

ROLL  
Internet-Draft  
Updates: [6550](#) (if approved)  
Intended status: Standards Track  
Expires: 5 April 2021

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

Root initiated routing state in RPL  
draft-ietf-roll-dao-projection-14

## Abstract

This document updates [RFC 6550](#) 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 5 April 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

Internet-Draft

DAO Projection

October 2020

and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the [Trust Legal Provisions](#) and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction</a>	<a href="#">3</a>
<a href="#">2.</a>	<a href="#">Terminology</a>	<a href="#">5</a>
<a href="#">2.1.</a>	<a href="#">Requirements Language</a>	<a href="#">5</a>
<a href="#">2.2.</a>	<a href="#">Glossary</a>	<a href="#">5</a>
<a href="#">2.3.</a>	<a href="#">Other Terms</a>	<a href="#">5</a>
<a href="#">2.4.</a>	<a href="#">References</a>	<a href="#">6</a>
<a href="#">3.</a>	<a href="#">Updating <a href="#">RFC 6550</a></a>	<a href="#">6</a>
<a href="#">4.</a>	<a href="#">New RPL Control Messages and Options</a>	<a href="#">8</a>
<a href="#">4.1.</a>	<a href="#">New P-DAO Request Control Message</a>	<a href="#">8</a>
<a href="#">4.2.</a>	<a href="#">New PDR-ACK Control Message</a>	<a href="#">9</a>
<a href="#">4.3.</a>	<a href="#">Route Projection Options</a>	<a href="#">10</a>
<a href="#">4.4.</a>	<a href="#">Sibling Information Option</a>	<a href="#">12</a>
<a href="#">5.</a>	<a href="#">Projected DAO</a>	<a href="#">14</a>
<a href="#">5.1.</a>	<a href="#">Requesting a Track</a>	<a href="#">15</a>
<a href="#">5.2.</a>	<a href="#">Identifying a Track</a>	<a href="#">16</a>
<a href="#">5.3.</a>	<a href="#">Forwarding Along a Track</a>	<a href="#">17</a>
<a href="#">5.4.</a>	<a href="#">Non-Storing Mode Projected Route</a>	<a href="#">18</a>
<a href="#">5.5.</a>	<a href="#">Storing Mode Projected Route</a>	<a href="#">19</a>
<a href="#">6.</a>	<a href="#">Security Considerations</a>	<a href="#">21</a>
<a href="#">7.</a>	<a href="#">IANA Considerations</a>	<a href="#">21</a>
<a href="#">7.1.</a>	<a href="#">New RPL Control Codes</a>	<a href="#">21</a>
<a href="#">7.2.</a>	<a href="#">New RPL Control Message Options</a>	<a href="#">22</a>
<a href="#">7.3.</a>	<a href="#">SubRegistry for the Projected DAO Request Flags</a>	<a href="#">22</a>
<a href="#">7.4.</a>	<a href="#">SubRegistry for the PDR-ACK Flags</a>	<a href="#">23</a>
<a href="#">7.5.</a>	<a href="#">Subregistry for the PDR-ACK Acceptance Status Values</a>	<a href="#">23</a>
<a href="#">7.6.</a>	<a href="#">Subregistry for the PDR-ACK Rejection Status Values</a>	<a href="#">23</a>
<a href="#">7.7.</a>	<a href="#">SubRegistry for the Route Projection Options Flags</a>	<a href="#">24</a>
<a href="#">7.8.</a>	<a href="#">SubRegistry for the Sibling Information Option Flags</a>	<a href="#">24</a>
<a href="#">7.9.</a>	<a href="#">Error in Projected Route ICMPv6 Code</a>	<a href="#">25</a>
<a href="#">8.</a>	<a href="#">Acknowledgments</a>	<a href="#">25</a>
<a href="#">9.</a>	<a href="#">Normative References</a>	<a href="#">25</a>
<a href="#">10.</a>	<a href="#">Informative References</a>	<a href="#">26</a>
<a href="#">Appendix A.</a>	<a href="#">Applications</a>	<a href="#">27</a>
<a href="#">A.1.</a>	<a href="#">Loose Source Routing</a>	<a href="#">27</a>
<a href="#">A.2.</a>	<a href="#">Transversal Routes</a>	<a href="#">29</a>
	<a href="#">Authors' Addresses</a>	<a href="#">31</a>

## 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. A node at the ingress of more than one Segment in a Track may use any combination of those Segments to forward a packet towards the Track Egress.

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  
RPO: A Route Projection Option; it can be a VIO or an SRVIO.  
RTO: RPL Target Option  
RUL: RPL-Unaware Leaf  
SIO: RPL Sibling Information Option  
SRVIO: A Source-Routed Via Information Option, used in Non-Storing Mode P-DAO messages.  
SMPR: Storing Mode Projected Route  
TIO: RPL Transit Information Option  
VIO: A Via Information Option, used in Storing Mode P-DAO messages.

### [2.3.](#) Other Terms

Projected Route: A Projected Route is a path segment that is computed remotely, and installed and maintained by a RPL Root.  
Projected DAO: A DAO message used to install a Projected Route.  
Track: A complex path with redundant Segments to a destination.  
TrackID: A RPL Local InstanceID with the 'D' bit set. The TrackID is associated with a IPv6 Address to the Track Egress Node.

### [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.](#) Updating [RFC 6550](#)

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 one or more Projected Route(s) using the new CMOs presented therein.

A Projected Route can be an additional route of higher precedence within the main DODAG. In that case, it is installed with a P-DAO using the parameters of the main DODAG, typically a global RPLInstanceID and the DODAGID field elided as shown in Section 6.4.1. of [RPL].

A Projected Route can also be a Segment within a Track. A stand-alone Segment can be used as a Serial Track. Segments can also be combined to form a Complex Track. The Root uses a local RPL Instance rooted at the Track Egress to signal the Track. The local RPLInstanceID of the Track is called the TrackID, more in Section 5.2. A P-DAO message for a Track signals the IPv6 Address of the Track Egress in the DODAGID field of the DAO Base Object, and the TrackID in the RPLInstanceID field, as shown in Figure 1.

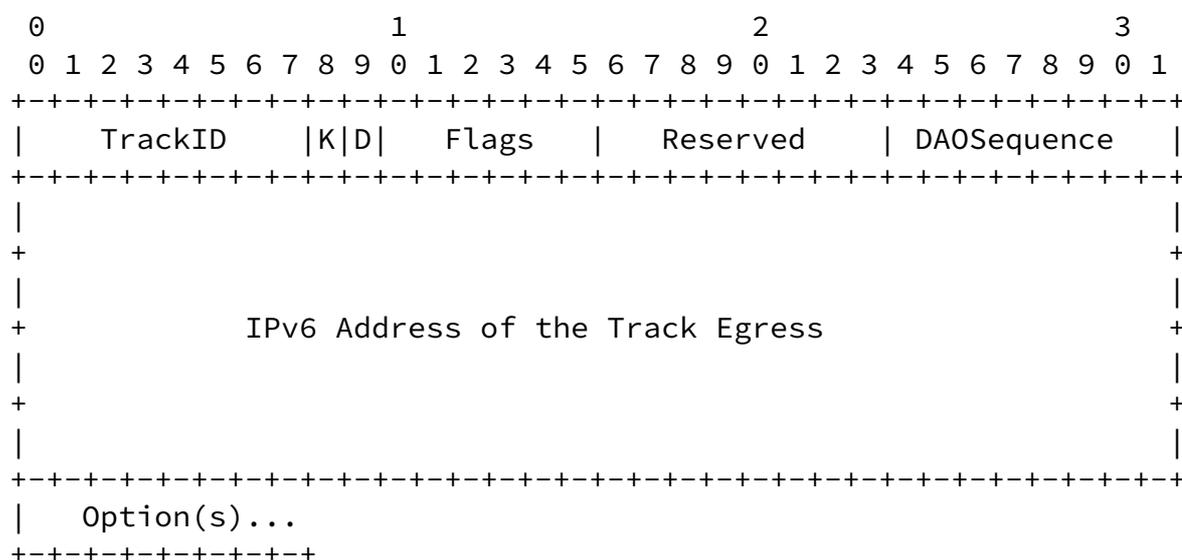


Figure 1: Projected DAO Format for a Track

In RPL Non-Storing Mode, the TIO and RT0 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 RT0s 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 RT0s. 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 Route Projection Options (RPO) are introduced for use in P-DAO messages as a multihop alternative to the TIO. One RPO is the Via Information Option (VIO); the VIO installs a state at each hop along a Storing Mode Projected Route (SMPR). The other is the Source-Routed VIO (SRVIO); the SRVIO installs a source-routing state at the Segment ingress, which uses that state to encapsulate a packet with a Routing Header (RH) along a Non-Storing Mode Projected Route (NMPR).

Like in a DAO message, the RTOs can be factorized in a P-DAO, but the RPOs cannot. A P-DAO contains one or more RTOs that indicate the destinations that can be reached via the Track, and exactly one RPO that signals the sequence of nodes between the Track Ingress and the Track Egress, both included. 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 at each hop. In both cases the Track Ingress generates the P-DAO-ACK when the installation is successful.

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 4.4](#). The sibling selection process is out of scope.

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 4.1](#) 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.

#### 4.1. 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 5.1](#).

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

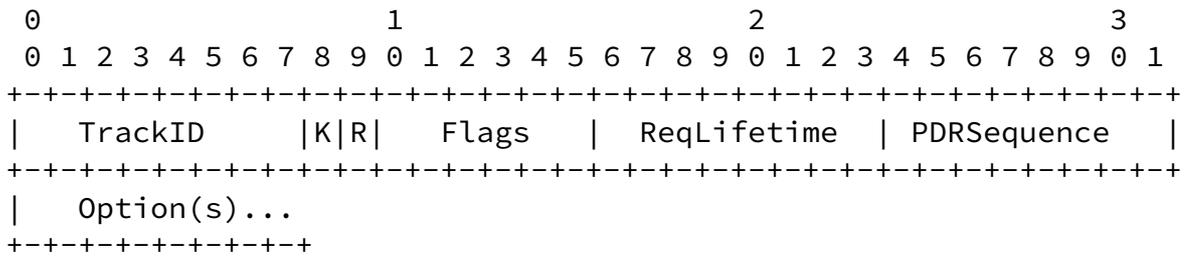


Figure 2: New P-DAO Request Format

**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 be initialized to zero by the sender and MUST be ignored by the receiver.

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

A PDR with a fresher PDRSequence refreshes the lifetime, and a PDRLifetime of 0 indicates that the track should be destroyed.

**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.

4.2. 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:

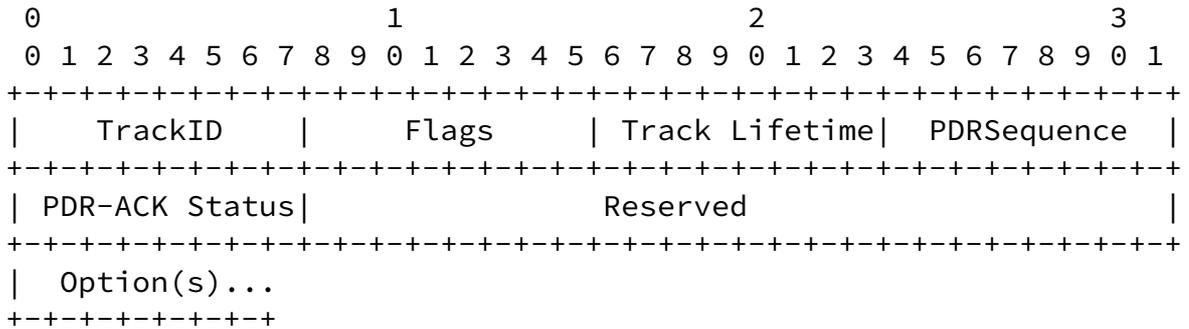


Figure 3: New PDR-ACK Control Message Format

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 4:

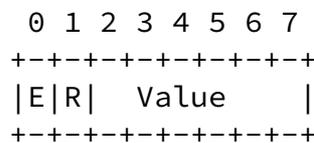


Figure 4: PDR-ACK status Format

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 4 and Table 5.

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

### 4.3. Route Projection Options

An RPO 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 RPO MUST contain at least one Via Address, and a Via Address MUST NOT be present more than once, otherwise the RPO MUST be ignored. The format of the RPOs is as follows:

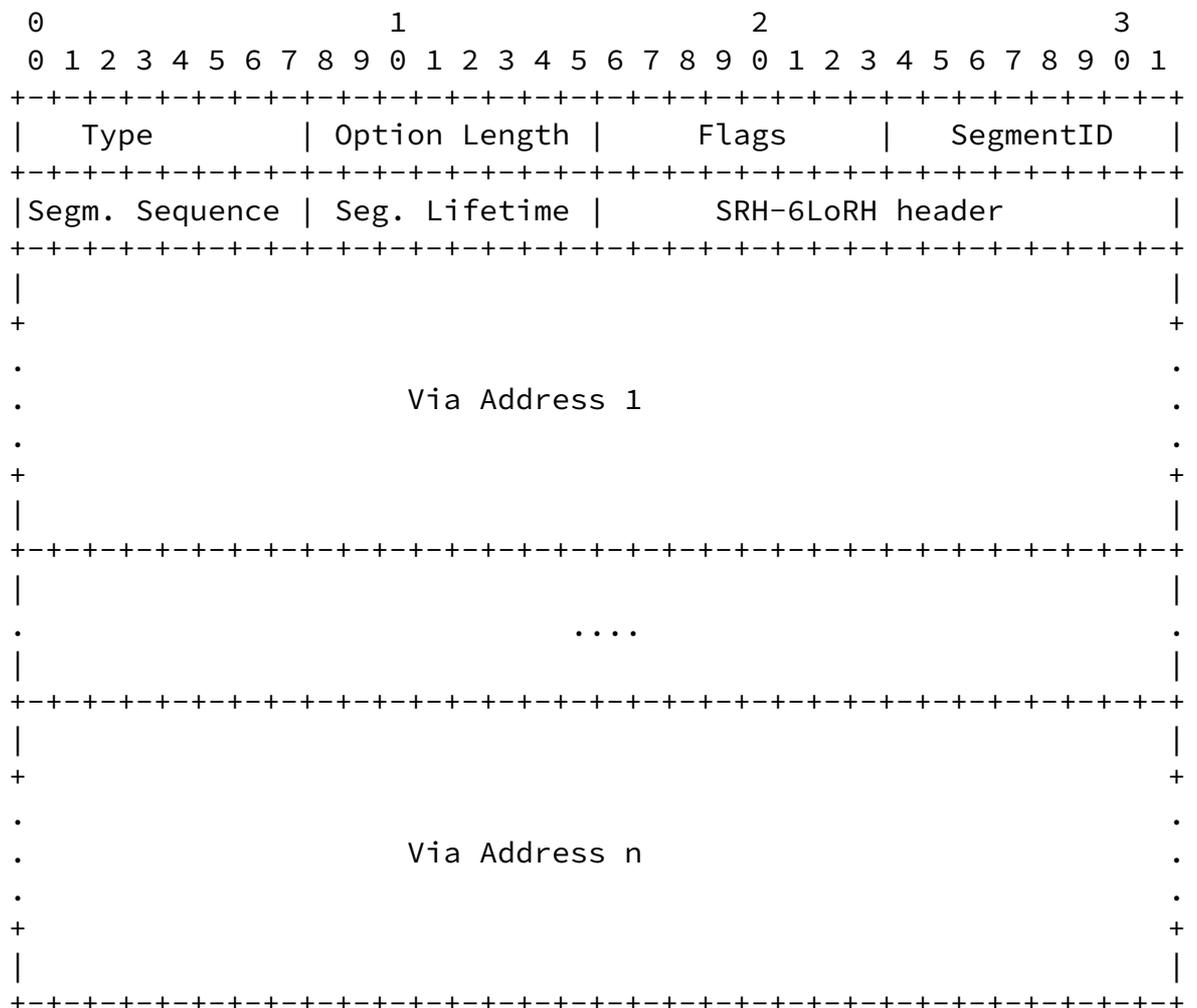


Figure 5: Route Projection Option format (uncompressed form)

Option Type: 0x0B for VIO, 0x0C for SRVIO (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 RPO deprecates any state for the Segment indicated by the SegmentID within the indicated Track and sets up the new information.

An RPO with a Segment Sequence that is not as fresh as the current one is ignored.

An RPO for a given Track Egress 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 for a Track Egress is referred as a No-



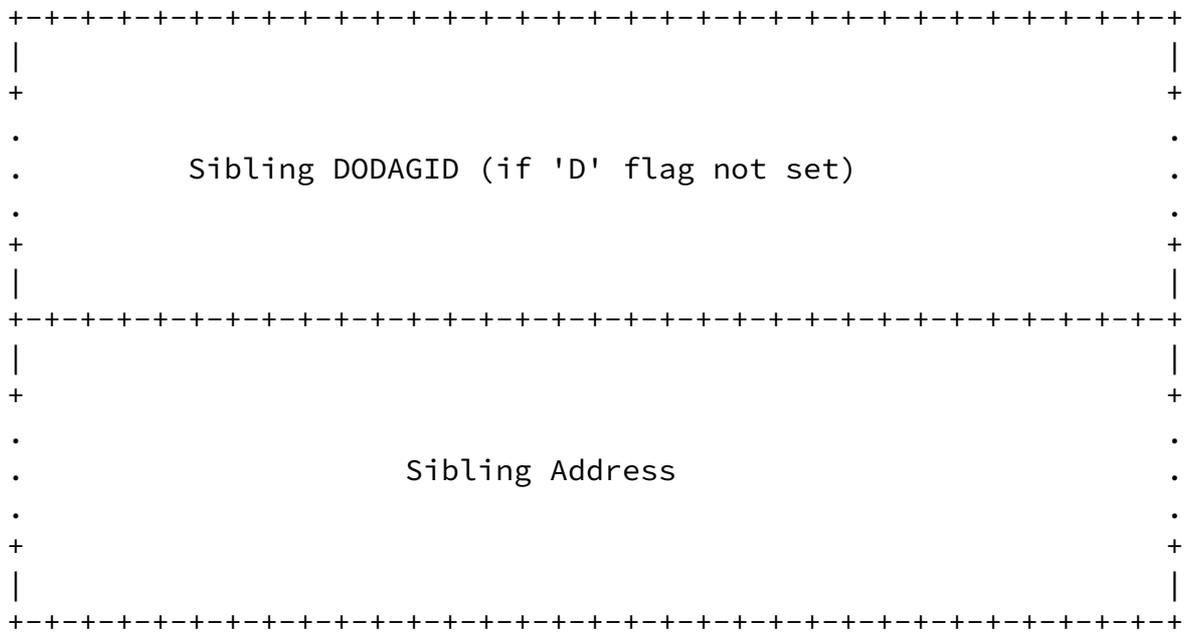


Figure 6: Sibling Information Option Format

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 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 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.

## [5.](#) Projected DAO

This draft adds a capability to RPL whereby the Root of a DODAG projects a Track by sending one or more Projected-DAO (P-DAO) messages to selected routers in the DODAG. 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 Route Projection Option (RPO). A P-DAO message MUST contain exactly one RPO, which is either a VIO or an SRVIO, and MUST follow one or more RTOs. There can be at most one such sequence of RTO(s) and an RPO.

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 RPO as opposed to the Path Sequence from a TIO. Also, a Segment Lifetime of 0 in an RPO 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 5.4](#). A Non-Storing Mode P-DAO carries an SRVIO 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 ingress router of the source-routed Segment. The ingress router 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 5.5](#). A Storing Mode P-DAO carries a 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 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 Segment, that it can also be neither but it cannot be both.

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 7.9](#)). 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 5.4](#) and [Section 5.5](#) respectively.

### [5.1](#). 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 Track Egress 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 resrequesting 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 resrequesting 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 resrequesting 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.

## [5.2.](#) 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 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 Egress that serves as DODAGID for the Track. This way, a Track is uniquely identified by the tuple (Track Egress Address, TrackID) where the TrackID is always represented with the 'D' flag set.

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

### [5.3.](#) Forwarding Along a Track

Sending a Packet within a RPL Local Instance requires the presence of an RPL Packet Information (RPI) (see [\[USEofRPLinfo\]](#)) in the outer IPv6 Header chain. The RPI carries a local RPLInstanceID which, in association with the IPv6 final destination, indicates the RPL Instance that the packet follows.

This draft leverages the RPL Forwarding model follows:

- \* The RPI carries a local RPLInstanceID called the TrackID, which, in association with the IPv6 final destination, indicates the Track along which the packet is forwarded. The 'D' flag in the RPLInstanceID MUST be set to indicate that the final destination address in the IPv6 header owns the local RPLInstanceID, more in [Section 5.3.](#)

In the data packets, the Track Egress Address and the TrackID MUST be respectively signaled as the IPv6 Address of the final destination and the RPLInstanceID field of the RPI that MUST be placed in the outer chain of IPv6 Headers.

In case of a NMPR, the outer chain of IPv6 Headers contains an IPv6 RH as well. If it is not fully consumed, then the final destination is the last entry in the RH; else it is the Destination Address in the IPv6 Header. When using the [\[RFC8138\]](#) compression, it is the last hop of the last SRH-6LoRH of the outer header in either case.

- \* 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. Else, i.e., if the Track Ingress is forwarding a packet into the Track, or if the the final destination is reached over the Track via the Track Egress but is located beyond it, then an IP-in-IP encapsulation is needed.

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



## Figure 7: Projecting a Non-Storing Route

A route indicated by an SRVIO may be loose, meaning that the node that owns the next listed Via Address is not necessarily a neighbor. Without proper loop avoidance mechanisms, the interaction of loose source routing and other mechanisms may effectively cause loops.

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:

Thubert, et al.

Expires 5 April 2021

[Page 18]

---

Internet-Draft

DAO Projection

October 2020

- \* In the uncompressed form the source of the packet is this router, the destination is the first Via Address in the SRVIO, 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 SRVIO, 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.

### 5.5. Storing Mode Projected Route

As illustrated in Figure 8, a P-DAO that carries a 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.

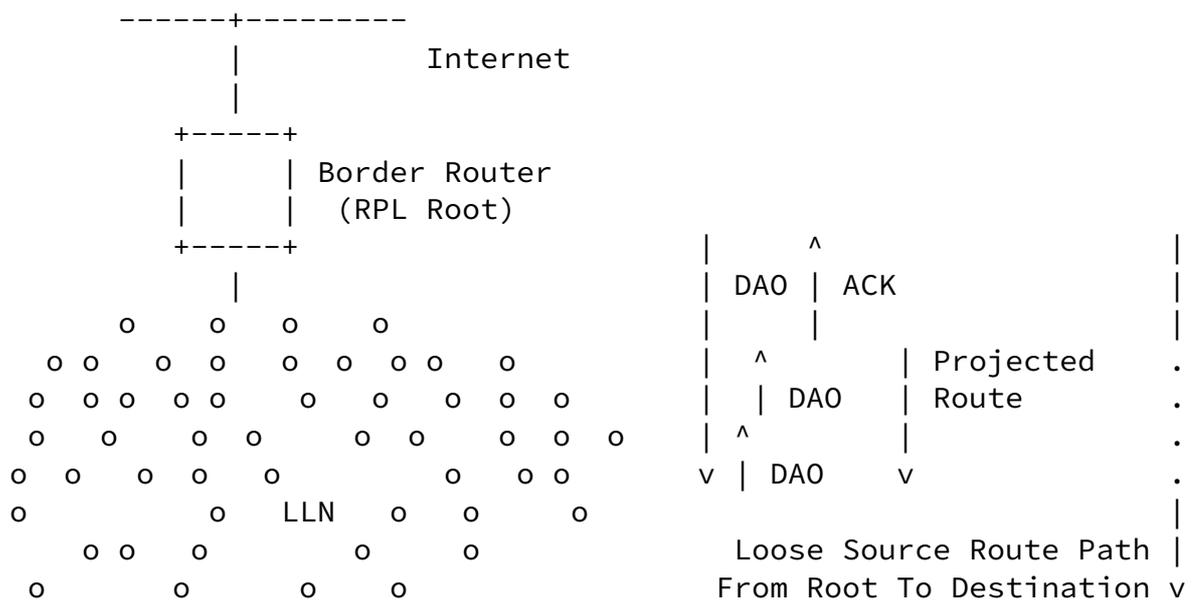


Figure 8: Projecting a route

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 VIO with the direct sequence of Via Addresses. The VIO follows one or more RTOs indicating the Targets to which the Track leads. The 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 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 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 VIO. The router MAY install additional routes towards the VIA Addresses that are the 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 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.

## 6. 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?

## 7. IANA Considerations

### 7.1. New RPL Control Codes

This document extends the IANA Subregistry created by RFC 6550 for RPL Control Codes as indicated in Table 1:

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

0x0A   PDR-ACK	This document
----------------	---------------

Table 1: New RPL Control Codes

### 7.2. New RPL Control Message Options

This document extends the IANA Subregistry created by [RFC 6550](#) for RPL Control Message Options as indicated in Table 2:

Value	Meaning	Reference
0x0B	Via Information option	This document
0x0C	Source-Routed Via Information option	This document
0x0D	Sibling Information option	This document

Table 2: RPL Control Message Options

### 7.3. 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 3:

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

Table 3: Initial PDR Flags

#### [7.4.](#) 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.

#### [7.5.](#) 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 4:

Value	Meaning	Reference
0	Unqualified acceptance	This document

Table 4: Acceptance values of the PDR-ACK Status

#### [7.6.](#) Subregistry for the PDR-ACK Rejection Status Values

IANA is requested to create a Subregistry for the PDR-ACK Rejection Status values.

Internet-Draft

DAO Projection

October 2020

- \* Possible values are 6-bit unsigned integers (0..63).
- \* Registration procedure is "Standards Action" [[RFC8126](#)].
- \* Initial allocation is as indicated in Table 5:

Value	Meaning	Reference
0	Unqualified rejection	This document

Table 5: Rejection values of the PDR-ACK Status

### [7.7.](#) SubRegistry for the Route Projection Options Flags

IANA is requested to create a Subregistry for the 5-bit Route Projection Options (RPO) 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 Route Projection Options (RPO) Flags.

### [7.8.](#) 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 6:

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

Table 6: Initial SIO Flags

### 7.9. 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.

### 8. Acknowledgments

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

### 9. 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

Thubert, et al.

Expires 5 April 2021

[Page 25]

---

Internet-Draft

DAO Projection

October 2020

Low-Power and Lossy Networks", [RFC 6550](#), DOI 10.17487/RFC6550, March 2012, <<https://www.rfc-editor.org/info/rfc6550>>.

- [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>>.

## 10. 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-04](#), 6 July 2020, <<https://tools.ietf.org/html/draft-pthubert-raw-architecture-04>>.

Thubert, et al.

Expires 5 April 2021

[Page 26]

---

Internet-Draft

DAO Projection

October 2020

[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-41](https://tools.ietf.org/html/draft-ietf-roll-useofrplinfo-41), 21 September 2020, <<https://tools.ietf.org/html/draft-ietf-roll-useofrplinfo-41>>.

[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 9. 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.

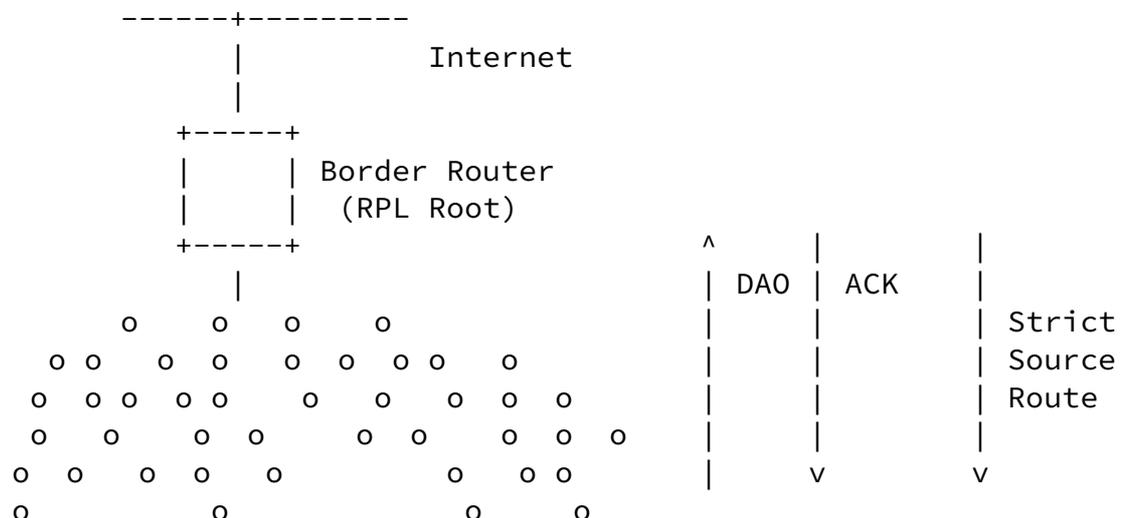


Figure 9: RPL Non-Storing Mode of operation

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 10:

- \* 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.

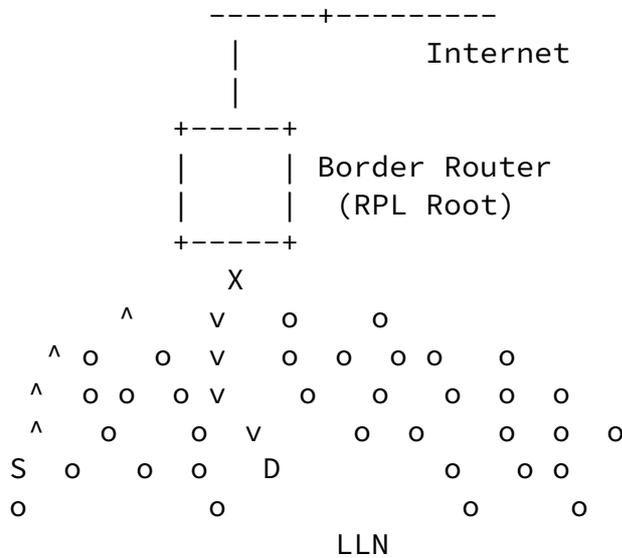


Figure 10: Routing Stretch between S and D via common parent X

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.

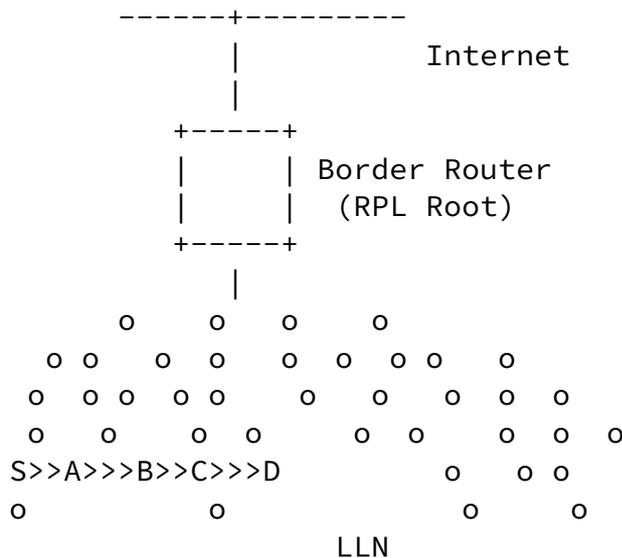


Figure 11: Projected Transversal Route

Internet-Draft

DAO Projection

October 2020

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

