Internet Engineering Task Force (IETF)
Request for Comments: 9008
Updates: 6550, 6553, 8138
Category: Standards Track
ISSN: 2070-1721

M.I. Robles
UTN-FRM/Aalto
M. Richardson
SSW
P. Thubert
Cisco

April 2021

Using RPI Option Type, Routing Header for Source Routes, and IPv6-in-IPv6 Encapsulation in the RPL Data Plane

Abstract

This document looks at different data flows through Low-Power and Lossy Networks (LLN) where RPL (IPv6 Routing Protocol for Low-Power and Lossy Networks) is used to establish routing. The document enumerates the cases where RPL Packet Information (RPI) Option Type (RFC 6553), RPL Source Route Header (RFC 6554), and IPv6-in-IPv6 encapsulation are required in the data plane. This analysis provides the basis upon which to design efficient compression of these headers. This document updates RFC 6553 by adding a change to the RPI Option Type. Additionally, this document updates RFC 6550 by defining a flag in the DODAG Information Object (DIO) Configuration option to indicate this change and updates RFC 8138 as well to consider the new Option Type when the RPL Option is decompressed.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9008.

Table of Contents

1. Introduction

RPL (IPv6 Routing Protocol for Low-Power and Lossy Networks) [RFC6550] is a routing protocol for constrained networks. [RFC6553] defines the RPL Option carried within the IPv6 Hop-by-Hop Options header to carry the RPLInstanceID and quickly identify inconsistencies (loops) in the routing topology. The RPL Option is commonly referred to as the RPL Packet Information (RPI), although the RPI is the routing information that is defined in [RFC6550] and transported in the RPL Option. RFC 6554 [RFC6554] defines the "RPL Source Route Header" (RH3), an IPv6 extension header to deliver datagrams within a RPL routing domain, particularly in Non-Storing mode.

These various items are referred to as RPL artifacts, and they are seen on all of the data plane traffic that occurs in RPL-routed networks; they do not, in general, appear on the RPL control plane at all, which is mostly hop-by-hop traffic (one exception being Destination Advertisement Object (DAO) messages in Non-Storing mode).

It has become clear from attempts to do multi-vendor interoperability, and from a desire to compress as many of the above artifacts as possible, that not all implementers agree when artifacts are necessary, or when they can be safely omitted, or removed.

The ROLL (Routing Over Low power and Lossy networks) Working Group analyzed how IPv6 rules [RFC2460] apply to the Storing and Non-Storing use of RPL. The result was 24 data-plane use cases. They are exhaustively outlined here in order to be completely unambiguous. During the processing of this document, new rules were published as [RFC8200], and this document was updated to reflect the normative changes in that document.

This document updates [RFC6553], changing the value of the Option Type of the RPL Option to make routers compliant with [RFC8200] ignore this option when it is not recognized.

A Routing Header Dispatch for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) (6LoRH) [RFC8138] defines a mechanism for compressing RPL Option information and Routing Header type 3 (RH3) [RFC6554], as well as an efficient IPv6-in-IPv6 technique.

Most of the use cases described herein require the use of IPv6-in-IPv6 packet encapsulation. When encapsulating and decapsulating packets, [RFC6040] MUST be applied to map the setting of the explicit congestion notification (ECN) field between inner and outer headers. Additionally, [TUNNELS] is recommended reading to explain the relationship of IP tunnels to existing protocol layers and the

challenges in supporting IP tunneling.

Unconstrained uses of RPL are not in scope of this document, and applicability statements for those uses may provide different advice, e.g., [ACP].

1.1. Overview

The rest of the document is organized as follows: Section 2 describes the terminology that is used. Section 3 provides a RPL overview. Section 4 describes the updates to RFC 6553, RFC 6550, and RFC 8138. Section 5 provides the reference topology used for the use cases. Section 6 describes the use cases included. Section 7 describes the Storing mode cases and Section 8 the Non-Storing mode cases. Section 9 describes the operational considerations of supporting RPL-unaware leaves. Section 10 depicts operational considerations for the proposed change on RPI Option Type, Section 11 the IANA considerations, and then Section 12 describes the security aspects.

2. Terminology and 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.

The following terminology defined in [RFC7102] applies to this document: LLN, RPL, RPL domain, and ROLL.

Consumed: A Routing Header is consumed when the Segments Left field is zero, which indicates that the destination in the IPv6 header is the final destination of the packet and that the hops in the Routing Header have been traversed.

RPL Leaf: An IPv6 host that is attached to a RPL router and obtains connectivity through a RPL Destination-Oriented Directed Acyclic Graph (DODAG). As an IPv6 node, a RPL leaf is expected to ignore a consumed Routing Header, and as an IPv6 host, it is expected to ignore a Hop-by-Hop Options header. Thus, a RPL leaf can correctly receive a packet with RPL artifacts. On the other hand, a RPL leaf is not expected to generate RPL artifacts or to support IP-in-IP encapsulation. For simplification, this document uses the standalone term leaf to mean a RPL leaf.

RPL Packet Information (RPI): The information defined abstractly in [RFC6550] to be placed in IP packets. The term is commonly used, including in this document, to refer to the RPL Option [RFC6553] that transports that abstract information in an IPv6 Hop-by-Hop Options header. [RFC8138] provides an alternate (more compressed) formatting for the same abstract information.

RPL-Aware Node (RAN): A device that implements RPL. Please note that the device can be found inside the LLN or outside LLN.

RPL-Aware Leaf (RAL): A RPL-aware node that is also a RPL leaf.

RPL-Unaware Node: A device that does not implement RPL, thus the device is RPL unaware. Please note that the device can be found inside the LLN.

RPL-Unaware Leaf (RUL): A RPL-unaware node that is also a RPL leaf.

6LoWPAN Node (6LN): [RFC6775] defines it as the following: "A 6LoWPAN node is any host or router participating in a LoWPAN. This term is used when referring to situations in which either a host or router can play the role described." In this document, a 6LN acts as a leaf.

6LoWPAN Router (6LR): [RFC6775] defines it as the following: "An intermediate router in the LoWPAN that is able to send and receive Router Advertisements (RAs) and Router Solicitations (RSs) as well as forward and route IPv6 packets. 6LoWPAN routers are present only in route-over topologies."

6LoWPAN Border Router (6LBR): [RFC6775] defines it as the following: "A border router located at the junction of separate 6LoWPAN networks or between a 6LoWPAN network and another IP network. There may be one or more 6LBRs at the 6LoWPAN network boundary. A 6LBR is the responsible authority for IPv6 prefix propagation for the 6LoWPAN network it is serving. An isolated LoWPAN also contains a 6LBR in the network, which provides the prefix(es) for the isolated network."

Flag Day: A flag day is caused when a network is reconfigured in a way that nodes running the older configuration cannot communicate with nodes running the new configuration. An example of a flag day is when the ARPANET changed from IP version 3 to IP version 4 on January 1, 1983 [RFC0801]. In the context of this document, a switch from RPI Option Type (0x63) to Option Type (0x23) presents as a disruptive changeover. In order to reduce the amount of time for such a changeover, Section 4.1.3 provides a mechanism to allow nodes to be incrementally upgraded.

Non-Storing Mode (Non-SM): A RPL mode of operation in which the RPL-aware nodes send information to the root about their parents. Thus, the root knows the topology. Because the root knows the topology, the intermediate 6LRs do not maintain routing state, and source routing is needed.

Storing Mode (SM): A RPL mode of operation in which RPL-aware nodes (6LRs) maintain routing state (of the children) so that source routing is not needed.

> Note: Due to lack of space in some tables, we refer to IPv6-in-IPv6 as IP6-IP6.

3. RPL Overview

RPL defines the RPL control message (control plane), which is an ICMPv6 message [RFC4443] with a Type of 155. DIS (DODAG Information Solicitation), DIO (DODAG Information Object), and DAO (Destination

Advertisement Object) messages are all RPL control messages but with different Code values. A RPL stack is shown in Figure 1.

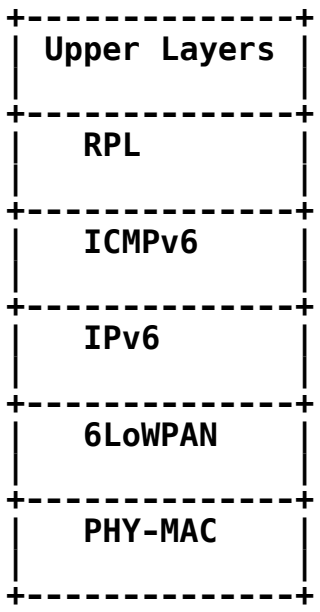


Figure 1: RPL Stack

RPL supports two modes of Downward internal traffic: in Storing mode (SM), it is fully stateful; in Non-Storing mode (non-SM), it is fully source routed. A RPL Instance is either fully Storing or fully Non-Storing, i.e., a RPL Instance with a combination of fully Storing and Non-Storing nodes is not supported with the current specifications at the time of writing this document. External routes are advertised with non-SM messaging even in an SM network, see Section 4.1.1

4. Updates to RFC 6550, RFC 6553, and RFC 8138

4.1. Updates to RFC 6550

4.1.1. Advertising External Routes with Non-Storing Mode Signaling

Section 6.7.8 of [RFC6550] introduces the 'E' flag that is set to indicate that the 6LR that generates the DAO redistributes external targets into the RPL network. An external target is a target that has been learned through an alternate protocol, for instance, a route to a prefix that is outside the RPL domain but reachable via a 6LR. Being outside of the RPL domain, a node that is reached via an external target cannot be guaranteed to ignore the RPL artifacts and cannot be expected to process the compression defined in [RFC8138] correctly. This means that the RPL artifacts should be contained in an IP-in-IP encapsulation that is removed by the 6LR, and that any remaining compression should be expanded by the 6LR before it forwards a packet outside the RPL domain.

This specification updates [RFC6550] to say that advertising external targets using Non-Storing mode DAO messaging even in a Storing mode network is RECOMMENDED. This way, external routes are not advertised within the DODAG, and all packets to an external target reach the

root like normal Non-Storing mode traffic. The Non-Storing mode DAO informs the root of the address of the 6LR that injects the external route, and the root uses IP-in-IP encapsulation to that 6LR, which terminates the IP-in-IP tunnel and forwards the original packet outside the RPL domain free of RPL artifacts.

In the other direction, for traffic coming from an external target into the LLN, the parent (6LR) that injects the traffic always encapsulates to the root. This whole operation is transparent to intermediate routers that only see traffic between the 6LR and the root, and only the root and the 6LRs that inject external routes in the network need to be upgraded to add this function to the network.

A RUL is a special case of external target when the target is actually a host, and it is known to support a consumed Routing Header and to ignore a Hop-by-Hop Options header as prescribed by [RFC8200]. The target may have been learned through an external routing protocol or may have been registered to the 6LR using [RFC8505].

In order to enable IP-in-IP all the way to a 6LN, it is beneficial that the 6LN supports decapsulating IP-in-IP, but that is not assumed by [RFC8504]. If the 6LN is a RUL, the root that encapsulates a packet SHOULD terminate the tunnel at a parent 6LR. The root may encapsulate all the way to the RUL if it is aware that the RUL supports IP-in-IP decapsulation and the artifacts in the outer header chain.

A node that is reachable over an external route is not expected to support [RFC8138]. Whether a decapsulation took place or not and even when the 6LR is delivering the packet to a RUL, the 6LR that injected an external route MUST undo the [RFC8138] compression on the packet before forwarding over that external route.

4.1.2. Configuration Options and Mode of Operation

Section 6.7.6 of [RFC6550] describes the DODAG Configuration option as containing a series of flags in the first octet of the payload.

Anticipating future work to revise RPL relating to how the LLN and DODAG are configured, this document renames the IANA "DODAG Configuration Option Flags" subregistry so that it applies to Mode of Operation (MOP) values zero (0) through six (6) only, leaving the flags unassigned for MOP value seven (7). The MOP is described in [RFC6550], Section 6.3.1.

In addition, this document reserves MOP value 7 for future expansion.

See Sections 11.2 and 11.3.

4.1.3. Indicating the New RPI in the DODAG Configuration Option Flag

In order to avoid a flag day caused by lack of interoperation between nodes of the new RPI Option Type (0x23) and old RPI Option Type (0x63), this section defines a flag in the DODAG Configuration option, to indicate when the new RPI Option Type can be safely used. This means that the flag is going to indicate the value of Option

Type that the network will be using for the RPL Option. Thus, when a node joins to a network, it will know which value to use. With this, RPL-capable nodes know if it is safe to use 0x23 when creating a new RPL Option. A node that forwards a packet with an RPI MUST NOT modify the Option Type of the RPL Option.

This is done using a DODAG Configuration option flag that will signal "RPI 0x23 enable" and propagate through the network. Section 6.3.1 of [RFC6550] defines a 3-bit Mode of Operation (MOP) in the DIO Base Object. The flag is defined only for MOP value between 0 to 6.

For a MOP value of 7, a node MUST use the RPI 0x23 option.

As stated in [RFC6550], the DODAG Configuration option is present in DIO messages. The DODAG Configuration option distributes configuration information. It is generally static, and it does not change within the DODAG. This information is configured at the DODAG root and distributed throughout the DODAG with the DODAG Configuration option. Nodes other than the DODAG root do not modify this information when propagating the DODAG Configuration option.

Currently, the DODAG Configuration option in [RFC6550] states that the unused bits "MUST be initialized to zero by the sender and MUST be ignored by the receiver." If the flag is received with a value zero, which is the default, then new nodes will remain compatible with RFC 6553 -- originating traffic with the old RPI Option Type value (0x63). If the flag is received with a value of 1, then the value for the RPL Option MUST be set to 0x23.

Bit number three of the Flags field in the DODAG Configuration option is to be used as shown in Table 1 (which is the same as Table 36 in Section 11 and is shown here for convenience):

Bit number	Description	Reference
3	RPI 0x23 enable	This document

Table 1: DODAG Configuration Option Flag to Indicate the RPI Flag Day

In the case of reboot, the node (6LN or 6LR) does not remember the RPI Option Type (i.e., whether or not the flag is set), so the node will not trigger DIO messages until a DIO message is received that indicates the RPI value to be used. The node will use the value 0x23 if the network supports this feature.

4.2. Updates to RFC 6553: Indicating the New RPI Option Type

This modification is required in order to be able to send, for example, IPv6 packets from a RPL-aware leaf to a RPL-unaware node through the Internet (see Section 7.2.1) without requiring IPv6-in-IPv6 encapsulation.

Section 6 of [RFC6553] states, as shown in Table 2, that in the

Option Type field of the RPL Option, the two high-order bits must be set to '01' and the third bit is equal to '1'. The first two bits indicate that the IPv6 node must discard the packet if it doesn't recognize the Option Type, and the third bit indicates that the Option Data may change in route. The remaining bits serve as the Option Type.

Hex Value	Binary Value			Description	Reference
	act	chg	rest		
0x63	01	1	00011	RPL Option	[RFC6553]

Table 2: Option Type in RPL Option

This document illustrates that it is not always possible to know for sure at the source whether a packet will travel only within the RPL domain or whether it will leave it.

At the time [RFC6553] was published, leaking a Hop-by-Hop Options header in the outer IPv6 header chain could potentially impact core routers in the Internet. So at that time, it was decided to encapsulate any packet with a RPL Option using IPv6-in-IPv6 in all cases where it was unclear whether the packet would remain within the RPL domain. In the exception case where a packet would still leak, the Option Type would ensure that the first router in the Internet that does not recognize the option would drop the packet and protect the rest of the network.

Even with [RFC8138], where the IPv6-in-IPv6 header is compressed, this approach yields extra bytes in a packet; this means consuming more energy and more bandwidth, incurring higher chances of loss, and possibly causing a fragmentation at the 6LoWPAN level. This impacts the daily operation of constrained devices for a case that generally does not happen and would not heavily impact the core anyway.

While the intention was and remains that the Hop-by-Hop Options header with a RPL Option should be confined within the RPL domain, this specification modifies this behavior in order to reduce the dependency on IPv6-in-IPv6 and protect the constrained devices. Section 4 of [RFC8200] clarifies the behavior of routers in the Internet as follows: "it is now expected that nodes along a packet's delivery path only examine and process the Hop-by-Hop Options header if explicitly configured to do so."

When unclear about the travel of a packet, it becomes preferable for a source not to encapsulate, accepting the fact that the packet may leave the RPL domain on its way to its destination. In that event, the packet should reach its destination and should not be discarded by the first node that does not recognize the RPL Option. However, with the current value of the Option Type, if a node in the Internet is configured to process the Hop-by-Hop Options header, and if such a node encounters an Option Type with the first two bits set to 01 and the node conforms to [RFC8200], it will drop the packet. Host

systems should do the same, irrespective of the configuration.

Thus, this document updates the Option Type of the RPL Option [RFC6553], naming it RPI Option Type for simplicity (Table 3): the two high order bits MUST be set to '00', and the third bit is equal to '1'. The first two bits indicate that the IPv6 node MUST skip over this option and continue processing the header ([RFC8200], Section 4.2) if it doesn't recognize the Option Type, and the third bit continues to be set to indicate that the Option Data may change en route. The rightmost five bits remain at 0x3(00011). This ensures that a packet that leaves the RPL domain of an LLN (or that leaves the LLN entirely) will not be discarded when it contains the RPL Option.

With the new Option Type, if an IPv6 (intermediate) node (RPL unaware) receives a packet with a RPL Option, it should ignore the Hop-by-Hop RPL Option (skip over this option and continue processing the header). This is relevant, as it was mentioned previously, in the case that there is a flow from RAL to Internet (see Section 7.2.1).

This is a significant update to [RFC6553].

Hex Value	Binary Value			Description	Reference
	act	chg	rest		
0x23	00	1	00011	RPL Option	This document

Table 3: Revised Option Type in RPL Option

Without the signaling described below, this change would otherwise create a lack of interoperation (flag day) for existing networks that are currently using 0x63 as the RPI Option Type value. A move to 0x23 will not be understood by those networks. It is suggested that RPL implementations accept both 0x63 and 0x23 when processing the header.

When forwarding packets, implementations SHOULD use the same value of RPI Type as was received. This is required because the RPI Option Type does not change en route ([RFC8200], Section 4.2). It allows the network to be incrementally upgraded and allows the DODAG root to know which parts of the network have been upgraded.

When originating new packets, implementations should have an option to determine which value to originate with. This option is controlled by the DODAG Configuration option (Section 4.1.3).

The change of RPI Option Type from 0x63 to 0x23 makes all nodes that are compliant with Section 4.2 of [RFC8200] tolerant of the RPL artifacts. There is no longer a need to remove the artifacts when sending traffic to the Internet. This change clarifies when to use IPv6-in-IPv6 headers and how to address them: the Hop-by-Hop Options header containing the RPI MUST always be added when 6LRs originate

packets (without IPv6-in-IPv6 headers), and IPv6-in-IPv6 headers MUST always be added when a 6LR finds that it needs to insert a Hop-by-Hop Options header containing the RPL Option. The IPv6-in-IPv6 header is to be addressed to the RPL root when on the way up, and to the end host when on the way down.

In the Non-Storing case, dealing with RPL-unaware leaf nodes is much easier as the 6LBR (DODAG root) has complete knowledge about the connectivity of all DODAG nodes, and all traffic flows through the root node.

The 6LBR can recognize RPL-unaware leaf nodes because it will receive a DAO about that node from the 6LR immediately above that RPL-unaware node.

The Non-Storing mode case does not require the Type change from 0x63 to 0x23, as the root can always create the right packet. The Type change does not adversely affect the Non-Storing case (see Section 4.1.3).

4.3. Updates to RFC 8138: Indicating the Way to Decompress with the New RPI Option Type

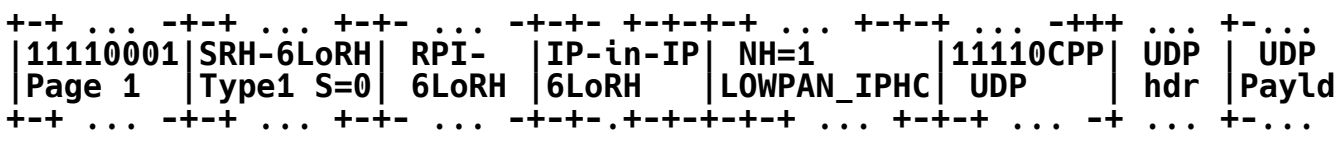
This modification is required in order to be able to decompress the RPL Option with the new Option Type of 0x23.

The RPI-6LoRH header provides a compressed form for the RPL RPI; see [RFC8138], Section 6. A node that is decompressing this header MUST decompress using the RPI Option Type that is currently active, that is, a choice between 0x23 (new) and 0x63 (old). The node will know which to use based upon the presence of the flag in the DODAG Configuration option defined in Section 4.1.3. For example, if the network is in 0x23 mode (by DIO option), then it should be decompressed to 0x23.

Section 7 of [RFC8138] documents how to compress the IPv6-in-IPv6 header.

There are potential significant advantages to having a single code path that always processes IPv6-in-IPv6 headers with no conditional branches.

In Storing mode, the scenarios where the flow goes from RAL to RUL and RUL to RUL include compression of the IPv6-in-IPv6 and RPI headers. The IPv6-in-IPv6 header MUST be used in this case, and it SHOULD be compressed as specified in [RFC8138], Section 7. Figure 2 illustrates the case in Storing mode where the packet is received from the Internet, then the root encapsulates the packet to insert the RPI. In that example, the leaf is not known to support RFC 8138, and the packet is encapsulated to the 6LR that is the parent and last hop to the final destination.



<-4bytes->   <-   RFC 6282   ->
                                        No RPL artifact

Figure 2: RPI Inserted by the Root in Storing Mode

In Figure 2, the source of the IPv6-in-IPv6 encapsulation is the root, so it is elided in the IP-in-IP 6LoRH. The destination is the parent 6LR of the destination of the inner packet so it cannot be elided. It is placed as the single entry in a Source Route Header 6LoRH (SRH-6LoRH) as the first 6LoRH. There is a single entry so the SRH-6LoRH Size is zero. In that example, the Type is 1 so the 6LR address is compressed to two bytes. This results in the total length of the SRH-6LoRH being four bytes. The RPI-6LoRH and then the IP-in-IP 6LoRH follow. When the IP-in-IP 6LoRH is removed, all the router headers that precede it are also removed. The Paging Dispatch [RFC8025] may also be removed if there was no previous Page change to a Page other than 0 or 1, since the LOWPAN_IPHC is encoded in the same fashion in the default Page 0 and in Page 1. The resulting packet to the destination is the inner packet compressed with [RFC6282].

5. Reference Topology

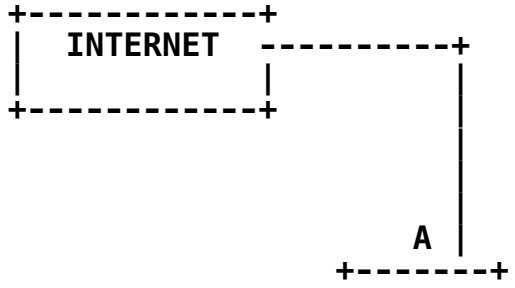
A RPL network in general is composed of a 6LBR, a Backbone Router (6BBR), a 6LR, and a 6LN as a leaf logically organized in a DODAG structure.

Figure 3 shows the reference RPL topology for this document. The nodes are labeled with letters so that they may be referenced in subsequent sections. In the figure, 6LR represents a full router node. The 6LN is a RPL-aware router or host (as a leaf). Additionally, for simplification purposes, it is supposed that the 6LBR has direct access to Internet and is the root of the DODAG, thus the 6BBR is not present in the figure.

The 6LN leaves marked as RAL (F, H, and I) are RPL nodes with no children hosts.

The leaves marked as RUL (G and J) are devices that do not speak RPL at all (RPL unaware), but use Router Advertisements, 6LoWPAN Duplicate Address Request and Duplicate Address Confirmation (DAR/DAC), and 6LoWPAN Neighbor Discovery (ND) only to participate in the network [RFC8505]. In the document, these leaves (G and J) are also referred to as a RUL.

The 6LBR (A) in the figure is the root of the Global DODAG.



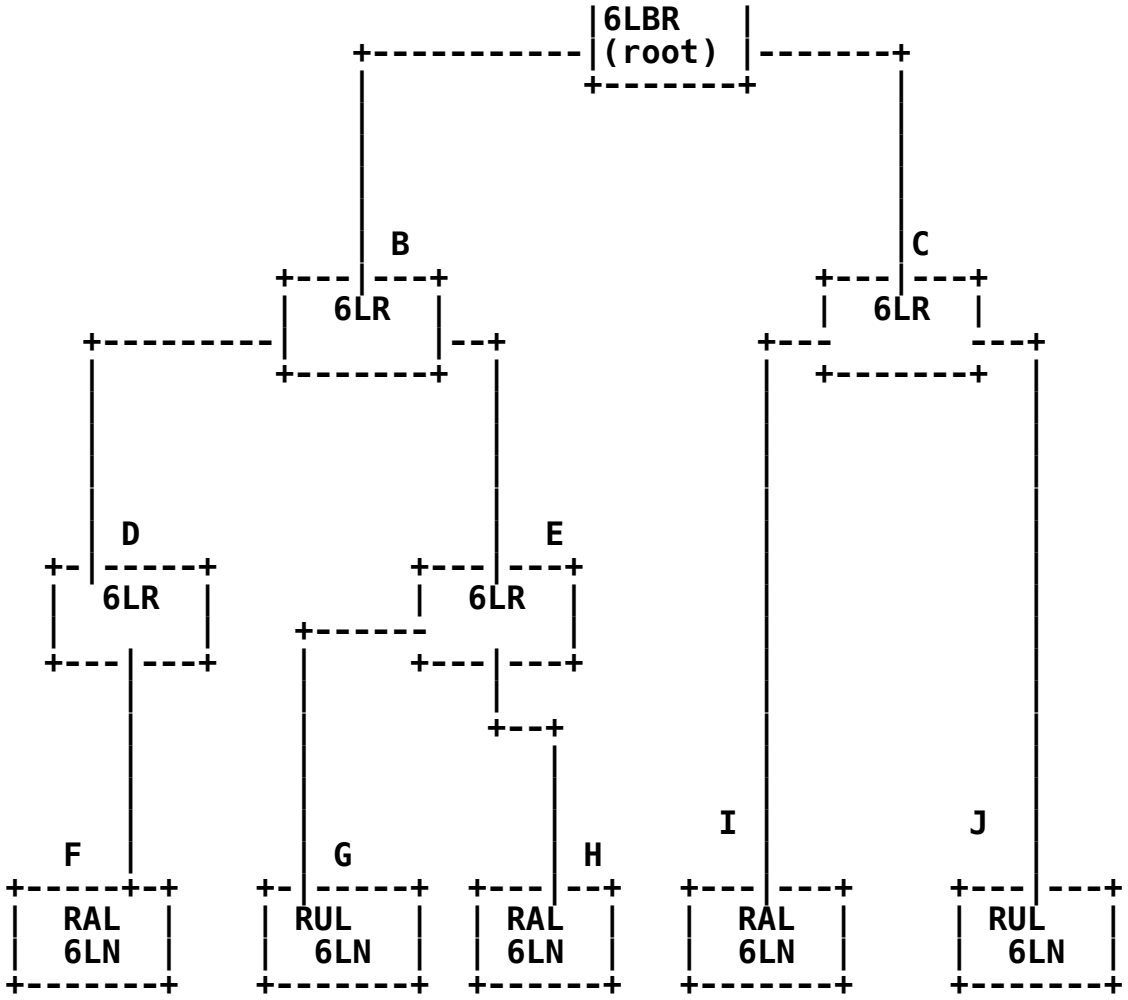


Figure 3: A Reference RPL Topology

6. Use Cases

In the data plane, a combination of RFC 6553, RFC 6554, and IPv6-in-IPv6 encapsulation are going to be analyzed for a number of representative traffic flows.

The use cases describe the communication in the following cases:

* Between RPL-aware nodes with the root (6LBR)

* Between RPL-aware nodes with the Internet

* Between RUL nodes within the LLN (e.g., see Section 7.1.4)

* Inside of the LLN when the final destination address resides outside of the LLN (e.g., see Section 7.2.3)

The use cases are as follows:

Interaction between leaf and root:

RAL to root

root to RAL

RUL to root

root to RUL

Interaction between leaf and Internet:

RAL to Internet

Internet to RAL

RUL to Internet

Internet to RUL

Interaction between leaves:

RAL to RAL

RAL to RUL

RUL to RAL

RUL to RUL

This document is consistent with the rule that a header cannot be inserted or removed on the fly inside an IPv6 packet that is being routed. This is a fundamental precept of the IPv6 architecture as outlined in [RFC8200].

As the Rank information in the RPI artifact is changed at each hop, it will typically be zero when it arrives at the DODAG root. The DODAG root MUST force it to zero when passing the packet out to the Internet. The Internet will therefore not see any SenderRank information.

Despite being legal to leave the RPI artifact in place, an intermediate router that needs to add an extension header (e.g., RH3 or RPL Option) MUST still encapsulate the packet in an (additional) outer IP header. The new header is placed after this new outer IP header.

A corollary is that an intermediate router can remove an RH3 or RPL Option only if it is placed in an encapsulating IPv6 header that is addressed _to_ this intermediate router. When doing the above, the whole encapsulating header must be removed. (A replacement may be added.)

Both the RPL Option and the RH3 headers may be modified in very specific ways by routers on the path of the packet without the need to add and remove an encapsulating header. Both headers were designed with this modification in mind, and both the RPL RH3 and the RPL Option are marked mutable but recoverable: so an IPsec Authentication Header (AH) can be applied across these headers, but

it cannot secure the values that mutate.

The RPI MUST be present in every single RPL data packet.

Prior to [RFC8138], there was significant interest in creating an exception to this rule and removing the RPI for Downward flows in Non-Storing mode. This exception covered a very small number of cases, and caused significant interoperability challenges while adding significant interest in the code and tests. The ability to compress the RPI down to three bytes or less removes much of the pressure to optimize this any further.

Throughout the following subsections, the examples are described in more detail in the first subsections, and more concisely in the later ones.

The use cases are delineated based on the following IPV6 and RPL mandates:

The RPI has to be in every packet that traverses the LLN.

- Because of the above requirement, packets from the Internet have to be encapsulated.

- A header cannot be inserted or removed on the fly inside an IPv6 packet that is being routed.

- Extension headers may not be added or removed except by the sender or the receiver.

- RPI and RH3 headers may be modified by routers on the path of the packet without the need to add and remove an encapsulating header.

- An RH3 or RPL Option can only be removed by an intermediate router if it is placed in an encapsulating IPv6 header, which is addressed to the intermediate router.

- The Non-Storing mode requires downstream encapsulation by the root for RH3.

The use cases are delineated based on the following assumptions:

This document assumes that the LLN is using the no-drop RPI Option Type (0x23).

- Each IPv6 node (including Internet routers) obeys [RFC8200], so that the 0x23 RPI Option Type can be safely inserted.

- All 6LRs obey [RFC8200].

- The RPI is ignored at the IPv6 destination (dst) node (RUL).

- In the use cases, we assume that the RAL supports IP-in-IP encapsulation.

- In the use cases, we don't assume that the RUL supports IP-in-IP encapsulation.

- For traffic leaving a RUL, if the RUL adds an opaque RPI, then the 6LR as a RPL Border Router SHOULD rewrite the RPI to indicate the selected Instance and set the flags.

- The description for RALs applies to RAN in general.

- Unconstrained uses of RPL are not in scope of this document.

- Compression is based on [RFC8138].

- The flow label [RFC6437] is not needed in RPL.

7. Storing Mode

In Storing mode (SM) (fully stateful), the sender can determine if the destination is inside the LLN by looking if the destination address is matched by the DIO's Prefix Information Option (PIO) option.

Table 4 itemizes which headers are needed in each of the following scenarios. It indicates whether an IPv6-in-IPv6 header must be added and to which destination it must be addressed:

1. the final destination (the RAL node that is the target (tgt)),

2. the "root", or

3. the 6LR parent of a RUL.

In cases where no IPv6-in-IPv6 header is needed, the column states "No", and the destination is N/A (Not Applicable). If the IPv6-in-IPv6 header is needed, the column shows "must".

In all cases, the RPI is needed, since it identifies inconsistencies (loops) in the routing topology. In general, the RH3 is not needed because it is not used in Storing mode. However, there is one scenario (from the root to the RUL in SM) where the RH3 can be used to point at the RUL (Table 8).

The leaf can be a router 6LR or a host, both indicated as 6LN. The root refers to the 6LBR (see Figure 3).

Interaction between	Use Case	IPv6-in-IPv6	IPv6-in-IPv6 dst
Leaf - Root	RAL to root	No	N/A
	root to RAL	No	N/A
	root to RUL	must	6LR

	RUL to root	must	root
Leaf - Internet	RAL to Int	may	root
	Int to RAL	must	RAL (tgt)
	RUL to Int	must	root
	Int to RUL	must	6LR
Leaf - Leaf	RAL to RAL	No	N/A
	RAL to RUL	No(up)	N/A
		must(down)	6LR
	RUL to RAL	must(up)	root
		must(down)	RAL
	RUL to RUL	must(up)	root
		must(down)	6LR

Table 4: IPv6-in-IPv6 Encapsulation in Storing Mode

7.1. Storing Mode: Interaction between Leaf and Root

This section describes the communication flow in Storing mode (SM) between the following:

RAL to root

root to RAL

RUL to root

root to RUL

7.1.1. SM: Example of Flow from RAL to Root

In Storing mode, RPI [RFC6553] is used to send the RPLInstanceID and Rank information.

In this case, the flow comprises:

RAL (6LN) --> 6LR_i --> root (6LBR)

For example, a communication flow could be: Node F (6LN) --> Node D (6LR_i) --> Node B (6LR_i) --> Node A root (6LBR)

The RAL (Node F) inserts the RPI, and sends the packet to the 6LR (Node D), which decrements the Rank in the RPI and sends the packet up. When the packet arrives at the 6LBR (Node A), the RPI is removed and the packet is processed.

No IPv6-in-IPv6 header is required.

The RPI can be removed by the 6LBR because the packet is addressed to the 6LBR. The RAL must know that it is communicating with the 6LBR to make use of this scenario. The RAL can know the address of the 6LBR because it knows the address of the root via the DODAGID in the DIO messages.

Table 5 summarizes which headers are needed for this use case.

Header	RAL src	6LR_i	6LBR dst
Added headers	RPI	--	--
Modified headers	--	RPI	--
Removed headers	--	--	RPI
Untouched headers	--	--	--

Table 5: SM: Summary of the Use of Headers from RAL to Root

7.1.2. SM: Example of Flow from Root to RAL

In this case, the flow comprises:

root (6LBR) --> 6LR_i --> RAL (6LN)

For example, a communication flow could be: Node A root (6LBR) --> Node B (6LR_i) --> Node D (6LR_i) --> Node F (6LN)

In this case, the 6LBR inserts RPI and sends the packet down. The 6LR increments the Rank in the RPI (it examines the RPLInstanceID to identify the right forwarding table). The packet is processed in the RAL, and the RPI is removed.

No IPv6-in-IPv6 header is required.

Table 6 summarizes which headers are needed for this use case.

Header	6LBR src	6LR_i	RAL dst
Added headers	RPI	--	--

Modified headers	--	RPI	--
Removed headers	--	--	RPI
Untouched headers	--	--	--

Table 6: SM: Summary of the Use of Headers from Root to RAL

7.1.3. SM: Example of Flow from Root to RUL

In this case, the flow comprises:

root (6LBR) --> 6LR_i --> RUL (IPv6 dst node)

For example, a communication flow could be: Node A (6LBR) --> Node B (6LR_i) --> Node E (6LR_n) --> Node G (RUL)

6LR_i (Node B) represents the intermediate routers from the source (6LBR) to the destination (RUL), and 1 <= i <= n, where n is the total number of routers (6LR) that the packet goes through, from the 6LBR (Node A) to the RUL (Node G).

The 6LBR will encapsulate the packet in an IPv6-in-IPv6 header and prepend an RPI. The IPv6-in-IPv6 header is addressed to the 6LR parent of the RUL (6LR_n). The 6LR parent of the RUL removes the header and sends the packet to the RUL.

Table 7 summarizes which headers are needed for this use case.

Header	6LBR src	6LR_i	6LR_n	RUL dst
Added headers	IP6-IP6 (RPI)	--	--	--
Modified headers	--	RPI	--	--
Removed headers	--	--	IP6-IP6 (RPI)	--
Untouched headers	--	IP6-IP6	--	--

Table 7: SM: Summary of the Use of Headers from Root to RUL

IP-in-IP encapsulation may be avoided for root-to-RUL communication. In SM, it can be replaced by a loose RH3 header that indicates the RUL. In which case, the packet is routed to the 6LR as a normal SM operation, then the 6LR forwards to the RUL based on the RH3, and the RUL ignores both the consumed RH3 and the RPI, as in Non-Storing mode.

Table 8 summarizes which headers are needed for this scenario.

Header	6LBR src	6LR_i i=(1,..,n-1)	6LR_n	RUL dst
Added headers	RPI, RH3	--	--	--
Modified headers	--	RPI	RPI, RH3(consumed)	--
Removed headers	--	--	--	--
Untouched headers	--	RH3	--	RPI, RH3 (both ignored)

Table 8: SM: Summary of the Use of Headers from Root to RUL without Encapsulation

7.1.4. SM: Example of Flow from RUL to Root

In this case, the flow comprises:

RUL (IPv6 src node) --> 6LR_1 --> 6LR_i --> root (6LBR)

For example, a communication flow could be: Node G (RUL) --> Node E (6LR_1) --> Node B (6LR_i) --> Node A root (6LBR)

6LR_i represents the intermediate routers from the source (RUL) to the destination (6LBR), and 1 <= i <= n, where n is the total number of routers (6LR) that the packet goes through, from the RUL to the 6LBR.

When the packet arrives from the RUL (Node G) to 6LR_1 (Node E), the 6LR_1 will encapsulate the packet in an IPv6-in-IPv6 header with an RPI. The IPv6-in-IPv6 header is addressed to the root (Node A). The root removes the header and processes the packet.

Table 9 summarizes which headers are needed for this use case where the IPv6-in-IPv6 header is addressed to the root (Node A).

Header	RUL src	6LR_1	6LR_i	6LBR dst
Added headers	--	IP6-IP6 (RPI)	--	--
Modified headers	--	--	RPI	--
Removed headers	--	--	--	IP6-IP6 (RPI)
Untouched headers	--	--	IP6-IP6	--

Table 9: SM: Summary of the Use of Headers from RUL to Root

7.2. SM: Interaction between Leaf and Internet

This section describes the communication flow in Storing mode (SM) between the following:

RAL to Internet

Internet to RAL

RUL to Internet

Internet to RUL

7.2.1. SM: Example of Flow from RAL to Internet

In this case, the flow comprises:

RAL (6LN) --> 6LR_i --> root (6LBR) --> Internet

For example, the communication flow could be: Node F (RAL) --> Node D (6LR_i) --> Node B (6LR_i) --> Node A root (6LBR) --> Internet

6LR_i represents the intermediate routers from the source (RAL) to the root (6LBR), and 1 <= i <= n, where n is the total number of routers (6LR) that the packet goes through, from the RAL to the 6LBR.

RPL information from RFC 6553 may go out to Internet as it will be ignored by nodes that have not been configured to be RPL aware. No IPv6-in-IPv6 header is required.

On the other hand, the RAL may insert the RPI encapsulated in an IPv6-in-IPv6 header to the root. Thus, the root removes the RPI and sends the packet to the Internet.

> Note: In this use case, a leaf node is used, but this use case can also be applicable to any RPL-aware node type (e.g., 6LR).

Table 10 summarizes which headers are needed for this use case when there is no encapsulation. Note that the RPI is modified by 6LBR to set the SenderRank to zero in the case that it is not already zero. Table 11 summarizes which headers are needed when encapsulation to the root takes place.

Header	RAL src	6LR_i	6LBR	Internet dst
Added headers	RPI	--	--	--
Modified headers	--	RPI	RPI	--
Removed headers	--	--	--	--
Untouched headers	--	--	--	RPI (Ignored)

Table 10: SM: Summary of the Use of Headers from RAL to Internet with No Encapsulation

Header	RAL src	6LR_i	6LBR	Internet dst
Added headers	IP6-IP6 (RPI)	--	--	--
Modified headers	--	RPI	--	--
Removed headers	--	--	IP6-IP6 (RPI)	--
Untouched headers	--	IP6-IP6	--	--

Table 11: SM: Summary of the Use of Headers from RAL to Internet with Encapsulation to the Root (6LBR)

7.2.2. SM: Example of Flow from Internet to RAL

In this case, the flow comprises:

Internet --> root (6LBR) --> 6LR_i --> RAL (6LN)

For example, a communication flow could be: Internet --> Node A root (6LBR) --> Node B (6LR_1) --> Node D (6LR_n) --> Node F (RAL)

When the packet arrives from Internet to 6LBR, the RPI is added in a outer IPv6-in-IPv6 header (with the IPv6-in-IPv6 destination address set to the RAL) and sent to the 6LR, which modifies the Rank in the RPI. When the packet arrives at the RAL, the packet is decapsulated, which removes the RPI before the packet is processed.

Table 12 summarizes which headers are needed for this use case.

Header	Internet src	6LBR	6LR_i	RAL dst
Added headers	--	IP6-IP6 (RPI)	--	--
Modified headers	--	--	RPI	--
Removed headers	--	--	--	IP6-IP6 (RPI)
Untouched headers	--	--	--	--

Table 12: SM: Summary of the Use of Headers from Internet to RAL

7.2.3. SM: Example of Flow from RUL to Internet

In this case, the flow comprises:

RUL (IPv6 src node) --> 6LR_1 --> 6LR_i --> root (6LBR) --> Internet

For example, a communication flow could be: Node G (RUL) --> Node E (6LR_1) --> Node B (6lR_i) --> Node A root (6LBR) --> Internet

The node 6LR_1 (i=1) will add an IPv6-in-IPv6 (RPI) header addressed to the root such that the root can remove the RPI before passing upwards. In the intermediate 6LR, the Rank in the RPI is modified.

The originating node will ideally leave the IPv6 flow label as zero so that the packet can be better compressed through the LLN. The 6LBR will set the flow label of the packet to a non-zero value when sending to the Internet. For details, check [RFC6437].

Table 13 summarizes which headers are needed for this use case.

Header	IPv6 src (RUL)	6LR_1	6LR_i i=(2,..,n)	6LBR	Internet dst
Added headers	--	IP6-IP6 (RPI)	--	--	--
Modified headers	--	--	RPI	--	--
Removed headers	--	--	--	IP6-IP6 (RPI)	--
Untouched headers	--	--	--	--	--

Table 13: SM: Summary of the Use of Headers from RUL to Internet

7.2.4. SM: Example of Flow from Internet to RUL

In this case, the flow comprises:

Internet --> root (6LBR) --> 6LR_i --> RUL (IPv6 dst node)

For example, a communication flow could be: Internet --> Node A root (6LBR) --> Node B (6LR_i) --> Node E (6LR_n) --> Node G (RUL)

The 6LBR will have to add an RPI within an IPv6-in-IPv6 header. The IPv6-in-IPv6 encapsulating header is addressed to the 6LR parent of the RUL.

Further details about this are mentioned in [RFC9010], which specifies RPL routing for a 6LN acting as a plain host and being unaware of RPL.

The 6LBR may set the flow label on the inner IPv6-in-IPv6 header to zero in order to aid in compression [RFC8138] [RFC6437].

Table 14 summarizes which headers are needed for this use case.

Header	Internet src	6LBR	6LR_i i=(1,..,n-1)	6LR_n	RUL dst
Added headers	--	IP6-IP6 (RPI)	--	--	--
Modified headers	--	--	RPI	--	--
Removed headers	--	--	--	IP6-IP6 (RPI)	--
Untouched headers	--	--	--	--	--

Table 14: SM: Summary of the Use of Headers from Internet to RUL

7.3. SM: Interaction between Leaf and Leaf

This section describes the communication flow in Storing mode (SM) between the following:

RAL to RAL

RAL to RUL

RUL to RAL

RUL to RUL

7.3.1. SM: Example of Flow from RAL to RAL

In [RFC6550], RPL allows a simple, one-hop optimization for both Storing and Non-Storing networks. A node may send a packet destined to a one-hop neighbor directly to that node. See Section 9 of [RFC6550].

When the nodes are not directly connected, then the flow comprises the following in the Storing mode:

RAL src (6LN) --> 6LR_ia --> common parent (6LR_x) --> 6LR_id --> RAL dst (6LN)

For example, a communication flow could be: Node F (RAL src) --> Node D (6LR_ia) --> Node B (6LR_x) --> Node E (6LR_id) --> Node H (RAL dst)

6LR_ia (Node D) represents the intermediate routers from the source

to the common parent 6LR_x (Node B), and 1 <= ia <= n, where n is the total number of routers (6LR) that the packet goes through, from the RAL (Node F) to the common parent 6LR_x (Node B).

6LR_id (Node E) represents the intermediate routers from the common parent 6LR_x (Node B) to the destination RAL (Node H), and 1 <= id <= m, where m is the total number of routers (6LR) that the packet goes through, from the common parent (6LR_x) to the destination RAL (Node H).

It is assumed that the two nodes are in the same RPL domain (that they share the same DODAG root). At the common parent (Node B), the direction flag ('O' flag) of the RPI is changed (from decreasing ranks to increasing ranks).

While the 6LR nodes will update the RPI, no node needs to add or remove the RPI, so no IPv6-in-IPv6 headers are necessary.

Table 15 summarizes which headers are needed for this use case.

Header	RAL src	6LR_ia	6LR_x (common parent)	6LR_id	RAL dst
Added headers	RPI	--	--	--	--
Modified headers	--	RPI	RPI	RPI	--
Removed headers	--	--	--	--	RPI
Untouched headers	--	--	--	--	--

Table 15: SM: Summary of the Use of Headers from RAL to RAL

7.3.2. SM: Example of Flow from RAL to RUL

In this case, the flow comprises:

RAL src (6LN) --> 6LR_ia --> common parent (6LBR, the root) --> 6LR_id --> RUL (IPv6 dst node)

For example, a communication flow could be: Node F (RAL) --> Node D --> Node B --> Node A --> Node B --> Node E --> Node G (RUL)

6LR_ia represents the intermediate routers from the source (RAL) to the common parent (the root), and 1 <= ia <= n, where n is the total number of routers (6LR) that the packet goes through, from the RAL to the root.

6LR_id (Node E) represents the intermediate routers from the root (Node B) to the destination RUL (Node G). In this case, 1 <= id <=

m, where m is the total number of routers (6LR) that the packet goes through, from the root down to the destination RUL.

In this case, the packet from the RAL goes to the 6LBR because the route to the RUL is not injected into the RPL SM. Thus, the RAL inserts an RPI (RPI1) addressed to the root (6LBR). The root does not remove the RPI1 (the root cannot remove an RPI if there is no encapsulation). The root inserts an IPv6-in-IPv6 encapsulation with an RPI2 and sends it to the 6LR parent of the RUL, which removes the encapsulation and RPI2 before passing the packet to the RUL.

Table 16 summarizes which headers are needed for this use case.

Header	RAL src	6LR_ia	6LBR	6LR_id	6LR_m	RUL dst
Added headers	RPI1	--	IP6-IP6 (RPI2)	--	--	--
Modified headers	--	RPI1	--	RPI2	--	--
Removed headers	--	--	--	--	IP6-IP6 (RPI2)	--
Untouched headers	--	--	RPI1	RPI1	RPI1	RPI1 (ignored)

Table 16: SM: Summary of the Use of Headers from RAL to RUL

7.3.3. SM: Example of Flow from RUL to RAL

In this case, the flow comprises:

RUL (IPv6 src node) --> 6LR_ia --> 6LBR --> 6LR_id --> RAL dst (6LN)

For example, a communication flow could be: Node G (RUL) --> Node E --> Node B --> Node A --> Node B --> Node D --> Node F (RAL)

6LR_ia (Node E) represents the intermediate routers from the source (RUL) (Node G) to the root (Node A). In this case, 1 <= ia <= n, where n is the total number of routers (6LR) that the packet goes through, from the source to the root.

6LR_id represents the intermediate routers from the root (Node A) to the destination RAL (Node F). In this case, 1 <= id <= m, where m is the total number of routers (6LR) that the packet goes through, from the root to the destination RAL.

The 6LR_1 (Node E) receives the packet from the RUL (Node G) and inserts the RPI (RPI1) encapsulated in an IPv6-in-IPv6 header to the root. The root removes the outer header including the RPI (RPI1) and inserts a new RPI (RPI2) addressed to the destination RAL (Node F).

Table 17 summarizes which headers are needed for this use case.

Header	RUL src	6LR_1	6LR_ia	6LBR	6LR_id	RAL dst
Added headers	--	IP6-IP6 (RPI1)	--	IP6-IP6 (RPI2)	--	--
Modified headers	--	--	RPI1	--	RPI2	--
Removed headers	--	--	--	IP6-IP6 (RPI1)	--	IP6-IP6 (RPI2)
Untouched headers	--	--	--	--	--	--

Table 17: SM: Summary of the Use of Headers from RUL to RAL

7.3.4. SM: Example of Flow from RUL to RUL

In this case, the flow comprises:

RUL (IPv6 src node) --> 6LR_1 --> 6LR_ia --> 6LBR --> 6LR_id --> RUL (IPv6 dst node)

For example, a communication flow could be: Node G (RUL src) --> Node E --> Node B --> Node A (root) --> Node C --> Node J (RUL dst)

Internal nodes 6LR_ia (e.g., Node E or Node B) is the intermediate router from the RUL source (Node G) to the root (6LBR) (Node A). In this case, 1 <= ia <= n, where n is the total number of routers (6LR) that the packet goes through, from the RUL to the root. 6LR_1 applies when ia=1.

6LR_id (Node C) represents the intermediate routers from the root (Node A) to the destination RUL (Node J). In this case, 1 <= id <= m, where m is the total number of routers (6LR) that the packet goes through, from the root to the destination RUL.

The 6LR_1 (Node E) receives the packet from the RUL (Node G) and adds the RPI (RPI1) in an IPv6-in-IPv6 encapsulation directed to the root. The root removes the outer header including the RPI (RPI1) and inserts a new RPI (RPI2) addressed to the 6LR parent of the RUL.

Table 18 summarizes which headers are needed for this use case.

Header	RUL src	6LR_1	6LR_ia	6LBR	6LR_id	6LR_n	RUL dst
Added headers	--	IP6-IP6 (RPI1)	--	IP6-IP6 (RPI1)	--	--	--

Modified headers	--	--	RPI1	--	RPI2	--	--
Removed headers	--	--	--	IP6-IP6 (RPI1)	--	IP6-IP6 (RPI2)	--
Untouched headers	--	--	--	--	--	--	--

Table 18: SM: Summary of the Use of Headers from RUL to RUL

8. Non-Storing Mode

In Non-Storing mode (Non-SM) (fully source routed), the 6LBR (DODAG root) has complete knowledge about the connectivity of all DODAG nodes and all traffic flows through the root node. Thus, there is no need for all nodes to know about the existence of RPL-unaware nodes. Only the 6LBR needs to act if compensation is necessary for RPL-unaware receivers.

Table 19 summarizes which headers are needed in the following scenarios and indicates when the RPI, RH3, and IPv6-in-IPv6 header are to be inserted. The last column depicts the target destination of the IPv6-in-IPv6 header: 6LN (indicated by "RAL"), 6LR (parent of a RUL), or the root. In cases where no IPv6-in-IPv6 header is needed, the column indicates "No". There is no expectation on RPL that RPI can be omitted because it is needed for routing, quality of service, and compression. This specification expects that an RPI is always present. The term "may(up)" means that the IPv6-in-IPv6 header may be necessary in the Upward direction. The term "must(up)" means that the IPv6-in-IPv6 header must be present in the Upward direction. The term "must(down)" means that the IPv6-in-IPv6 header must be present in the Downward direction.

The leaf can be a router 6LR or a host, both indicated as 6LN (Figure 3). In Table 19, the (1) indicates a 6TiSCH case [RFC8180], where the RPI may still be needed for the RPLInstanceID to be available for priority/channel selection at each hop.

Interaction between	Use Case	RPI	RH3	IPv6-in-IPv6	IP-in-IP dst
Leaf - Root	RAL to root	Yes	No	No	No
	root to RAL	Yes	Yes	No	No
	root to RUL	Yes (1)	Yes	No	6LR
	RUL to root	Yes	No	must	root

Leaf - Internet	RAL to Int	Yes	No	may(up)	root	
	Int to RAL	Yes	Yes	must	RAL	
	RUL to Int	Yes	No	must	root	
	Int to RUL	Yes	Yes	must	6LR	
Leaf - Leaf	RAL to RAL	Yes	Yes	may(up)	root	
				must(down)	RAL	
	RAL to RUL	Yes	Yes	may(up)	root	
				must(down)	6LR	
	RUL to RAL	Yes	Yes	must(up)	root	
				must(down)	RAL	
	RUL to RUL	Yes	Yes	must(up)	root	
				must(down)	6LR	

Table 19: Headers Needed in Non-Storing Mode: RPI, RH3, IPv6-in-IPv6 Encapsulation

8.1. Non-Storing Mode: Interaction between Leaf and Root

This section describes the communication flow in Non-Storing mode (Non-SM) between the following:

RAL to root

root to RAL

RUL to root

root to RUL

8.1.1. Non-SM: Example of Flow from RAL to Root

In Non-Storing mode, the leaf node uses default routing to send traffic to the root. The RPI must be included since it contains the Rank information, which is used to avoid and/or detect loops.

RAL (6LN) --> 6LR_i --> root(6LBR)

For example, a communication flow could be: Node F --> Node D --> Node B --> Node A (root)

6LR_i represents the intermediate routers from the source to the destination. In this case, 1 <= i <= n, where n is the total number of routers (6LR) that the packet goes through, from the source (RAL) to the destination (6LBR).

This situation is the same case as Storing mode.

Table 20 summarizes which headers are needed for this use case.

Header	RAL src	6LR_i	6LBR dst
Added headers	RPI	--	--
Modified headers	--	RPI	--
Removed headers	--	--	RPI
Untouched headers	--	--	--

Table 20: Non-SM: Summary of the Use of Headers from RAL to Root

8.1.2. Non-SM: Example of Flow from Root to RAL

In this case, the flow comprises:

root (6LBR) --> 6LR_i --> RAL (6LN)

For example, a communication flow could be: Node A (root) --> Node B --> Node D --> Node F

6LR_i represents the intermediate routers from the source to the destination. In this case, 1 <= i <= n, where n is the total number of routers (6LR) that the packet goes through, from the source (6LBR) to the destination (RAL).

The 6LBR inserts an RH3 and an RPI. No IPv6-in-IPv6 header is necessary as the traffic originates with a RPL-aware node, the 6LBR. The destination is known to be RPL aware because the root knows the whole topology in Non-Storing mode.

Table 21 summarizes which headers are needed for this use case.

Header	6LBR src	6LR_i	RAL dst
Added headers	RPI, RH3	--	--
Modified headers	--	RPI, RH3	--
Removed headers	--	--	RPI, RH3
Untouched headers	--	--	--

Table 21: Non-SM: Summary of the Use of Headers from Root to RAL

8.1.3. Non-SM: Example of Flow from Root to RUL

In this case, the flow comprises:

root (6LBR) --> 6LR_i --> RUL (IPv6 dst node)

For example, a communication flow could be: Node A (root) --> Node B --> Node E --> Node G (RUL)

6LR_i represents the intermediate routers from the source to the destination. In this case, 1 <= i <= n, where n is the total number of routers (6LR) that the packet goes through, from the source (6LBR) to the destination (RUL).

In the 6LBR, the RH3 is added; it is then modified at each intermediate 6LR (6LR_1 and so on), and it is fully consumed in the last 6LR (6LR_n) but is left in place. When the RPI is added, the RUL, which does not understand the RPI, will ignore it (per [RFC8200]); thus, encapsulation is not necessary.

Table 22 summarizes which headers are needed for this use case.

Header	6LBR src	6LR_i i=(1,..,n-1)	6LR_n	RUL dst
Added headers	RPI, RH3	--	--	--
Modified headers	--	RPI, RH3	RPI, RH3(consumed)	--
Removed headers	--	--	--	--
Untouched headers	--	--	--	RPI, RH3 (both ignored)

Table 22: Non-SM: Summary of the Use of Headers from Root to RUL

8.1.4. Non-SM: Example of Flow from RUL to Root

In this case, the flow comprises:

RUL (IPv6 src node) --> 6LR_1 --> 6LR_i --> root (6LBR) dst

For example, a communication flow could be: Node G --> Node E --> Node B --> Node A (root)

6LR_i represents the intermediate routers from the source to the destination. In this case, 1 <= i <= n, where n is the total number

of routers (6LR) that the packet goes through, from the source (RUL) to the destination (6LBR). For example, 6LR_1 (i=1) is the router that receives the packets from the RUL.

In this case, the RPI is added by the first 6LR (6LR_1) (Node E), encapsulated in an IPv6-in-IPv6 header, and modified in the subsequent 6LRs in the flow. The RPI and the entire packet are consumed by the root.

Table 23 summarizes which headers are needed for this use case.

Header	RUL src	6LR_1	6LR_i	6LBR dst
Added headers	--	IPv6-in-IPv6 (RPI)	--	--
Modified headers	--	--	RPI	--
Removed headers	--	--	--	IPv6-in-IPv6 (RPI)
Untouched headers	--	--	--	--

Table 23: Non-SM: Summary of the Use of Headers from RUL to Root

8.2. Non-Storing Mode: Interaction between Leaf and Internet

This section describes the communication flow in Non-Storing mode (Non-SM) between the following:

RAL to Internet

Internet to RAL

RUL to Internet

Internet to RUL

8.2.1. Non-SM: Example of Flow from RAL to Internet

In this case, the flow comprises:

RAL (6LN) src --> 6LR_i --> root (6LBR) --> Internet dst

For example, a communication flow could be: Node F (RAL) --> Node D --> Node B --> Node A --> Internet. Having the RAL information about the RPL domain, the packet may be encapsulated to the root when the destination is not in the RPL domain of the RAL.

6LR_i represents the intermediate routers from the source to the destination, and 1 <= i <= n, where n is the total number of routers (6LR) that the packet goes through, from the source (RAL) to the

6LBR.

In this case, the encapsulation from the RAL to the root is optional. The simplest case is when the RPI gets to the Internet (as the Table 24 shows it), knowing that the Internet is going to ignore it.

The IPv6 flow label should be set to zero to aid in compression [RFC8138], and the 6LBR will set it to a non-zero value when sending towards the Internet [RFC6437].

Table 24 summarizes which headers are needed for this use case when no encapsulation is used. Table 25 summarizes which headers are needed for this use case when encapsulation to the root is used.

Header	RAL src	6LR_i	6LBR	Internet dst
Added headers	RPI	--	--	--
Modified headers	--	RPI	RPI	--
Removed headers	--	--	--	--
Untouched headers	--	--	--	RPI (Ignored)

Table 24: Non-SM: Summary of the Use of Headers from RAL to Internet with No Encapsulation

Header	RAL src	6LR_i	6LBR	Internet dst
Added headers	IP6v6-in-IPv6 (RPI)	--	--	--
Modified headers	--	RPI	--	--
Removed headers	--	--	IPv6-in-IPv6 (RPI)	--
Untouched headers	--	--	--	--

Table 25: Non-SM: Summary of the Use of Headers from RAL to Internet with Encapsulation to the Root

8.2.2. Non-SM: Example of Flow from Internet to RAL

In this case, the flow comprises:

Internet --> root (6LBR) --> 6LR_i --> RAL dst (6LN)

For example, a communication flow could be: Internet --> Node A (root) --> Node B --> Node D --> Node F (RAL)

6LR_i represents the intermediate routers from source to destination, and 1 <= i <= n, where n is the total number of routers (6LR) that the packet goes through, from the 6LBR to the destination (RAL).

The 6LBR must add an RH3 header. As the 6LBR will know the path and address of the target node, it can address the IPv6-in-IPv6 header to that node. The 6LBR will zero the flow label upon entry in order to aid compression [RFC8138].

Table 26 summarizes which headers are needed for this use case.

Header	Internet src	6LBR	6LR_i	RAL dst
Added headers	--	IPv6-in-IPv6 (RH3, RPI)	--	--
Modified headers	--	--	IPv6-in-IPv6 (RH3, RPI)	--
Removed headers	--	--	--	IPv6-in-IPv6 (RH3, RPI)
Untouched headers	--	--	--	--

Table 26: Non-SM: Summary of the Use of Headers from Internet to RAL

8.2.3. Non-SM: Example of Flow from RUL to Internet

In this case, the flow comprises:

RUL (IPv6 src node) --> 6LR_1 --> 6LR_i --> root (6LBR) --> Internet dst

For example, a communication flow could be: Node G --> Node E --> Node B --> Node A --> Internet

6LR_i represents the intermediate routers from the source to the destination, and 1 <= i <= n, where n is the total number of routers (6LRs) that the packet goes through, from the source (RUL) to the 6LBR, e.g., 6LR_1 (i=1).

In this case, the flow label is recommended to be zero in the RUL. As the RUL parent adds RPL headers in the RUL packet, the first 6LR (6LR_1) will add an RPI inside a new IPv6-in-IPv6 header. The IPv6-in-IPv6 header will be addressed to the root. This case is identical to the Storing mode case (see Section 7.2.3).

Table 27 summarizes which headers are needed for this use case.

Header	RUL src	6LR_1	6LR_i	6LBR	Internet

			i=(2,..,n)		dst
Added headers	--	IP6-IP6 (RPI)	--	--	--
Modified headers	--	--	RPI	--	--
Removed headers	--	--	--	IP6-IP6 (RPI)	--
Untouched headers	--	--	--	--	--

Table 27: Non-SM: Summary of the Use of Headers from RUL to Internet

8.2.4. Non-SM: Example of Flow from Internet to RUL

In this case, the flow comprises:

Internet src --> root (6LBR) --> 6LR_i --> RUL (IPv6 dst node)

For example, a communication flow could be: Internet --> Node A (root) --> Node B --> Node E --> Node G

6LR_i represents the intermediate routers from the source to the destination, and 1 <= i <= n, where n is the total number of routers (6LR) that the packet goes through, from the 6LBR to the RUL.

The 6LBR must add an RH3 header inside an IPv6-in-IPv6 header. The 6LBR will know the path and will recognize that the final node is not a RPL-capable node as it will have received the connectivity DAO from the nearest 6LR. The 6LBR can therefore make the IPv6-in-IPv6 header destination be the last 6LR. The 6LBR will set to zero the flow label upon entry in order to aid compression [RFC8138].

Table 28 summarizes which headers are needed for this use case.

Header	Internet src	6LBR	6LR_i	6LR_n	RUL dst
Added headers	--	IP6-IP6 (RH3, RPI)	--	--	--
Modified headers	--	--	IP6-IP6 (RH3, RPI)	--	--
Removed headers	--	--	--	IP6-IP6 (RH3, RPI)	--
Untouched headers	--	--	--	--	--

Table 28: Non-SM: Summary of the Use of Headers from Internet to RUL

8.3. Non-SM: Interaction between Leaves

This section describes the communication flow in Non-Storing mode (Non-SM) between the following:

- RAL to RAL
- RAL to RUL
- RUL to RAL
- RUL to RUL

8.3.1. Non-SM: Example of Flow from RAL to RAL

In this case, the flow comprises:

RAL src --> 6LR_ia --> root (6LBR) --> 6LR_id --> RAL dst

For example, a communication flow could be: Node F (RAL src) --> Node D --> Node B --> Node A (root) --> Node B --> Node E --> Node H (RAL dst)

6LR_ia represents the intermediate routers from the source to the root, and 1 <= ia <= n, where n is the total number of routers (6LR) that the packet goes through, from the RAL to the root.

6LR_id represents the intermediate routers from the root to the destination, and 1 <= id <= m, where m is the total number of the intermediate routers (6LR).

This case involves only nodes in same RPL domain. The originating node will add an RPI to the original packet and send the packet Upward.

The originating node may put the RPI (RPI1) into an IPv6-in-IPv6 header addressed to the root so that the 6LBR can remove that header. If it does not, then the RPI1 is forwarded down from the root in the inner header to no avail.

The 6LBR will need to insert an RH3 header, which requires that it add an IPv6-in-IPv6 header. It removes the RPI (RPI1), as it was contained in an IPv6-in-IPv6 header addressed to it. Otherwise, there may be an RPI buried inside the inner IP header, which should be ignored. The root inserts an RPI (RPI2) alongside the RH3.

Networks that use the RPL point-to-point extension [RFC6997] are essentially Non-Storing DODAGs and fall into this scenario or the scenario given in Section 8.1.2, with the originating node acting as a 6LBR.

Table 29 summarizes which headers are needed for this use case when

encapsulation to the root takes place.

Table 30 summarizes which headers are needed for this use case when there is no encapsulation to the root. Note that in the Modified headers row, going up in each 6LR_ia only the RPI1 is changed. Going down, in each 6LR_id the IPv6 header is swapped with the RH3 so both are changed alongside with the RPI2.

Header	RAL src	6LR_ia	6LBR	6LR_id	RAL dst
Added headers	IP6-IP6 (RPI1)	--	IP6-IP6 (RH3 -> RAL, RPI2)	--	--
Modified headers	--	RPI1	--	IP6-IP6 (RH3, RPI2)	--
Removed headers	--	--	IP6-IP6 (RPI1)	--	IP6-IP6 (RH3, RPI2)
Untouched headers	--	--	--	--	--

Table 29: Non-SM: Summary of the Use of Headers from RAL to RAL with Encapsulation to the Root

Header	RAL src	6LR_ia	6LBR	6LR_id	RAL dst
Added headers	RPI1	--	IP6-IP6 (RH3, RPI2)	--	--
Modified headers	--	RPI1	--	IP6-IP6 (RH3, RPI2)	--
Removed headers	--	--	--	--	IP6-IP6 (RH3, RPI2)
Untouched headers	--	--	RPI1	RPI1	RPI1 (Ignored)

Table 30: Non-SM: Summary of the Use of Headers from RAL to RAL without Encapsulation to the Root

8.3.2. Non-SM: Example of Flow from RAL to RUL

In this case, the flow comprises:

RAL --> 6LR_ia --> root (6LBR) --> 6LR_id --> RUL (IPv6 dst node)

For example, a communication flow could be: Node F (RAL) --> Node D --> Node B --> Node A (root) --> Node B --> Node E --> Node G (RUL)

6LR_ia represents the intermediate routers from the source to the root, and 1 <= ia <= n, where n is the total number of intermediate routers (6LR).

6LR_id represents the intermediate routers from the root to the destination, and 1 <= id <= m, where m is the total number of the intermediate routers (6LRs).

As in the previous case, the RAL (6LN) may insert an RPI (RPI1) header, which must be in an IPv6-in-IPv6 header addressed to the root so that the 6LBR can remove this RPI. The 6LBR will then insert an RH3 inside a new IPv6-in-IPv6 header addressed to the last 6LR_id (6LR_id = m) alongside the insertion of RPI2.

If the originating node does not put the RPI (RPI1) into an IPv6-in-IPv6 header addressed to the root, then the RPI1 is forwarded down from the root in the inner header to no avail.

Table 31 summarizes which headers are needed for this use case when encapsulation to the root takes place. Table 32 summarizes which headers are needed for this use case when no encapsulation to the root takes place.

Header	RAL src	6LR_ia	6LBR	6LR_id	6LR_m	RUL dst
Added headers	IP6-IP6 (RPI1)	--	IP6-IP6 (RH3, RPI2)	--	--	--
Modified headers	--	RPI1	--	IP6-IP6 (RH3, RPI2)	--	--
Removed headers	--	--	IP6-IP6 (RPI1)	--	IP6-IP6 (RH3, RPI2)	--
Untouched headers	--	--	--	--	--	--

Table 31: Non-SM: Summary of the Use of Headers from RAL to RUL with Encapsulation to the Root

Header	RAL src	6LR_ia	6LBR	6LR_id	6LR_n	RUL dst
Added headers	RPI1	--	IP6-IP6 (RH3,	--	--	--

			RPI2)			
Modified headers	--	RPI1	--	IP6-IP6 (RH3, RPI2)	--	--
Removed headers	--	--	--	--	IP6-IP6 (RH3, RPI2)	--
Untouched headers	--	--	RPI1	RPI1	RPI1	RPI1 (ignored)

Table 32: Non-SM: Summary of the Use of Headers from RAL to RUL without Encapsulation to the Root

8.3.3. Non-SM: Example of Flow from RUL to RAL

In this case, the flow comprises:

RUL (IPv6 src node) --> 6LR_1 --> 6LR_ia --> root (6LBR) --> 6LR_id --> RAL dst (6LN)

For example, a communication flow could be: Node G (RUL) --> Node E --> Node B --> Node A (root) --> Node B --> Node E --> Node H (RAL)

6LR_ia represents the intermediate routers from source to the root, and 1 <= ia <= n, where n is the total number of intermediate routers (6LR).

6LR_id represents the intermediate routers from the root to the destination, and 1 <= id <= m, where m is the total number of the intermediate routers (6LR).

In this scenario, the RPI (RPI1) is added by the first 6LR (6LR_1) inside an IPv6-in-IPv6 header addressed to the root. The 6LBR will remove this RPI and add its own IPv6-in-IPv6 header containing an RH3 header and an RPI (RPI2).

Table 33 summarizes which headers are needed for this use case.

Header	RUL src	6LR_1	6LR_ia	6LBR	6LR_id	RAL dst
Added headers	--	IP6-IP6 (RPI1)	--	IP6-IP6 (RH3, RPI2)	--	--
Modified headers	--	--	RPI1	--	IP6-IP6 (RH3, RPI2)	--
Removed headers	--	--	--	IP6-IP6 (RPI1)	--	IP6-IP6 (RH3,

						RPI2)
Untouched headers	--	--	--	--	--	--

Table 33: Non-SM: Summary of the Use of Headers from RUL to RAL

8.3.4. Non-SM: Example of Flow from RUL to RUL

In this case, the flow comprises:

RUL (IPv6 src node) --> 6LR_1 --> 6LR_ia --> root (6LBR) --> 6LR_id --> RUL (IPv6 dst node)

For example, a communication flow could be: Node G --> Node E --> Node B --> Node A (root) --> Node C --> Node J

6LR_ia represents the intermediate routers from the source to the root, and 1 <= ia <= n, where n is the total number of intermediate routers (6LR).

6LR_id represents the intermediate routers from the root to the destination, and 1 <= id <= m, where m is the total number of the intermediate routers (6LR).

This scenario is the combination of the previous two cases.

Table 34 summarizes which headers are needed for this use case.

Header	RUL src	6LR_1	6LR_ia	6LBR	6LR_id	6LR_m	RUL dst
Added headers	--	IP6-IP6 (RPI1)	--	IP6-IP6 (RH3, RPI2)	--	--	--
Modified headers	--	--	RPI1	--	IP6-IP6 (RH3, RPI2)	--	--
Removed headers	--	--	--	IP6-IP6 (RPI1)	--	IP6-IP6 (RH3, RPI2)	--
Untouched headers	--	--	--	--	--	--	--

Table 34: Non-SM: Summary of the Use of Headers from RUL to RUL

9. Operational Considerations of Supporting RULs

Roughly half of the situations described in this document involve leaf ("host") nodes that do not speak RPL. These nodes fall into two

further categories: ones that drop a packet that have RPI or RH3 headers, and ones that continue to process a packet that has RPI and/or RH3 headers.

[RFC8200] provides for new rules that suggest that nodes that have not been configured (explicitly) to examine Hop-by-Hop Options headers should ignore those headers and continue processing the packet. Despite this, and despite the switch from 0x63 to 0x23, there may be nodes that predate RFC 8200 or are simply intolerant. Those nodes will drop packets that continue to have RPL artifacts in them. In general, such nodes cannot be easily supported in RPL LLNs.

There are some specific cases where it is possible to remove the RPL artifacts prior to forwarding the packet to the leaf host. The critical thing is that the artifacts have been inserted by the RPL root inside an IPv6-in-IPv6 header, and that the header has been addressed to the 6LR immediately prior to the leaf node. In that case, in the process of removing the IPv6-in-IPv6 header, the artifacts can also be removed.

The above case occurs whenever traffic originates from the outside the LLN (the "Internet" cases above), and Non-Storing mode is used. In Non-Storing mode, the RPL root knows the exact topology (as it must create the RH3 header) and therefore knows which 6LR is prior to the leaf. For example, in Figure 3, Node E is the 6LR prior to leaf Node G, or Node C is the 6LR prior to leaf Node J.

Traffic originating from the RPL root (such as when the data collection system is co-located on the RPL root), does not require an IPv6-in-IPv6 header (in Storing or Non-Storing mode), as the packet is originating at the root, and the root can insert the RPI and RH3 headers directly into the packet as it is formed. Such a packet is slightly smaller, but can only be sent to nodes (whether RPL aware or not) that will tolerate the RPL artifacts.

An operator that finds itself with a high amount of traffic from the RPL root to RPL-unaware leaves will have to do IPv6-in-IPv6 encapsulation if the leaf is not tolerant of the RPL artifacts. Such an operator could otherwise omit this unnecessary header if it was certain of the properties of the leaf.

As the Storing mode cannot know the final path of the traffic, intolerant leaf nodes, which drop packets with RPL artifacts, cannot be supported.

10. Operational Considerations of Introducing 0x23

This section describes the operational considerations of introducing the new RPI Option Type of 0x23.

During bootstrapping, the node receives the DIO with the information of RPI Option Type, indicating the new RPI in the DODAG Configuration option flag. The DODAG root is in charge of configuring the current network with the new value, through DIO messages, and determining when all the nodes have been set with the new value. The DODAG should change to a new DODAG version. In case of rebooting, the node

does not remember the RPI Option Type. Thus, the DIO is sent with a flag indicating the new RPI Option Type.

The DODAG Configuration option is contained in a RPL DIO message, which contains a unique Destination Advertisement Trigger Sequence Number (DTSN) counter. The leaf nodes respond to this message with DAO messages containing the same DTSN. This is a normal part of RPL routing; the RPL root therefore knows when the updated DODAG Configuration option has been seen by all nodes.

Before the migration happens, all the RPL-aware nodes should support both values. The migration procedure is triggered when the DIO is sent with the flag indicating the new RPI Option Type. Namely, it remains at 0x63 until it is sure that the network is capable of 0x23, then it abruptly changes to 0x23. The 0x23 RPI Option allows the sending of packets to non-RPL nodes. The non-RPL nodes should ignore the option and continue processing the packets.

As mentioned previously, indicating the new RPI in the DODAG Configuration option flag is a way to avoid the flag day (abrupt changeover) in a network using 0x63 as the RPI Option Type value. It is suggested that RPL implementations accept both 0x63 and 0x23 RPI Option Type values when processing the header to enable interoperability.

11. IANA Considerations

11.1. Option Type in RPL Option

This document updates the registration made in the "Destination Options and Hop-by-Hop Options" subregistry [RFC6553] from 0x63 to 0x23 as shown in Table 35.

Hex Value	Binary Value			Description	Reference
	act	chg	rest		
0x23	00	1	00011	RPL Option	This document
0x63	01	1	00011	RPL Option (DEPRECATED)	[RFC6553], this document

Table 35: Option Type in RPL Option

The "DODAG Configuration Option Flags for MOP 0..6" subregistry is updated as follows (Table 36):

Bit Number	Capability Description	Reference
3	RPI 0x23 enable	This document

Table 36: DODAG Configuration Option Flag to

11.2. Change to the "DODAG Configuration Option Flags" Subregistry

IANA has changed the name of the "DODAG Configuration Option Flags" subregistry to "DODAG Configuration Option Flags for MOP 0..6".

The subregistry references this document for this change.

11.3. Change MOP Value 7 to Reserved

IANA has changed the registration status of value 7 in the "Mode of Operation" subregistry from Unassigned to Reserved. This change is in support of future work.

This document is listed as a reference for this entry in the subregistry.

12. Security Considerations

The security considerations covered in [RFC6553] and [RFC6554] apply when the packets are in the RPL Domain.

The IPv6-in-IPv6 mechanism described in this document is much more limited than the general mechanism described in [RFC2473]. The willingness of each node in the LLN to decapsulate packets and forward them could be exploited by nodes to disguise the origin of an attack.

While a typical LLN may be a very poor origin for attack traffic (as the networks tend to be very slow, and the nodes often have very low duty cycles), given enough nodes, LLNs could still have a significant impact, particularly if the attack is targeting another LLN. Additionally, some uses of RPL involve large-backbone, ISP-scale equipment [ACP], which may be equipped with multiple 100 Gb/s interfaces.

Blocking or careful filtering of IPv6-in-IPv6 traffic entering the LLN as described above will make sure that any attack that is mounted must originate from compromised nodes within the LLN. The use of network ingress filtering [BCP38] on egress traffic at the RPL root will alert the operator to the existence of the attack as well as drop the attack traffic. As the RPL network is typically numbered from a single prefix, which is itself assigned by RPL, network ingress filtering [BCP38] involves a single prefix comparison and should be trivial to automatically configure.

There are some scenarios where IPv6-in-IPv6 traffic should be allowed to pass through the RPL root, such as the IPv6-in-IPv6 mediated communications between a new pledge and the Join Registrar/ Coordinator (JRC) when using [BRSKI] and [ZEROTOUCH-JOIN]. This is the case for the RPL root to do careful filtering: it occurs only when the Join Coordinator is not co-located inside the RPL root.

With the above precautions, an attack using IPv6-in-IPv6 tunnels can only be by a node within the LLN on another node within the LLN.

Such an attack could, of course, be done directly. An attack of this kind is meaningful only if the source addresses are either fake or if the point is to amplify return traffic. Such an attack could also be done without the use of IPv6-in-IPv6 headers, by using forged source addresses instead. If the attack requires bidirectional communication, then IPv6-in-IPv6 provides no advantages.

Whenever IPv6-in-IPv6 headers are being proposed, there is a concern about creating security issues. In the Security Considerations section of [RFC2473] (Section 9), it was suggested that tunnel entry and exit points can be secured by securing the IPv6 path between them. This recommendation is not practical for RPL networks. [RFC5406] provides guidance on what on what additional details are needed in order to "Use IPsec". While the use of Encapsulating Security Payload (ESP) would prevent source address forgeries, in order to use it with [RFC8138], compression would have to occur before encryption, as the [RFC8138] compression is lossy. Once encrypted, there would be no further redundancy to compress. These are minor issues. The major issue is how to establish trust enough such that Internet Key Exchange Protocol Version 2 (IKEv2) could be used. This would require a system of certificates to be present in every single node, including any Internet nodes that might need to communicate with the LLN. Thus, using IPsec requires a global PKI in the general case.

More significantly, the use of IPsec tunnels to protect the IPv6-in-IPv6 headers would, in the general case, scale with the square of the number of nodes. This is a lot of resources for a constrained nodes on a constrained network. In the end, the IPsec tunnels would be providing only BCP38-like origin authentication! That is, IPsec provides a transitive guarantee to the tunnel exit point that the tunnel entry point did network ingress filtering [BCP38] on traffic going in. Just doing origin filtering per BCP 38 at the entry and exit of the LLN provides a similar level of security without all the scaling and trust problems related to IPv6 tunnels as discussed in [RFC2473]. IPsec is not recommended.

An LLN with hostile nodes within it would not be protected against impersonation within the LLN by entry/exit filtering.

The RH3 header usage described here can be abused in equivalent ways. An external attacker may form a packet with an RH3 that is not fully consumed and encapsulate it to hide the RH3 from intermediate nodes and disguise the origin of traffic. As such, the attacker's RH3 header will not be seen by the network until it reaches the destination, which will decapsulate it. As indicated in Section 4.2 of [RFC6554], RPL routers are responsible for ensuring that an SRH is only used between RPL routers. As such, if there is an RH3 that is not fully consumed in the encapsulated packet, the node that decapsulates it MUST ensure that the outer packet was originated in the RPL domain and drop the packet otherwise.

Also, as indicated by Section 2 of [RFC6554], RPL Border Routers "do not allow datagrams carrying an SRH header to enter or exit a RPL routing domain." This sentence must be understood as concerning non-fully-consumed packets. A consumed (inert) RH3 header could be

present in a packet that flows from one LLN, crosses the Internet, and enters another LLN. Per the discussion in this document, such headers do not need to be removed. However, there is no case described in this document where an RH3 is inserted in a Non-Storing network on traffic that is leaving the LLN, but this document should not preclude such a future innovation.

In short, a packet that crosses the border of the RPL domain MAY carry an RH3, and if so, that RH3 MUST be fully consumed.

The RPI, if permitted to enter the LLN, could be used by an attacker to change the priority of a packet by selecting a different RPLInstanceID, perhaps one with a higher energy cost, for instance. It could also be that not all nodes are reachable in an LLN using the default RPLInstanceID, but a change of RPLInstanceID would permit an attacker to bypass such filtering. Like the RH3, an RPI is to be inserted by the RPL root on traffic entering the LLN by first inserting an IPv6-in-IPv6 header. The attacker's RPI therefore will not be seen by the network. Upon reaching the destination node, the RPI has no further meaning and is just skipped; the presence of a second RPI will have no meaning to the end node as the packet has already been identified as being at its final destination.

For traffic leaving a RUL, if the RUL adds an uninitialized RPI (e.g., with a value of zero), then the 6LR as a RPL Border Router SHOULD rewrite the RPI to indicate the selected Instance and set the flags. This is done in order to avoid the following scenarios: 1) The leaf is an external router that passes a packet that it did not generate and that carries an unrelated RPI, and 2) The leaf is an attacker or presents misconfiguration and tries to inject traffic in a protected Instance. Also, this applies to the case where the leaf is aware of the RPL Instance and passes a correct RPI; the 6LR needs a configuration that allows that leaf to inject in that instance.

The RH3 and RPIs could be abused by an attacker inside of the network to route packets in nonobvious ways, perhaps eluding observation. This usage appears consistent with a normal operation of [RFC6997] and cannot be restricted at all. This is a feature, not a bug.

[RFC7416] deals with many other threats to LLNs not directly related to the use of IPv6-in-IPv6 headers, and this document does not change that analysis.

Nodes within the LLN can use the IPv6-in-IPv6 mechanism to mount an attack on another part of the LLN, while disguising the origin of the attack. The mechanism can even be abused to make it appear that the attack is coming from outside the LLN, and unless countered, this could be used to mount a DDOS attack upon nodes elsewhere in the Internet. See [DDOS-KREBS] for an example of such attacks already seen in the real world.

If an attack comes from inside of LLN, it can be alleviated with SAVI (Source Address Validation Improvement) using [RFC8505] with [RFC8928]. The attacker will not be able to source traffic with an address that is not registered, and the registration process checks for topological correctness. Notice that there is Layer 2

authentication in most of the cases. If an attack comes from outside LLN, IPv6-in-IPv6 can be used to hide inner routing headers, but by construction, the RH3 can typically only address nodes within the LLN. That is, an RH3 with a CmprI less than 8 should be considered an attack (see Section 3 of [RFC6554]).

Nodes outside of the LLN will need to pass IPv6-in-IPv6 traffic through the RPL root to perform this attack. To counter, the RPL root SHOULD either restrict ingress of IPv6-in-IPv6 packets (the simpler solution), or it SHOULD walk the IP header extension chain until it can inspect the upper-layer payload as described in [RFC7045]. In particular, the RPL root SHOULD do network ingress filtering [BCP38] on the source addresses of all IP headers that it examines in both directions.

Note: there are some situations where a prefix will spread across multiple LLNs via mechanisms such as the one described in [RFC8929]. In this case, the network ingress filtering [BCP38] needs to take this into account, either by exchanging detailed routing information on each LLN or by moving the network ingress filtering [BCP38] further towards the Internet, so that the details of the multiple LLNs do not matter.

13. References

13.1. Normative References

[BCP38] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

<https://rfc-editor.org/info/bcp38>

[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>.

[RFC6040] Briscoe, B., "Tunnelling of Explicit Congestion Notification", RFC 6040, DOI 10.17487/RFC6040, November 2010, <https://www.rfc-editor.org/info/rfc6040>.

[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>.

[RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <https://www.rfc-editor.org/info/rfc6550>.

[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>.

[RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing of IPv6 Extension Headers", RFC 7045, DOI 10.17487/RFC7045, December 2013, <https://www.rfc-editor.org/info/rfc7045>.

[RFC8025] Thubert, P., Ed. and R. Cragie, "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Paging Dispatch", RFC 8025, DOI 10.17487/RFC8025, November 2016, <https://www.rfc-editor.org/info/rfc8025>.

[RFC8138] Thubert, P., Ed., Bormann, C., Toutain, L., and R. Cragie, "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing Header", RFC 8138, DOI 10.17487/RFC8138, April 2017, <https://www.rfc-editor.org/info/rfc8138>.

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

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

13.2. Informative References

[ACP] Eckert, T., Behringer, M. H., and S. Bjarnason, "An Autonomic Control Plane (ACP)", Work in Progress, Internet-Draft, draft-ietf-anima-autonomic-control-plane-30, 30 October 2020, <https://tools.ietf.org/html/draft-ietf-anima-autonomic-control-plane-30>.

[BRSKI] Pritikin, M., Richardson, M. C., Eckert, T., Behringer, M. H., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructures (BRSKI)", Work in Progress, Internet-Draft, draft-ietf-anima-bootstrapping-keyinfra-45, 11 November 2020, <https://tools.ietf.org/html/draft-ietf-anima-bootstrapping-keyinfra-45>.

[DDOS-KREBS] Goodin, D., "Record-breaking DDoS reportedly delivered by >145k hacked cameras", September 2016, <https://arstechnica.com/information-technology/2016/09/botnet-of-145k-cameras-reportedly-deliver-internets-biggest-ddos-ever/>.

[RFC0801] Postel, J., "NCP/TCP transition plan", RFC 801, DOI 10.17487/RFC0801, November 1981, <https://www.rfc-editor.org/info/rfc801>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <https://www.rfc-editor.org/info/rfc2460>.

[RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473, December 1998, <https://www.rfc-editor.org/info/rfc2473>.

[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>.

[RFC5406] Bellovin, S., "Guidelines for Specifying the Use of IPsec Version 2", BCP 146, RFC 5406, DOI 10.17487/RFC5406, February 2009, <https://www.rfc-editor.org/info/rfc5406>.

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <https://www.rfc-editor.org/info/rfc6437>.

[RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <https://www.rfc-editor.org/info/rfc6775>.

[RFC6997] Goyal, M., Ed., Baccelli, E., Philipp, M., Brandt, A., and J. Martocci, "Reactive Discovery of Point-to-Point Routes in Low-Power and Lossy Networks", RFC 6997, DOI 10.17487/RFC6997, August 2013, <https://www.rfc-editor.org/info/rfc6997>.

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

[RFC7416] Tsao, T., Alexander, R., Dohler, M., Daza, V., Lozano, A., and M. Richardson, Ed., "A Security Threat Analysis for the Routing Protocol for Low-Power and Lossy Networks (RPLs)", RFC 7416, DOI 10.17487/RFC7416, January 2015, <https://www.rfc-editor.org/info/rfc7416>.

[RFC8180] Vilajosana, X., Ed., Pister, K., and T. Watteyne, "Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) Configuration", BCP 210, RFC 8180, DOI 10.17487/RFC8180, May 2017, <https://www.rfc-editor.org/info/rfc8180>.

[RFC8504] Chown, T., Loughney, J., and T. Winters, "IPv6 Node Requirements", BCP 220, RFC 8504, DOI 10.17487/RFC8504,

January 2019, <https://www.rfc-editor.org/info/rfc8504>.

[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>.

[RFC8928] Thubert, P., Ed., Sarikaya, B., Sethi, M., and R. Struik, "Address-Protected Neighbor Discovery for Low-Power and Lossy Networks", RFC 8928, DOI 10.17487/RFC8928, November 2020, <https://www.rfc-editor.org/info/rfc8928>.

[RFC8929] Thubert, P., Ed., Perkins, C.E., and E. Levy-Abegnoli, "IPv6 Backbone Router", RFC 8929, DOI 10.17487/RFC8929, November 2020, <https://www.rfc-editor.org/info/rfc8929>.

[RFC9010] Thubert, P., Ed. and M. Richardson, "Routing for RPL (Routing Protocol for Low-Power and Lossy Networks) Leaves", RFC 9010, DOI 10.17487/RFC9010, April 2021, <https://www.rfc-editor.org/rfc/rfc9010>.

[TUNNELS] Touch, J. and M. Townsley, "IP Tunnels in the Internet Architecture", Work in Progress, Internet-Draft, draft-ietf-intarea-tunnels-10, 12 September 2019, <https://tools.ietf.org/html/draft-ietf-intarea-tunnels-10>.

[ZEROTOUCH-JOIN]
Richardson, M., "6tisch Zero-Touch Secure Join protocol", Work in Progress, Internet-Draft, draft-ietf-6tisch-dtsecurity-zerotouch-join-04, 8 July 2019, <https://tools.ietf.org/html/draft-ietf-6tisch-dtsecurity-zerotouch-join-04>.

Acknowledgments

This work is done thanks to the grant given by the StandICT.eu project.

A special BIG thanks to C. M. Heard for the help with Section 4. Much of the editing in that section is based on his comments.

Additionally, the authors would like to acknowledge the review, feedback, and comments of the following (in alphabetical order): Dominique Barthel, Robert Cragie, Ralph Droms, Simon Duquennoy, Cenk Guendogan, Rahul Jadhav, Benjamin Kaduk, Matthias Kovatsch, Gustavo Mercado, Subramanian Moonesamy, Marcela Orbiscay, Cristian Perez, Charlie Perkins, Alvaro Retana, Peter van der Stok, Xavier Vilajosana, Éric Vyncke, and Thomas Watteyne.

Authors' Addresses

Maria Ines Robles
Universidad Tecno. Nac.(UTN)-FRM, Argentina /Aalto University Finland
Coronel Rodríguez 273

M5500 Mendoza
Provincia de Mendoza
Argentina

Email: mariainesrobles@gmail.com

Michael C. Richardson
Sandelman Software Works
470 Dawson Avenue
Ottawa ON K1Z 5V7
Canada

Email: mcr+ietf@sandelman.ca
URI: http://www.sandelman.ca/mcr/

Pascal Thubert
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