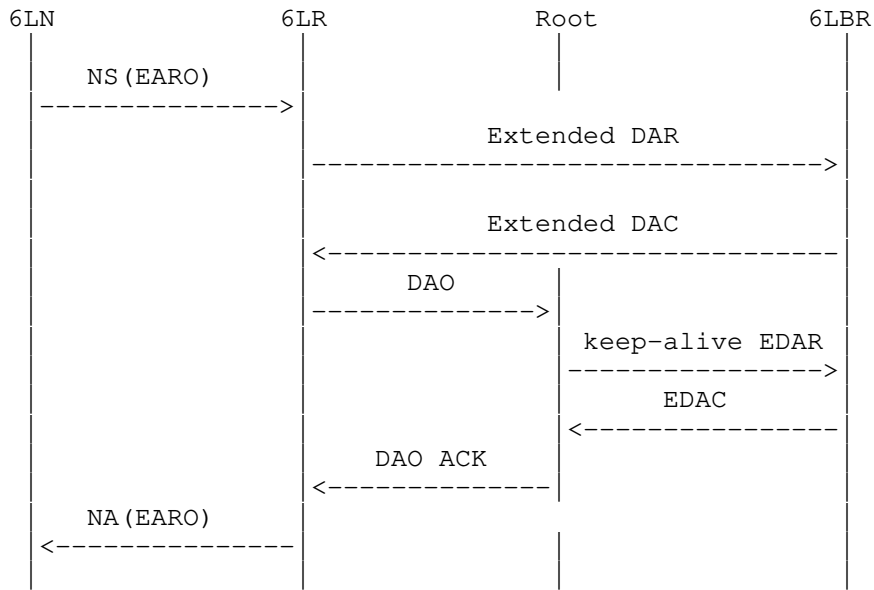


Figure 2: First Registration Flow

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.

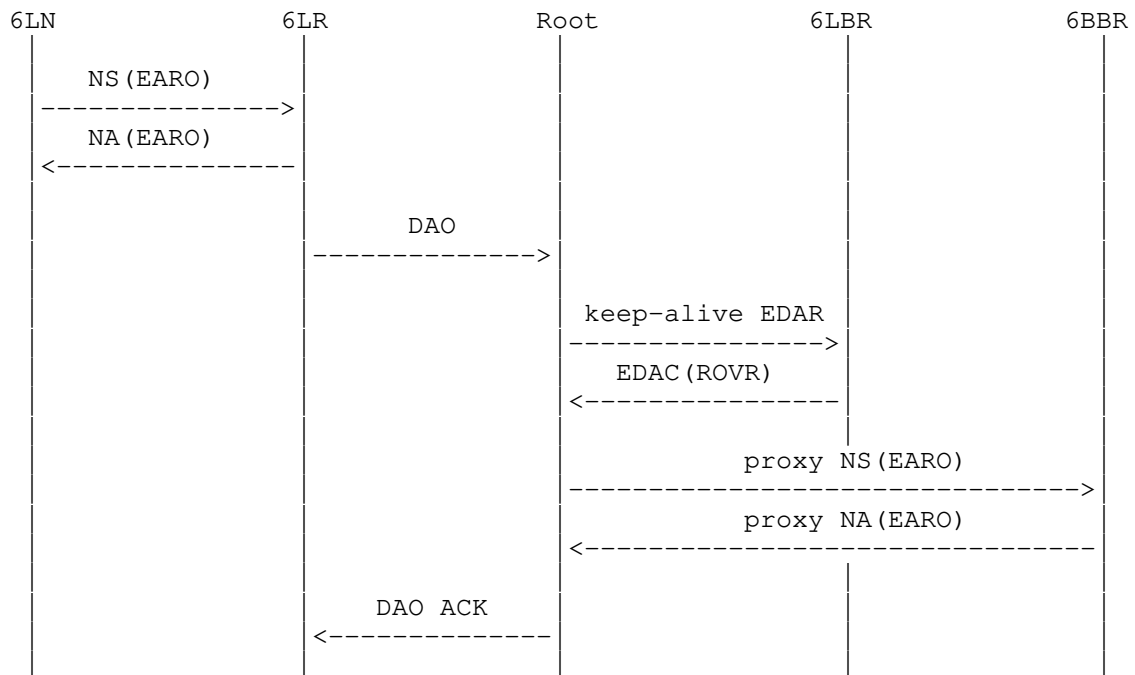


Figure 3: Next Registration Flow

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 EDAR, 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.

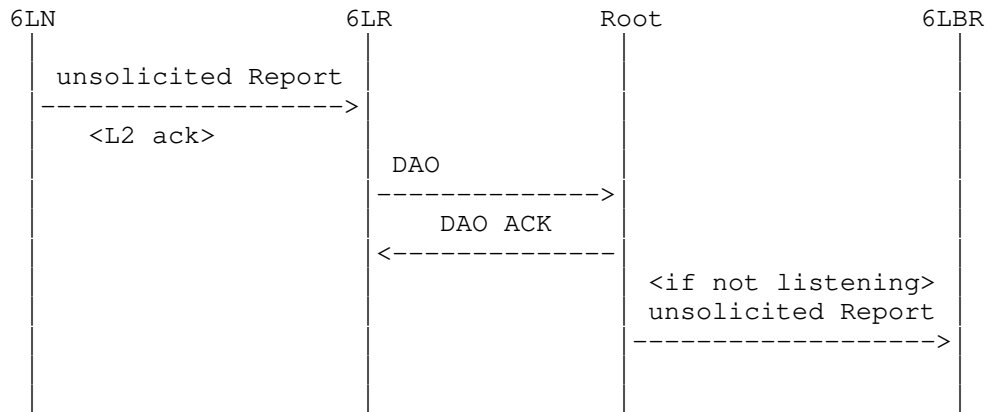


Figure 4: First Multicast Registration Flow

A re-registration is pulled by 6LR acting as querier. Note that the message may 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,

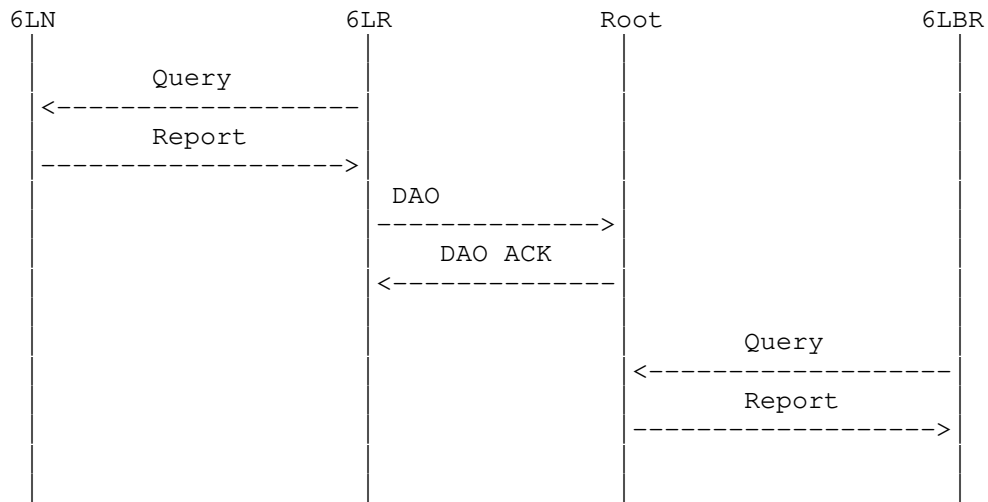


Figure 5: Next Registration Flow

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