ROLL Working Group
Internet-Draft
Intended status: Informational
Expires: October 7, 2016

M. Robles
Ericsson
M. Richardson
SSW
P. Thubert
Cisco

April 5, 2016

When to use RFC 6553, 6554 and IPv6-in-IPv6 draft-ietf-roll-useofrplinfo-04

Abstract

This document looks at different data flows through LLN networks where RPL is used to establish routing. The document enumerates the cases where RFC 6553, RFC 6554 and IPv6-in-IPv6 encapsulation is required. This analysis provides the basis on which to design efficient compression of these headers.

Status of This Memo

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

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

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

This Internet-Draft will expire on October 7, 2016.

Table of Contents

1. Introduction

RPL [RFC6550] is a routing protocol for constrained networks. RFC 6553 [RFC6553] defines the "RPL option" (RPI), carried within the IPv6 Hop-by-Hop header to quickly identify inconsistencies (loops) in the routing topology. 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 traffic at all which is mostly hop-by-hop traffic (one exception being 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 implementors agree when artifacts are necessary, or when they can be safely omitted, or removed.

An interim meeting went through the 24 cases defined here to discover if there were any shortcuts, and this document is the result of that discussion. This document should not be defining anything new, but it may clarify what is correct and incorrect behaviour.

The related document A Routing Header Dispatch for 6LoWPAN (6LoRH) [I-D.ietf-roll-routing-dispatch] defines a method to compress RPL Option information and Routing Header type 3 (RFC6554) and an efficient IP-in-IP technique. Uses cases proposed for the [Second6TischPlugtest] involving 6loRH.

2. Terminology and Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Terminology defined in [RFC7102]

3. Sample/reference topology

A RPL network is composed of a 6LBR (6LoWPAN Border Router), Backbone Router (6BBR), 6LR (6LoWPAN Router) and 6LN (6LoWPAN Node) as leaf logically organized in a DODAG structure (Destination Oriented Directed Acyclic Graph).

RPL defines the RPL Control messages (control plane), a new ICMPv6 message with Type 155. DIS, DIO and DAO messages are all RPL Control messages but with different Code values.

RPL supports two modes of Downward traffic: in storing mode, it is fully stateful or an in non-storing, 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 storing and non-storing nodes is not supported with the current specifications.

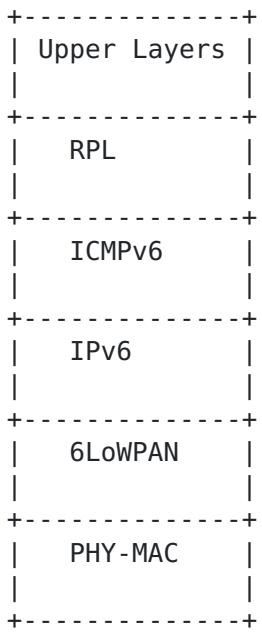

Figure 1: RPL Stack.

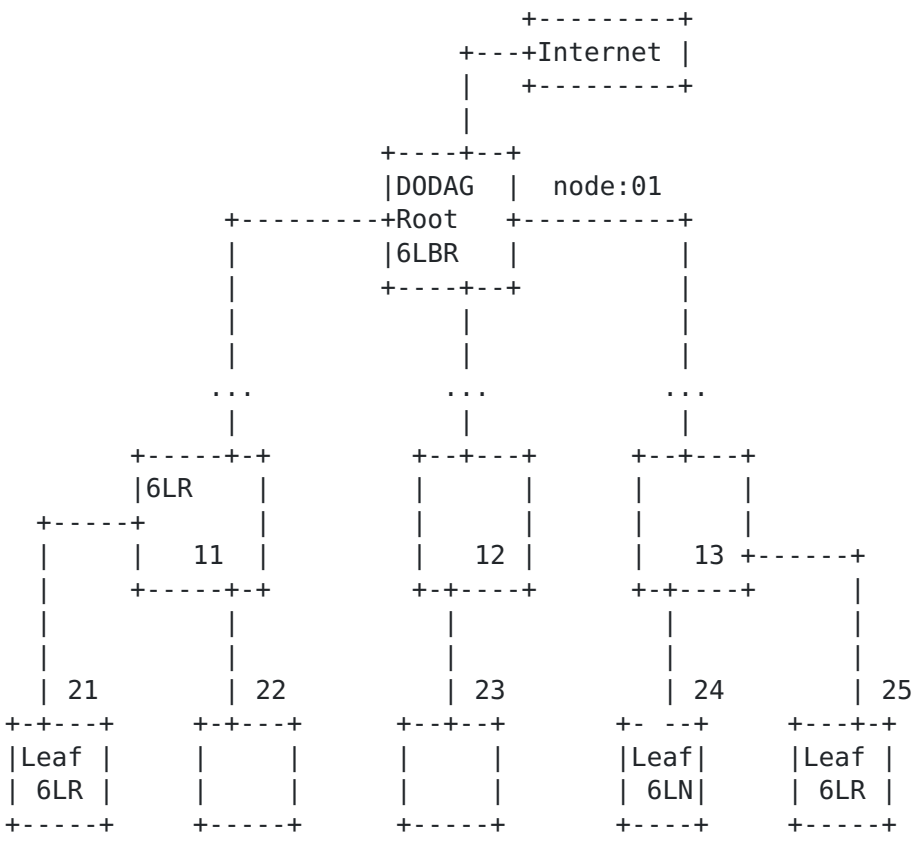

Figure 2: A reference RPL Topology.

The numbers in or above the nodes are there so that they may be referenced in subsequent sections. The leaf marked 6LN (24) is a device which does not speak RPL at all, but uses Router-Advertisements, 6LowPAN DAR/DAC and efficient-ND only to participate in the network.

This document is in part motivated by the work that is ongoing at the 6TiSCH working group. The 6TiSCH architecture [I-D.ietf-6tisch-architecture] draft explains the network architecture of a 6TiSCH network. This architecture is used for the remainder of this document.

The scope of the 6TiSCH Architecture is a Backbone Link that federates multiple LLNs (mesh) as a single IPv6 Multi-Link Subnet. Each LLN in the subnet is anchored at a Backbone Router (6BBR). The Backbone Routers interconnect the LLNs over the Backbone Link and emulate that the LLN nodes are present on the Backbone thus creating a so-called: Multi-Link Subnet. An LLN node can move freely from an LLN anchored at a Backbone Router to another LLN anchored at the same

or a different Backbone Router inside the Multi-Link Subnet and conserve its addresses.

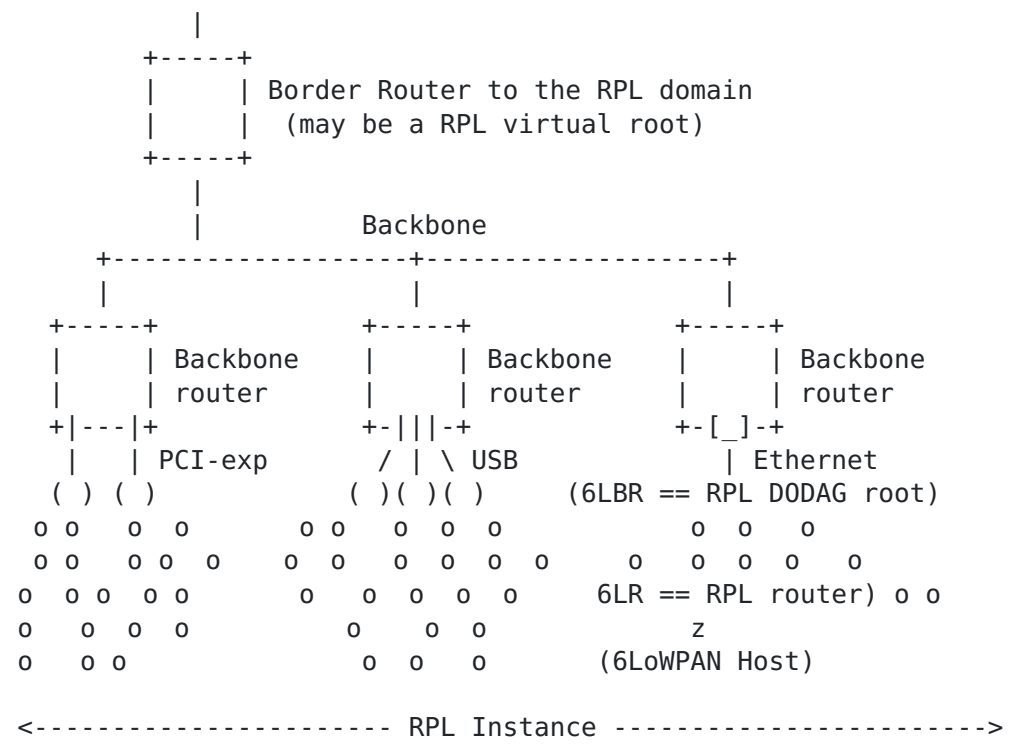

Figure 3: RPL domain architecture

4. Use cases

In data plane context a combination of RFC6553, RFC6554 and IPv6-in-IPv6 encapsulation is going to be analyzed for the following traffic flows:

RPL-aware-leaf to root

root to RPL-aware-leaf

not-RPL-aware-leaf to root

root to not-RPL-aware-leaf

RPL-aware-leaf to Internet

Internet to RPL-aware-leaf

not-RPL-aware-leaf to Internet

Internet to not-RPL-aware-leaf

RPL-aware-leaf to RPL-aware-leaf

RPL-aware-leaf to not-RPL-aware-leaf

not-RPL-aware-leaf to RPL-aware-leaf

not-RPL-aware-leaf to not-RPL-aware-leaf

This document assumes a 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 [RFC2460] is that Extensions may not be added or removed except by the sender or the receiver.

A second important thing is that packets with a Hop-by-Hop option which are marked with option type 01 ([RFC2460] section 4.2) must be discarded if received by a host or router which does not understand that option. This means that in general, any packet that leaves the RPL domain of an LLN (or leaves the LLN entirely) is likely to be discarded if it still contains an [RFC6553] RPL Option Header known as the RPI.

The combination of these two rules means that the arrangement of headers must be done so that traffic intended to exit the RPL domain can have the RPI option removed prior to leaving the RPL domain.

An intermediate router that needs to add a header must encapsulate the packet in an (additional) outer IP header where the new header can be placed.

This also means that a Header can only be removed by an intermediate router if it is placed in an encapsulating IPv6 Header, and in that case, the whole encapsulating header must be removed - a replacement may be added. Further, an intermediate router can only remove such an outer header if that outer header has the router as the destination!

Both RPI and RH3 headers may be modified by routers on the path of the packet without the need to add to remove an encapsulating header. Both headers were designed with this modification in mind, and both the RPL RH and the RPL option are marked mutable but recoverable, so an IPsec AH security header can be applied across these headers, but it may not secure all the values in those headers.

RPI should be present in every single RPL data packet. There is one exception in non-storing mode: when a packet is going down from the route. In a downward non-storing mode, the entire route is written, so there can be no loops by construction, nor any confusion about which forwarding table to use. There may be cases (such as in 6tisch) where the instanceID may still be needed to pick an appropriate priority or channel at each hop.

The applicability for storing (RPL-SN) and non-Storing (RPL-NSN) modes for the previous cases is showed as follows:

In tables, the term "RPL aware leaf" is has been shortened to "Raf", and "not-RPL aware leaf" has been shortened to "~Raf" to make the table fit in available space.

The earlier examples are more complete to make sure that the process is clear, while later examples are more consise.

5. Storing mode

This table summarizes what headers are needed in the following scenarios, and indicates the IPIP header must be inserted on a hop-by-hop basis, and when it can target the destination node directly. There are three possible situations: hop-by-hop necessary (indicated by "hop"), or destination address possible (indicated by "dst"). In all cases hop by hop can be used. In cases where no IPIP header is needed, the column is left blank.

Use Case	RPI	RH3	IP-in-IP	IPIP dst
Raf to root	Yes	No	No	--
root to Raf	Yes	No	No	--
root to ~Raf	Yes	No	Yes	hop
~Raf to root	Yes	No	Yes	root
Raf to Int	Yes	No	Yes	root
Int to Raf	Yes	No	Yes	raf
~Raf to Int	Yes	No	Yes	root
Int to ~Raf	Yes	No	Yes	hop
Raf to Raf	Yes	No	No	--
Raf to ~Raf	Yes	No	Yes	hop
~Raf to Raf	Yes	No	Yes	dst
~Raf to ~Raf	Yes	No	Yes	hop

Table 1: Headers needed in Storing mode: RPI, RH3, IP-in-IP encapsulation

5.1. Example of Flow from RPL-aware-leaf to root

As states in Section 16.2 of [RFC6550] a RPL-aware-leaf node does not generally issue DIO messages; a leaf node accepts DIO messages from upstream. (When the inconsistency in routing occurs, a leaf node will generate a DIO with an infinite rank, to fix it). It may issue DAO and DIS messages though it generally ignores DAO and DIS messages.

In storing mode, it is suitable to use RFC 6553 (RPI) to send RPL Information instanceID and rank information.

In this case the flow comprises:

RPL-aware-leaf (6LN) --> 6LR --> 6LR,... --> root (6LBR)

Note: In this document 6LRs, 6LBR are always full-fledge RPL routers, and are the RPL root node.

The 6LN inserts the RPI header, and send the packet to 6LR which decrement the rank in RPI and send the packet up. When the packet arrives to 6LBR, the RPI is removed and the packet is processed.

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

Header	6LN	6LR	6LBR
Inserted headers	RPI	--	--
Removed headers	--	--	RPI
Re-added headers	--	--	--
Modified headers	--	RPI	--
Untouched headers	--	--	--

Storing: Summary of the use of headers from RPL-aware-leaf to root

5.2. Example of Flow from root to RPL-aware-leaf

In this case the flow comprises:

root (6LBR)--> 6LR --> RPL-aware-leaf (6LN)

In this case the 6LBR insert RPI header and send the packet down, the 6LR is going to increment the rank in RPI (examines instanceID for multiple tables), the packet is processed in 6LN and RPI removed.

No IPIP header is required.

Header	6LBR	6LR	6LN
Inserted headers	RPI	--	--
Removed headers	--	--	RPI
Re-added headers	--	--	--
Modified headers	--	RPI	--
Untouched headers	--	--	--

Storing: Summary of the use of headers from root to RPL-aware-leaf

5.3. Example of Flow from root to not-RPL-aware-leaf

In this case the flow comprises:

root (6LBR)--> 6LR --> not-RPL-aware-leaf (6LN)

It includes IPv6-in-IPv6 encapsulation to transmit information not related with the RPL domain. In the 6LBR the RPI header is inserted into an IPv6-in-IPv6 header addressed to the last 6LR, which removes the header before pass the packet to the IPv6 node.

The question in this scenario is how the root knows how to address the IPv6-in-IPv6 header. It can not know that the destination isn't RPL aware, so it must insert an IPv6 that can be removed on the last RPL aware node. Since the root can not know in a storing network where the last RPL aware node is, the IPv6-in-IPv6 header must added hop-by-hop along the path from root to leaf.

An alternative option is to add an attribute in the RPL Target Option to indicate that the target is not RPL aware: future work may explore this possibility.

Header	6LBR	6LR	IPv6
Inserted headers	IPIP(RPI)	--	--
Removed headers	--	IPIP(RPI)	--
Re-added headers	--	--	--
Modified headers	--	--	--
Untouched headers	--	--	--

Storing: Summary of the use of headers from root to not-RPL-aware-leaf

5.4. Example of Flow from not-RPL-aware-leaf to root

In this case the flow comprises:

not-RPL-aware-leaf (6LN) --> 6LR --> root (6LBR)

When the packet arrives from IPv6 node to 6LR, the 6LR will insert an RPI header, encapsuladed in a IPv6-in-IPv6 header. The IPv6-in-IPv6 header can be addressed to the next hop, or to the root. The root removes the header and process the packet.

Header	IPv6	6LR	6LBR
Inserted headers	--	IPIP(RPI)	--
Removed headers	--	--	IPIP(RPI)
Re-added headers	--	--	--
Modified headers	--	--	--
Untouched headers	--	--	--

Storing: Summary of the use of headers from not-RPL-aware-leaf to root

5.5. Example of Flow from RPL-aware-leaf to Internet

RPL information from RFC 6553 should not go out to Internet as it will cause the packet to be discarded at the first non-RPI aware router. The 6LBR must be able to take this information out before sending the packet upwards to the Internet. This requires the RPI header be placed in an IPIP header that the root can remove.

In this case the flow comprises:

RPL-aware-leaf (6LN) --> 6LR --> root (6LBR) --> Internet

The 6LN will insert the RPI in a IPv6-in-IPv6 in a outer header, which may be addressed to the 6LBR (root), or alternatively, it could be addressed hop-by-hop.

Header	6LN	6LR	6LBR	Internet
Inserted headers	IPIP(RPI)	--	--	--
Removed headers	--	--	IPIP(RPI)	--
Re-added headers	--	--	--	--
Modified headers	--	RPI	--	--
Untouched headers	--	--	--	--

Storing: Summary of the use of headers from RPL-aware-leaf to Internet

5.6. Example of Flow from Internet to RPL-aware-leaf

In this case the flow comprises:

Internet --> root (6LBR) --> 6LR --> RPL-aware-leaf (6LN)

When the packet arrives from Internet to 6LBR the RPI header is added in a outer IPv6-in-IPv6 header and send to 6LR, which modifies the rank in the RPI. When the packet arrives 6LN the RPI header is removed and the packet processed.

Header	Internet	6LBR	6LR	6LN
Inserted headers	--	IPIP(RPI)	--	--
Removed headers	--	--	--	IPIP(RPI)
Re-added headers	--	--	--	--
Modified headers	--	--	RPI	--
Untouched headers	--	--	--	--

Storing: Summary of the use of headers from Internet to RPL-aware-leaf

5.7. Example of Flow from not-RPL-aware-leaf to Internet

In this case the flow comprises:

not-RPL-aware-leaf (6LN) --> 6LR --> root (6LBR) --> Internet

The 6LR node will add an IPIP(RPI) header addressed either to the root, or hop-by-hop such that the root can remove the RPI header before passing upwards.

The originating node will ideally leave the IPv6 flow label as zero so that it can be better compressed through the LLN, and the 6LBR will set the flow label to a non-zero value when sending to the Internet.

Header	6LN	6LR	6LBR	Internet
Inserted headers	--	IPIP(RPI)	--	--
Removed headers	--	--	IPIP(RPI)	--
Re-added headers	--	--	--	--
Modified headers	--	--	--	--
Untouched headers	--	--	--	--

Storing: Summary of the use of headers from not-RPL-aware-leaf to Internet

5.8. Example of Flow from Internet to non-RPL-aware-leaf

In this case the flow comprises:

Internet --> root (6LBR) --> 6LR --> not-RPL-aware-leaf (6LN)

The 6LBR will have to add an RPI header within an IPIP header. The IPIP will need to be addressed hop-by-hop along the path as in storing mode, the 6LBR has no idea if the 6LN is RPL aware or not, nor what the closest attached 6LR node is.

The 6LBR MAY set the flow label on the inner IPIP header to zero in order to aid in compression, as the packet will not emerge again from the LLN.

Header	Internet	6LBR	6LR	IPv6
Inserted headers	--	IPIP(RPI)	--	--
Removed headers	--	--	IPIP(RPI)	--
Re-added headers	--	--	--	--
Modified headers	--	--	--	--
Untouched headers	--	--	--	--

Storing: Summary of the use of headers from Internet to non-RPL-aware-leaf

5.9. Example of Flow from RPL-aware-leaf to RPL-aware-leaf

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. Section 9 in [RFC6550].

In this case the flow comprises:

6LN --> 6LR --> common parent (6LR) --> 6LR --> 6LN

This case is assumed in the same RPL Domain. In the common parent, the direction of RPI is changed (from increasing to decreasing the rank).

While the 6LR nodes will update the RPI, no node needs to add or remove the RPI, so no IPIP headers are necessary. The ability to do this depends upon the sending know that the destination is: a) inside the LLN, and b) RPL capable.

The sender can determine if the destination is inside the LLN by looking if the destination address is matched by the DIO's PIO option. This check may be modified by the use of backbone routers, but in this case it is assumed that the backbone routers are RPL capable and so can process the RPI header correctly.

The other check, that the destination is RPL capable is not currently discernible by the sender. This information is necessary to distinguish this test case from Section 5.10.

Header	6LN src	6LR	6LR (common parent)	6LR	6LN dst
Inserted headers	RPI	--	--	--	--
Removed headers	--	--	--	--	RPI
Re-added headers	--	--	--	--	--
Modified headers	--	RPI (decreasing rank)	RPI (increasing rank)	--	--
Untouched headers	--	--	--	--	--

Storing: Summary of the use of headers for RPL-aware-leaf to RPL-aware-leaf

5.10. Example of Flow from RPL-aware-leaf to non-RPL-aware-leaf

In this case the flow comprises:

6LN --> 6LR --> common parent (6LR) --> 6LR --> not-RPL-aware 6LN

The sender, being aware out of band, that the receiver is not RPL aware, sends adds an RPI header inside an IPIP header. The IPIP header needs to be addressed on a hop-by-hop basis so that the last 6LR can remove the RPI header.

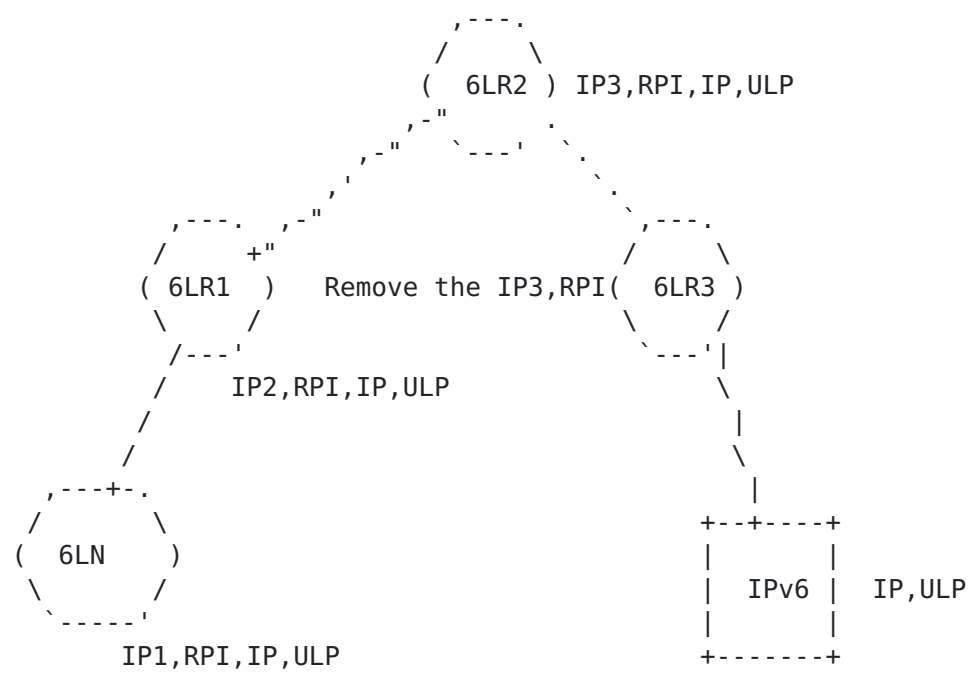

Figure 4: Solution IPv6-in-IPv6 in each hop

Alternatively, if the definition of the Option Type field of RPL Option '01' were changed so that it isn't a "discard if not recognized", then no IPIP header would be necessary. This change is an incompatible on-the-wire change and would require some kind of flag day, possibly a change that is done simultaenously with an updated 6LoRH compress.

Header	6LN	6LR	6LR (common parent)	6LR	IPv6
Inserted headers	IPIP(RPI)	--	--	--	--
Removed headers	--	--	--	IPIP(RPI)	--
Re-added headers	--	--	--	--	--
Modified headers	--	IPIP(RPI)	IPIP(RPI)	--	--
Untouched headers	--	--	--	--	--

Storing: Summary of the use of headers from RPL-aware-leaf to not-RPL-aware-leaf

5.11. Example of Flow from not-RPL-aware-leaf to RPL-aware-leaf

In this case the flow comprises:

not-RPL-aware 6LN --> 6LR --> common parent (6LR) --> 6LR --> 6LN

The 6LR receives the packet from the the IPv6 node and inserts and the RPI header encapsulated in IPv6-in-IPv6 header. The IPIP header could be addresses to the 6LN if the destination is known to the RPL aware, otherwise must send the packet using a hop-by-hop IPIP header. Similar considerations apply from section Section 5.10.

Header	IPv6	6LR	common parent (6LR)	6LR	6LN
Inserted headers	--	IPIP(RPI)	--	--	--
Removed headers	--	--	--	--	IPIP(RPI)
Re-added headers	--	--	--	--	--
Modified headers	--	--	IPIP(RPI)	IPIP(RPI)	--
Untouched headers	--	--	--	--	--

Storing: Summary of the use of headers from not-RPL-aware-leaf to RPL-aware-leaf

5.12. Example of Flow from not-RPL-aware-leaf to not-RPL-aware-leaf

In this case the flow comprises:

not-RPL-aware 6LN (IPv6 node)--> 6LR --> root (6LBR) --> 6LR --> not-RPL-aware 6LN (IPv6 node)

This flow combines the problems of the two previous sections. There is no choice at the first 6LR: it must insert an RPI, and to do that it must add an IPIP header. That IPIP header must be addressed on a hop-by-hop basis.

Header	IPv6 src	6LR	6LR (common parent)	6LR	IPv6 dst
Inserted headers	--	IPIP(RPI)	--	--	--
Removed headers	--	--	--	IPIP(RPI)	--
Re-added headers	--	--	--	--	--
Modified headers	--	--	--	--	--
Untouched headers	--	--	--	--	--

Storing: Summary of the use of headers from not-RPL-aware-leaf to not-RPL-aware-leaf

6. Non Storing mode

Use Case	RPI	RH3	IPIP	IPIP dst
Raf to root	Yes	No	No	--
root to Raf	Yes	Yes	No	--
root to ~Raf	No	Yes	Yes	6LR
~Raf to root	Yes	No	Yes	root
Raf to Int	Yes	No	Yes	root
Int to Raf	opt	Yes	Yes	dst
~Raf to Int	Yes	No	Yes	root
Int to ~Raf	opt	Yes	Yes	6LR
Raf to Raf	Yes	Yes	Yes	root/dst
Raf to ~Raf	Yes	Yes	Yes	root/6LR
~Raf to Raf	Yes	Yes	Yes	root/6LN
~Raf to ~Raf	Yes	Yes	Yes	root/6LR

Table 2: Headers needed in Non-Storing mode: RPI, RH3, IP-in-IP encapsulation

6.1. Example of Flow from RPL-aware-leaf to root

In non-storing mode the leaf node uses default routing to send traffic to the root. The RPI header must be included to avoid/detect loops.

RPL-aware-leaf (6LN) --> 6LR --> root (6LBR)

This situation is the same case as storing mode.

Header	6LN	6LR	6LBR
Inserted headers	RPI	--	--
Removed headers	--	--	RPI
Re-added headers	--	--	RPI
Modified headers	--	--	--
Untouched headers	--	--	--

Non Storing: Summary of the use of headers from RPL-aware-leaf to root

6.2. Example of Flow from root to RPL-aware-leaf

In this case the flow comprises:

root (6LBR)--> 6LR --> RPL-aware-leaf (6LN)

The 6LBR will insert an RH3, and may optionally insert an RPI header. No IPIP header is necessary as the traffic originates with an RPL aware node.

Header	6LBR	6LR	6LN
Inserted headers	(opt: RPI), RH3	--	--
Removed headers	--	--	RH3,RPI
Re-added headers	--	--	--
Modified headers	--	RH3	--
Untouched headers	--	--	--

Non Storing: Summary of the use of headers from root to RPL-aware-leaf

6.3. Example of Flow from root to not-RPL-aware-leaf

In this case the flow comprises:

root (6LBR)--> 6LR --> not-RPL-aware-leaf (IPv6 node)

In 6LBR the RH3 is added, and modified in 6LR where it is fully consumed, but left there. If the RPI is left present, the IPv6 node

which does not understand it will drop it, therefore the RPI should be removed before reaching the IPv6-only node. To permit removal, an IPIP header (hop-by-hop) or addressed to the last 6LR is necessary. Due the complete knowledge of the topology at the root, the 6LBR is able to address the IPIP header to the last 6LR.

Omitting the RPI entirely is therefore a better solution, as no IPIP header is necessary.

Header	6LBR	6LR	IPv6
Inserted headers	RH3	--	--
Removed headers	--	--	--
Re-added headers	--	--	--
Modified headers	--	RH3	--
Untouched headers	--	--	--

Non Storing: Summary of the use of headers from root to not-RPL-aware-leaf

6.4. Example of Flow from not-RPL-aware-leaf to root

In this case the flow comprises:

IPv6-node --> 6LR1 --> 6LR2 --> root (6LBR)

In this case the RPI is added by the first 6LR, encapsulated in an IPIP header, and is not modified in the followings 6LRs. The RPI and entire packet is consumed by the root.

Header	IPv6	6LR1	6LR2	6LBR
Inserted headers	--	IPIP(RPI)	--	--
Removed headers	--	--	--	IPIP(RPI)
Re-added headers	--	--	--	--
Modified headers	--	--	--	--
Untouched headers	--	IPIP(RPI)	--	--

Non Storing: Summary of the use of headers from not-RPL-aware-leaf to root

6.5. Example of Flow from RPL-aware-leaf to Internet

In this case the flow comprises:

RPL-aware-leaf (6LN) --> 6LR --> root (6LBR) --> Internet

This case requires that the RPI be added, but remoted by the 6LBR. The 6LN must therefore add the RPI inside an IPIP header, addressed to the root. This case is identical to storing-mode case.

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

Header	6LN	6LR	6LBR	Internet
Inserted headers	IPIP(RPI)	--	--	--
Removed headers	--	--	IPIP(RPI)	--
Re-added headers	--	--	--	--
Modified headers	--	--	--	--
Untouched headers	--	RPI	--	--

Non Storing: Summary of the use of headers from RPL-aware-leaf to Internet

6.6. Example of Flow from Internet to RPL-aware-leaf

In this case the flow comprises:

Internet --> root (6LBR) --> 6LR --> RPL-aware-leaf (6LN)

The 6LBR must add an RH3 header. As the 6LBR will know the path and address of the target not, it can address the IPIP header to that node. The 6LBR will zero the flow label upon entry in order to aid compression.

The RPI may be added or not.

Header	Internet	6LBR	6LR	6LN
Inserted headers	--	IPIP(RH3,opt:RPI)	--	--
Removed headers	--	--	IPIP(RH3)	--
Re-added headers	--	--	--	--
Modified headers	--	--	IPIP(RH3)	--
Untouched headers	--	--	--	--

Non Storing: Summary of the use of headers from Internet to RPL-aware-leaf

6.7. Example of Flow from not-RPL-aware-leaf to Internet

In this case the flow comprises:

not-RPL-aware-leaf (6LN) --> 6LR --> root (6LBR) --> Internet

In this case the flow label is recommended to be zero in the IPv6 node. As RPL headers are added in the IPv6 node, the first 6LN will add an RPI header inside a new IPIP header. The IPIP header will be addressed to the root. This case is identical to the storing-mode case.

Header	IPv6	6LR	6LBR	Internet
Inserted headers	--	IPIP(RPI)	--	--
Removed headers	--	--	IPIP(RPI)	--
Re-added headers	--	--	--	--
Modified headers	--	--	--	--
Untouched headers	--	--	--	--

Non Storing: Summary of the use of headers from not-RPL-aware-leaf to Internet

6.8. Example of Flow from Internet to non-RPL-aware-leaf

In this case the flow comprises:

Internet --> root (6LBR) --> 6LR --> not-RPL-aware-leaf (6LN)

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

Header	Internet	6LBR	6LR	IPv6
Inserted headers	--	IPIP(RH3,opt:RPI)	--	--
Removed headers	--	--	IPIP(RH3, RPI)	--
Re-added headers	--	--	--	--
Modified headers	--	--	--	--
Untouched headers	--	--	--	--

NonStoring: Summary of the use of headers from Internet to non-RPL-aware-leaf

6.9. Example of Flow from RPL-aware-leaf to RPL-aware-leaf

In this case the flow comprises:

6LN --> 6LR --> root (6LBR) --> 6LR --> 6LN

This case involves only nodes in same RPL Domain. The originating node will add an RPI header to the original packet, and send the packet upwards.

The originating node could put the RPI into an IPIP header addressed to the root, so that the 6LBR can remove that header.

The 6LBR will need to insert an RH3 header, which requires that it add an IPIP header. It may be able to remove the RPI if it was contained in an IPIP header addressed to it. Otherwise, there may be an RPI header buried inside the inner IP header, which should get ignored.

Networks that use the RPL P2P extension [RFC6997] are essentially non-storing DODAGs and fall into this scenario.

Header	6LN src	6LBR	6LR	6LN dst
Inserted headers	IPIP(RPI)	IPIP(RH3 to 6LN,RPI)	--	--
Removed headers	--	--	--	IPIP(RH3,RPI)
Re-added headers	--	--	--	--
Modified headers	--	--	--	--
Untouched headers	--	--	--	--

Non Storing: Summary of the use of headers for RPL-aware-leaf to RPL-aware-leaf

6.10. Example of Flow from RPL-aware-leaf to not-RPL-aware-leaf

In this case the flow comprises:

6LN --> 6LR --> root (6LBR) --> 6LR --> not-RPL-aware 6LN

As in the previous case, the 6LN will insert an RPI header which MUST be in an IPIP header addressed to the root so that the 6LBR can remove this RPI. The 6LBR will then insert an RH3 inside a new IPIP header addressed to the 6LN above the destination node.

Header	6LN	6LBR	6LR	IPv6
Inserted headers	IPIP(RPI)	IPIP(RH3, opt RPI)	--	--
Removed headers	--	IPIP(RPI)	IPIP(RH3, opt RPI)	--
Re-added headers	--	--	--	--
Modified headers	--	--	--	--
Untouched headers	--	--	--	--

Non Storing: Summary of the use of headers from RPL-aware-leaf to not-RPL-aware-leaf

6.11. Example of Flow from not-RPL-aware-leaf to RPL-aware-leaf

In this case the flow comprises:

not-RPL-aware 6LN --> 6LR --> root (6LBR) --> 6LR --> 6LN

This scenario is mostly identical to the previous one. The RPI is added by the first 6LR inside an IPIP header addressed to the root. The 6LBR will remove this RPI, and add it's own IPIP header containing an RH3 header.

Header	IPv6	6LR	6LBR	6LN
Inserted headers	--	IPIP(RPI)	IPIP(RH3)	--
Removed headers	--	IPIP(RPI)	--	IPIP(RH3)
Re-added headers	--	--	--	--
Modified headers	--	--	--	--
Untouched headers	--	--	--	--

Non Storing: Summary of the use of headers from not-RPL-aware-leaf to RPL-aware-leaf

6.12. Example of Flow from not-RPL-aware-leaf to not-RPL-aware-leaf

In this case the flow comprises:

not-RPL-aware 6LN --> 6LR --> root (6LBR) --> 6LR --> not-RPL-aware 6LN

This scenario is the combination of the previous two cases.

Header	IPv6	6LR	6LBR	6LR	IPv6
Inserted headers	--	IPIP(RPI)	IPIP(RH3)	--	--
Removed headers	--	--	IPIP(RPI)	IPIP(RH3, opt RPI)	--
Re-added headers	--	--	--	--	--
Modified headers	--	--	--	--	--
Untouched headers	--	--	--	--	--

Non Storing: Summary of the use of headers from not-RPL-aware-leaf to not-RPL-aware-leaf

7. Observations about the problem

7.1. Storing mode

In the completely general storing case, which includes not-RPL aware leaf nodes, it is not possible for a sending node to know if the destination is RPL aware, and therefore it must always use hop-by-hop IPIP encapsulation, and it can never omit the IPIP encapsulation. See table Table 1

The simplest fully general stiaution for storing mode is to always put in hop-by-hop IPIP headers. [I-D.ietf-roll-routing-dispatch] shows that this hop-by-hop IPIP header can be compressed down to {TBD} bytes.

There are potential significant advantages to having a single code path that always processes IPIP headers with no options.

If all RPL aware nodes can be told/configured that there are no non-RPL aware leaf nodes, then the only case where an IPIP header is needed is when communicating outside the LLN. The 6LBR knows well when the communication is from the outside, and the 6LN can tell by comparing the destination address to the prefix provided in the PIO. If it is known that there are no communications outside the RPL domain (noting that the RPL domain may well extend to outside the LLN), then RPI headers can be included in all packets, and IPIP headers are *never* needed. This may be significantly advantageous in relatively closed systems such as in building or industrial automation. Again, there are advantages to having a single code path.

In order to support the above two cases with full generality, the different situations (always do IPIP vs never use IPIP) should be signaled in the RPL protocol itself.

7.2. Non-Storing mode

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

The 6LBR can recognize non-RPL aware leaf nodes because it will receive a DAO about that node from the 6LN immediately above that node. This means that the non-storing mode case can avoid ever using hop-by-hop IPIP headers.

It is unclear what it would mean for an RH3 header to be present in a hop-by-hop IPIP header. The receiving node ought to consume the IPIP header, and therefore consume the RH3 as well, and then attempt to send the packet again. But intermediate 6LN nodes would not know how to forward the packet, so the RH3 would need to be retained. This is a new kind of IPv6 packet processing. Therefore it may be that on the outbound leg of non-storing RPL networks, that hop-by-hop IPIP header can NOT be used.

[I-D.ietf-roll-routing-dispatch] shows how the destination=root, and destination=6LN IPIP header can be compressed down to {TBD} bytes.

Unlike in the storing mode case, there are no need for all nodes to know about the existence of non-RPL aware nodes. Only the 6LBR needs to change when there are non-RPL aware nodes. Further, in the non-storing case, the 6LBR is informed by the DAOs when there are non-RPL aware nodes.

8. 6LoRH Compression cases

The [I-D.ietf-roll-routing-dispatch] proposes a compression method for RPI, RH3 and IPv6-in-IPv6.

In Storing Mode, for the examples of Flow from RPL-aware-leaf to non-RPL-aware-leaf and non-RPL-aware-leaf to non-RPL-aware-leaf comprise an IP-in-IP and RPI compression headers. The type of this case is critical since IP-in-IP is encapsulating a RPI header.

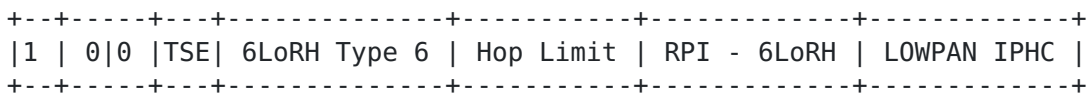

Figure 5: Critical IP-in-IP (RPI).

9. IANA Considerations

There are no IANA considerations related to this document.

10. Security Considerations

The security considerations covering of [RFC6553] and [RFC6554] apply when the packets get into RPL Domain.

11. Acknowledgments

This work is partially funded by the FP7 Marie Curie Initial Training Network (ITN) METRICS project (grant agreement No. 607728).

The authors would like to acknowledge the review, feedback, and comments of Thomas Watteyne, Xavier Vilajosana, Robert Cragie and Simon Duquennoy.

12. References

12.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[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, <http://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, <http://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, <http://www.rfc-editor.org/info/rfc6554>.

12.2. Informative References

[I-D.ietf-6tisch-architecture] Thubert, P., "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4", draft-ietf-6tisch-architecture-09 (work in progress), November 2015.

[I-D.ietf-roll-routing-dispatch] Thubert, P., Bormann, C., Toutain, L., and R. Cragie, "6LoWPAN Routing Header", draft-ietf-roll-routing-dispatch-00 (work in progress), March 2016.

[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, <http://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, <http://www.rfc-editor.org/info/rfc7102>.

[Second6TischPlugtest] "2nd 6Tisch Plugtest", <http://www.ietf.org/mail-archive/web/6tisch/current/pdfgDMQcdCkRz.pdf>.

Authors' Addresses

Maria Ines Robles
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: maria.ines.robles@ericsson.com

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

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

Pascal Thubert
Cisco Systems, Inc
Village d'Entreprises Green Side 400, Avenue de Roumanille
Batiment T3, Biot - Sophia Antipolis 06410
France

Email: pthubert@cisco.com