Workgroup: ROLL
Updates: 6554 (if approved)
Published: 3 June 2021
Intended Status: Standards Track
Expires: 5 December 2021
Authors: P. Thubert, Ed.   R.A. Jadhav   M. Gillmore
         Cisco Systems   Huawei Tech   Itron

Root initiated routing state in RPL

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

Copyright Notice

Table of Contents

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

MOP: RPL Mode of Operation

P-DAO: Projected DAO

P-Route: Projected Route

PDR: P-DAO Request

RAN: RPL-Aware Node (either a RPL Router or a RPL-Aware Leaf)

RAL: RPL-Aware Leaf

RH: Routing Header

RPI: RPL Packet Information

RTO: RPL Target Option

RUL: RPL-Unaware Leaf

SIO: RPL Sibling Information Option

SR-VIO: A Source-Routed Via Information Option, used in Non-Storing-Mode P-DAO messages.

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. The owner of the namespace is the Track Ingress.

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.

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

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   TrackID     |K|D|P|  Flags  |   Reserved    | DAOSequence   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    +                                                               +
    |                                                               |
    +               IPv6 Address of the Track Ingress               +
    |                                                               |
    +                                                               +
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Option(s)...
    +-+-+-+-+-+-+-+-+
```

Figure 1: Projected DAO Base Object

New fields:

TrackID:
In the case of a P-DAO, the RPLInstanceID field is called TrackID. This is a naming convenience but does not change the semantics and format of the RPLInstanceID that is used as TrackID.

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

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

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 VIO(SF-VIO); the SF-VIO installs Storing-Mode Projected Route along a strict segment. The other is the Source-Routed VIO (SR-VIO); the SR-VIO installs a Non-Storing-Mode Projected Route 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 Information Options cannot. A P-DAO contains one or more RTOs that indicate the destinations that can be reached via the Track, and exactly one VIOthat 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. The expectation is that a node reports a Sibling Address in a SIO based on an active address registration [RFC8505] from that sibling for that address with the 'R' flag not set in the EARO. The node may assess the liveliness of the sibling at any time by performing a registration for one of its own addresses, either a link local or a global one, depending on whether the node expects the sibling to perform a matching advertisement in its own SIO.

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 an 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:

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

Figure 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:

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1|0|E| Length  | 6LoRH Type 7  | RPLInstanceID |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 where this node is Track Ingress. The source IPv6 address of the PDR signals the Track Ingress of the requested Track, and the TrackID is indicated in the message itself. One and only one RPL Target Option MUST be present in the message. 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:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   TrackID     |K|R|   Flags   |  ReqLifetime  | PDRSequence   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Option(s)...
+-+-+-+-+-+-+-+-+
```

Figure 4: New P-DAO Request Format

TrackID: 8-bit field indicating the RPLInstanceID associated with the Track.

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:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    TrackID    |     Flags     | Track Lifetime|  PDRSequence  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| PDR-ACK Status|                Reserved                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Option(s)...
+-+-+-+-+-+-+-+-+
```

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:

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|E|R|  Value    |
+-+-+-+-+-+-+-+-+
```

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 27 and Table 28.

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

6.3. Via Information Options

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

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type        | Option Length |     Flags     |   SegmentID   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Segm. Sequence | Seg. Lifetime |      SRH-6LoRH header         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
.                                                               .
.                     Via Address 1                             .
.                                                               .
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                              ....                             .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
.                                                               .
.                     Via Address n                             .
.                                                               .
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: VIOformat (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 VIOdeprecates any state for the Segment indicated by the SegmentID within the indicated Track and sets up the new information.

An VIOwith 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 VIOwith 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. The router that processes an SF-VIO MAY create additional routing state towards the nodes after self via the node immediatelt after self in the SF-VIO, but in case of memory shortage the routes to the Targets have precedence since they are the ones that the router commits to store.

In an SR-VIO, the list starts at the first hop and ends at a Track Egress. In that case, the Track Egress MUST be considered as an implicit Target, so it MUST NOT be listed it in a RPL Target Option. 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 Information 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:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   Type        | Option Length |Comp.|B|D|Flags|    Opaque     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |            Step of Rank       |          Reserved             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   +                                                               +
   .                                                               .
   .       Sibling DODAGID (if the D flag not set)               .
   .                                                               .
   +                                                               +
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   +                                                               +
   .                                                               .
   .                     Sibling Address                           .
   .                                                               .
   +                                                               +
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 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 if and only if the D flag is not set.

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

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

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 VIO(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 Information Option. A track is indentified by a tupple 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 VIOas opposed to the Path Sequence from a TIO. Also, a Segment Lifetime of 0 in an VIOindicates 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 7.3.2. 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 7.3.1. 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 11.13). 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 7.3.2 and Section 7.3.1 respectively.

7.1. Requesting a Track

A Node is free to ask the Root for a new Track for which it will be Ingress at any time. This is done with a PDR message, that indicates the desired TrackID and 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 resquesting 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 resquesting 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 resquesting 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 Storing-Mode P-Route 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 Storing-Mode P-Route 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]), and the Local RPLInstanceID is called TrackID.

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

  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. If 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.

7.3.1. Storing-Mode P-Route

Profile 1 extends RPL opertation in a Non-Storing Mode network with Storing-Mode Projected Routes that install segments along the main DODAG and enable to loose source routing between the Root and the targets.

As illustrated in Figure 9, 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.

```
      ------+---------
            |          Internet
            |
         +-----+
         |     | Border Router
         |     |  (RPL Root)
         +-----+                      |     ^                   |
            |                         | DAO | ACK               |
      o    o   o    o                 |     |                   |
  o o   o  o   o  o  o o   o          |  ^       | Projected    .
 o  o o  o o    o   o   o  o  o       |  | DAO   | Route        .
 o   o    o  o     o  o    o  o  o    | ^        |              .
o  o   o  o   o         o   o o       v | DAO    v              .
o          o   LLN   o   o     o                                |
    o o   o        o     o              Loose Source Route Path |
 o       o      o    o                 From Root To Destination v
```

Figure 9: 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 VIOis 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 Storing-Mode P-Route, 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.

7.3.2. Non-Storing-Mode P-Route

As illustrated in Figure 10, a P-DAO that carries an SR-VIO enables the Root to install a source-routed path from a Track Ingress towards a Target along the main DODAG.

```
        ------+---------
              |          Internet
              |
           +-----+
           |     | Border Router
           |     |  (RPL Root)
           +-----+                    |  P  ^ ACK
              |        Track          | DAO |
        o    o   o  o  Ingress X      V     |   X
    o o   o  o   o  o     o   X   o             X Source
   o  o o  o o    o   o  o    X  o  o           X Routed
   o   o    °  o     o   o   o X     o          X Segment
  o  o   o  o   o  o    o  o     X Track        X
     o  o  o  o             o     Egress

    o       o               o    o
  o          o             o     o
                                destination

                    LLN
```

Figure 10: Projecting a Non-Storing Route

When forwarding a packet to a destination for which the router determines that routing happens via a Track Target, the router inserts the Source Routing Header in the packet with the final destination at 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.

7.4. 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 ot 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 form 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.

8. Profiles

This document provides a set of tools that may or may not be needed by an implementation depending on the type of application it serves. This sections described profiles that can be implemented separately and can be used to discriminate what an implementation can and cannot do.

Profile 0 Profile 0 is the legacy support of [RPL] Non-Storing Mode. It provides the minimal common functionality that must be implemented as a prerequisite to all the Track-supporting profiles. The other Profiles extend Profile 0 with selected capabilities that this specification introduces on top.

Profile 1 (Storing-Mode P-Route Segments along the main DODAG) Profile 1 does not create new paths; it combines Storing and Non-Storing Modes to balance the size of the routing header in the packet and the amount of state in the intermediate routers in a Non-Storing Mode RPL DODAG.

Profile 2 (Non-Storing-Mode P-Route Segments along the main DODAG) Profile 2 extends Profile 0 with Strict Source-Routing Non-Storing-Mode Projected Routes along the main DODAG. Profile 2 provides the same capability to compress the SRH in packets down the main DODAG as Profile 1, but it require an encapsulation, in order to insert an additional SRH between the loose source routing hops.

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

Profile 4 Profile 4 extends Profile 2 with Strict Source-Routing Non-Storing-Mode Projected Routes to form Tracks inside the main DODAG. A Track is formed as one or more strict source source routed paths between the Root that is the Track Ingress, and the Track Egress that is the last node

Profile 5 Profile 5 Combines Profile 4 with Profile 1 and enables to loose source routing between the Ingress and the Egress of the Track. As in Profile 1, Storing-Mode Projected Routes connect the dots in the loose source route.

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

9. Example Track Signaling

The remainder of the section provides an example of how a Track can be signaled

```
                                         ===> F
   A ===> B ===> C ===> D===> E <
                                         ===> G
```

Figure 11: Reference Track

A is Track ingress, E is track Egress. C is stitching point. F and G are E's neighbors, "external" to the Track, and reachable from A over the Track A->E.

In a general manner we want:

- *P-DAO 1 signals C==>B==>E

- *P-DAO 2 signals A==>B==>C

- *P-DAO 3 signals F and G via the A==>E Track

P-DAO 3 being loose, it can only be non-storing. Note that since the Root is always the ingress of a Track, and all SR-VIOs are now

Track, the Root being signaled in the DAO base object can now be elided in the VIA list in SR-VIO. This enables the construction by the main root of the RFC 8138 optimized SRH-6LoRH in the SR-VIO, to be placed as is in the packet by the Root.

9.1. Using Storing-Mode Segments

A==>B==>C and C==>D==>E are segments of a same Track. Note that the storing mode signaling imposes strict continuity in a segment. One benefit of strict routing is that loops are avoided along the Track.

9.1.1. Stitched Segments

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

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

Table 1: P-DAO Messages

As a result the RIBs are set as follows:

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

Table 2: RIB setting

E recognizes that it is the Track Egress because it is both a Target and a Segment Endpoint.

Packets originated by A to E, F, or G, do not require an encapsulation. In any fashion, the outer headers of the packets that are forwarded along the Track have the following settings:

Header	IPv6 Source Addr.	IPv6 Dest. Addr.	TrackID in RPI
Outer	A	E, F or G	(A, 129)

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

Table 3: Packet header settings

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

*From P-DAO 2: A forwards to B and B forwards to C.

*From P-DAO 1: C forwards to D and D forwards to E.

*From Neighbor Cache Entry: C delivers the packet to F.

9.1.2. External routes

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

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

Table 4: P-DAO Messages

As a result the RIBs are set as follows:

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

Table 5: RIB setting

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

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

Table 6: Packet header settings

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

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

9.1.3. Segment Routing

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

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

Table 7: P-DAO Messages

As a result the RIBs are set as follows:

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

Table 8: RIB setting

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

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

Table 9: Packet header settings

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

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

9.2. Using Non-Storing-Mode joining Tracks

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

9.2.1. Stitched Tracks

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

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

Table 10: P-DAO Messages

As a result the RIBs are set as follows:

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

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

Table 11: RIB setting

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

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

Table 12: Packet header settings

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

- From P-DAO 2: A encapsulates the packet with destination of F in the Track signaled by P-DAO 2. The outer header has source A, destination B, an SRH that indicates C as the next loose hop, and a RPI indicating a TrackId of 129 from A's namespace.

- From the SRH: Packets forwarded by B have source A, destination C , a consumed SRH, and a RPI indicating a TrackId of 129 from A's namespace. C decapsulates.

- From P-DAO 1: C encapsulates the packet with destination of F in the Track signaled by P-DAO 1. The outer header has source C, destination D, an SRH that indicates E as the next loose hop, and a RPI indicating a TrackId of 131 from C's namespace. E decapsulates.

9.2.2. External routes

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

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

Table 13: P-DAO Messages

As a result the RIBs are set as follows:

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

Table 14: RIB setting

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

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

Table 15: Packet header settings

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

*From P-DAO 3: A encapsulates the packet with destination of F in the Track signaled by P-DAO 3. The outer header has source A, destination E, and a RPI indicating a TrackId of 141 from A's namespace. This recurses with:

*From P-DAO 2: A encapsulates the packet with destination of E in the Track signaled by P-DAO 2. The outer header has source A, destination B, an SRH that indicates C as the next loose hop, and a RPI indicating a TrackId of 129 from A's namespace.

*From the SRH: Packets forwarded by B have source A, destination C , a consumed SRH, and a RPI indicating a TrackId of 129 from A's namespace. C decapsulates.

*From P-DAO 1: C encapsulates the packet with destination of E in the Track signaled by P-DAO 1. The outer header has source C, destination D, an SRH that indicates E as the next loose hop, and a RPI indicating a TrackId of 131 from C's namespace. E decapsulates.

9.2.3. Segment Routing

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

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

Table 16: P-DAO Messages

As a result the RIBs are set as follows:

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

Table 17: RIB setting

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

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

Table 18: Packet header settings

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

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

Table 19: Packet header settings

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

Header	IPv6 Source Addr.	IPv6 Dest. Addr.	TrackID in RPI
Outer	C	D till D then E	(C, 131)
Middle	A	E	(A, 141)

Header	IPv6 Source Addr.	IPv6 Dest. Addr.	TrackID in RPI
Inner	X	E, F or G	N/A

Table 20: Packet header settings

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

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

- *From P-DAO 2: A encapsulates the packet with destination of C in the Track signaled by P-DAO 2. The outer header has source A, destination B, and a RPI indicating a TrackId of 129 from A's namespace. B decapsulates forwards to C based on a sibling connected route.

- *From the SRH: C consumes the SRH and makes the destination E.

- *From P-DAO 1: C encapsulates the packet with destination of E in the Track signaled by P-DAO 1. The outer header has source C, destination D, an SRH that indicates E as the next loose hop, and a RPI indicating a TrackId of 131 from C's namespace. E decapsulates.

10. 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?

11. IANA Considerations

11.1. 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 21: New Elective 6LoWPAN Routing Header Type

11.2. 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 22: New Critical 6LoWPAN Routing Header Type

11.3. 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 26:

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 23: Initial PDR Flags

11.4. New RPL Control Codes

This document extends the IANA Subregistry created by RFC 6550 for RPL Control Codes as indicated in Table 24:

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

Table 24: New RPL Control Codes

11.5. New RPL Control Message Options

This document extends the IANA Subregistry created by RFC 6550 for RPL Control Message Options as indicated in Table 25:

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

Table 25: RPL Control Message Options

11.6. 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 26:

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

Table 26: Initial PDR Flags

11.7. 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.

11.8. 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 27:

Value	Meaning	Reference
0	Unqualified acceptance	This document

Table 27: Acceptance values of the PDR-ACK Status

11.9. 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 28:

Value	Meaning	Reference
0	Unqualified rejection	This document

Table 28: Rejection values of the PDR-ACK Status

11.10. SubRegistry for the Via Information Options Flags

IANA is requested to create a Subregistry for the 5-bit Via Information Options (Via Information Option) Flags field. Each bit is tracked with the following qualities:

- Bit number (counting from bit 0 as the most significant bit)
- Capability description
- Reference

Registration procedure is "Standards Action" [RFC8126]. No bit is currently defined for the Via Information Options (Via Information Option) Flags.

11.11. 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 29:

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

Table 29: Initial SIO Flags

11.12. New Destination Advertisement Object Flag

This document modifies the "Destination Advertisement Object (DAO) Flags" registry initially created in Section 20.11 of [RPL] .

Section 3.1 also defines one new entry in the Registry as follows:

Bit Number	Capability Description	Reference
2 (suggested)	Projected DAO (P)	THIS RFC

Table 30: New Destination Advertisement Object (DAO) Flag

11.13. 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.

12. Acknowledgments

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

13. 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>.

14. 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., Ed., "An Architecture for IPv6 over the Time-Slotted Channel Hopping Mode of IEEE 802.15.4 (6TiSCH)", RFC 9030, DOI 10.17487/RFC9030, May 2021, <https://www.rfc-editor.org/info/rfc9030>.

[RAW-ARCHI] Thubert, P., Papadopoulos, G. Z., 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., Ed. and L. Zhao, "A Routing Protocol for Low-Power and Lossy Networks (RPL) Destination-Oriented Directed Acyclic Graph (DODAG) Configuration Option for the 6LoWPAN Routing Header", RFC 9035, DOI 10.17487/RFC9035, April 2021, <https://www.rfc-editor.org/info/rfc9035>.

[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>.

[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>.

[USEofRPLinfo] Robles, M.I., Richardson, M., and P. Thubert, "Using RPI Option Type, Routing Header for Source Routes, and IPv6-in-IPv6 Encapsulation in the RPL Data Plane", RFC 9008, DOI 10.17487/RFC9008, April 2021, <https://www.rfc-editor.org/info/rfc9008>.

[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 12. 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.

```
           ------+---------
                 |          Internet
                 |
              +-----+
              |     | Border Router
              |     |  (RPL Root)
              +-----+                      ^     |        |
                 |                         | DAO | ACK    |
           o    o   o    o                 |     |        | Strict
       o o   o  o   o  o  o o   o          |     |        | Source
      o  o o  o o    o   o   o  o  o       |     |        | Route
      o   o    o  o     o  o    o  o  o    |     |        |
     o  o   o  o   o         o   o o       |     v        v
     o          o             o     o
                       LLN
```

Figure 12: 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 13:

- 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.

```
                ------+---------
                 |          Internet
                 |
              +-----+
              |     | Border Router
              |     |  (RPL Root)
              +-----+
                 X
           ^    v   o    o
       ^ o   o  v   o  o  o o   o
      ^  o o  o v    o   o   o  o  o
      ^   o    o  v     o  o    o  o  o
     S  o   o  o   D         o   o o
     o          o             o     o
                       LLN
```

Figure 13: 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.

```
              ------+---------
                    |          Internet
                    |
                 +-----+
                 |     | Border Router
                 |     |  (RPL Root)
                 +-----+
                    |
              o    o   o    o
          o o   o  o   o  o  o o   o
         o  o o  o o    o   o   o  o  o
         o   o    o  o     o  o    o  o  o
        S>>A>>>B>>C>>>D         o   o o
        o          o             o     o
                          LLN
```

Figure 14: 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 oculd 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