

ROLL Working Group
Internet-Draft
Updates: [6553](#), [6550](#), [8138](#) (if approved)
Intended status: Standards Track
Expires: April 30, 2018

M. Robles
Ericsson
M. Richardson
SSW
P. Thubert
Cisco
October 27, 2017

When to use [RFC 6553](#), 6554 and IPv6-in-IPv6
draft-ietf-roll-useofrplinfo-17

Abstract

This document looks at different data flows through LLN (Low-Power and Lossy Networks) where RPL (IPv6 Routing Protocol for Low-Power and Lossy Networks) 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. Additionally, this document updates the [RFC 6553](#) adding a change to the RPL Option Type and the [RFC 6550](#) to indicate about this change.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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

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

This Internet-Draft will expire on April 30, 2018.

Copyright Notice

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

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of

Internet-Draft

Useof6553

October 2017

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
2.	Terminology and Requirements Language	4
2.1.	hop-by-hop IPv6-in-IPv6 headers	5
3.	Updates to RFC6553 , RFC6550 and RFC 8138	5
3.1.	Updates to RFC 6553	5
3.2.	Updates to RFC 8138	6
3.3.	Updates to RFC 6550 : Indicating the new RPI in the DODAG Configuration Option Flag.	6
4.	Sample/reference topology	8
5.	Use cases	10
6.	Storing mode	12
6.1.	Storing Mode: Interaction between Leaf and Root	13
6.1.1.	SM: Example of Flow from RPL-aware-leaf to root	14
6.1.2.	SM: Example of Flow from root to RPL-aware-leaf	15
6.1.3.	SM: Example of Flow from root to not-RPL-aware-leaf	15
6.1.4.	SM: Example of Flow from not-RPL-aware-leaf to root	16
6.2.	Storing Mode: Interaction between Leaf and Internet	17
6.2.1.	SM: Example of Flow from RPL-aware-leaf to Internet	17
6.2.2.	SM: Example of Flow from Internet to RPL-aware-leaf	18
6.2.3.	SM: Example of Flow from not-RPL-aware-leaf to Internet	19
6.2.4.	SM: Example of Flow from Internet to non-RPL-aware-leaf	20
6.3.	Storing Mode: Interaction between Leaf and Leaf	21
6.3.1.	SM: Example of Flow from RPL-aware-leaf to RPL-aware-leaf	21
6.3.2.	SM: Example of Flow from RPL-aware-leaf to non-RPL-aware-leaf	22
6.3.3.	SM: Example of Flow from not-RPL-aware-leaf to RPL-aware-leaf	23
6.3.4.	SM: Example of Flow from not-RPL-aware-leaf to not-RPL-aware-leaf	24
7.	Non Storing mode	26
7.1.	Non-Storing Mode: Interaction between Leaf and Root	27

7.1.1.	Non-SM: Example of Flow from RPL-aware-leaf to root .	28
7.1.2.	on-SM: Example of Flow from root to RPL-aware-leaf .	28
7.1.3.	Non-SM: Example of Flow from root to not-RPL-aware-leaf	29
7.1.4.	Non-SM: Example of Flow from not-RPL-aware-leaf to	

	root	30
7.2.	Non-Storing Mode: Interaction between Leaf and Internet .	31
7.2.1.	Non-SM: Example of Flow from RPL-aware-leaf to Internet	31
7.2.2.	Non-SM: Example of Flow from Internet to RPL-aware-leaf	32
7.2.3.	Non-SM: Example of Flow from not-RPL-aware-leaf to Internet	33
7.2.4.	Non-SM: Example of Flow from Internet to not-RPL-aware-leaf	34
7.3.	Non-Storing Mode: Interaction between Leafs	35
7.3.1.	Non-SM: Example of Flow from RPL-aware-leaf to RPL-aware-leaf	35
7.3.2.	Non-SM: Example of Flow from RPL-aware-leaf to not-RPL-aware-leaf	37
7.3.3.	Non-SM: Example of Flow from not-RPL-aware-leaf to RPL-aware-leaf	38
7.3.4.	Non-SM: Example of Flow from not-RPL-aware-leaf to not-RPL-aware-leaf	39
8.	Observations about the cases	40
8.1.	Storing mode	40
8.2.	Non-Storing mode	41
9.	6LoRH Compression cases	41
10.	IANA Considerations	41
11.	Security Considerations	42
12.	Acknowledgments	45
13.	References	45
13.1.	Normative References	45
13.2.	Informative References	46
	Authors' Addresses	48

[1.](#) Introduction

RPL (IPv6 Routing Protocol for Low-Power and Lossy Networks) [[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 clarifies what is the correct and the 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](#)], an efficient IP-in-IP technique, and use 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](#)] applies to this document: LBR, LLN, RPL, RPL Domain and ROLL.

RPL-node: It is device which implements RPL, thus we can say that the device is RPL-capable or RPL-aware. Please note that the device can be found inside the LLN or outside LLN. In this document a RPL-node which is a leaf of a DODAG is called RPL-aware-leaf.

RPL-not-capable: It is device which does not implement RPL, thus we can say that the device is not-RPL-aware. Please note that the device can be found inside the LLN. In this document a not-RPL-aware node which is a leaf of a DODAG is called not-RPL-aware-leaf.

pledge: a new device which seeks admission to a network. (from [\[I-D.ietf-anima-bootstrapping-keyinfra\]](#))

Join Registrar and Coordinator (JRC): a device which brings new nodes (pledges) into a network. (from [\[I-D.ietf-anima-bootstrapping-keyinfra\]](#))

Flag day: A "flag day" is a procedure in which the network, or a part of it, is changed during a planned outage, or suddenly, causing an outage while the network recovers [\[RFC4192\]](#)

[2.1.](#) hop-by-hop IPv6-in-IPv6 headers

The term "hop-by-hop IPv6-in-IPv6" header refers to: adding a header that originates from a node to an adjacent node, using the addresses (usually the GUA or ULA, but could use the link-local addresses) of each node. If the packet must traverse multiple hops, then it must be decapsulated at each hop, and then re-encapsulated again in a similar fashion.

[3.](#) Updates to [RFC6553](#), [RFC6550](#) and [RFC 8138](#)

[3.1.](#) Updates to [RFC 6553](#)

[RFC6553] states as showed below, that in the Option Type field of the RPL Option header, 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 en route. The remaining bits serve as the option type.

Hex Value	Binary Value
-----------	--------------

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

Figure 1: Option Type in RPL Option.

Recent changes in [I-D.ietf-6man-rfc2460bis], state: "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". Processing of the Hop-by-Hop Options header (by IPv6 intermediate nodes) is now optional, but if they are configured to process the header, and if such nodes encounter an option with the first two bits set to 01, they will drop the packet (if they conform [I-D.ietf-6man-rfc2460bis]). The hosts should do the same, irrespective of the configuration.

Based on That, if an IPv6 (intermediate) node (RPL-not-capable) receives a packet with an RPL Option, it should ignore the HBH RPL option (skip over this option and continue processing the header).

Thus, this document updates the Option Type field to: 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 (Section 4.2.[I-D.ietf-6man-rfc2460bis]) 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 remaining bits serve as the option type and remain as 0x3. 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 [RFC6553] RPL Hop-by-Hop option known as RPI. This is an update to [RFC6553].

Hex Value	Binary Value			Description	Reference
	act	chg	rest		
0x23	00	1	00011	RPL Option	[RFCXXXX]

Figure 2: Proposed change to the Option Type in RPL Option.

This change creates a flag day for existing networks which are currently using 0x63 as the RPI value. A move to 0x23 will not be understood by those networks. It is suggested that implementations accept both 0x63 and 0x23 when processing. When forwarding packets, implementations SHOULD use the same value as it was received. When originating new packets, implementations SHOULD have an option to determine which value to originate with, this option is controlled by the DIO option described below.

A network which is switching from straight 6lowpan compression mechanism to those described in [[I-D.ietf-roll-routing-dispatch](#)] will experience a flag day in the data compression anyway, and if possible this change can be deployed at the same time.

[3.2.](#) Updates to [RFC 8138](#)

RPI-6LoRH header provides a compressed form for the RPL RPI [[RFC8138](#)]. It should be considered when the Option Type in RPL Option is decompressed, should take the value of 0x23 instead of 0x63.

[3.3.](#) Updates to [RFC 6550](#): Indicating the new RPI in the DODAG Configuration Option Flag.

In order to avoid a flag day caused by lack of interoperation between new RPI (0x23) and old RPI (0x63) nodes, the new nodes need to be told that there are old RPI nodes present. This can be done via the DODAG Configuration Option flag which will propagate through the network. Failure to receive this information will cause dual mode nodes to originate traffic with the old-RPI (0x63) value.

As states in [[RFC6550](#)] the DODAG Configuration option is present in DIO messages. the DODAG Configuration option distribute configuration information which is generally static and unchanging 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 must not modify this information when propagating the DODAG Configuration option.

The DODAG Configuration Option is as follow. We are interested in the Flag field. The remaining fields states as defined in [[RFC6550](#)].

Flags: The 4-bits remaining unused in the Flags field are reserved for flags. The field MUST be initialized to zero by the sender and MUST be ignored by the receiver.

0	1	2
+-----+	+-----+	+-----+
Type = 0x04	Opt Length = 14	Flags A PCS DIOIntDoub
+-----+	+-----+	+-----+
DIOIntMin.	DIORedund.	MaxRankIncrease
+-----+	+-----+	+-----+
MinHopRankIncrease		OCP
+-----+	+-----+	+-----+
Reserved	Def. Lifetime	Lifetime Unit
+-----+	+-----+	+-----+

Figure 3: DODAG Configuration Option.

We propose to use the flag in the DODAG Configuration option as follows:

+-----+	+-----+	+-----+
Bit number	Description	Reference
+-----+	+-----+	+-----+
3	RPI 0x23 enable	This document
+-----+	+-----+	+-----+

Figure 4: DODAG Configuration Option Flag to indicate the RPI-flag-day.

In case of rebooting, the DIO is sent with flag indicating the new RPI value.

A RPL network in general 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 [[RFC4443](#)] message with Type 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 showed in Figure 1.

RPL supports two modes of Downward traffic: in storing mode (RPL-SM), it is fully stateful or an in non-storing (RPL-NSM), 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 at the time of writing this document.

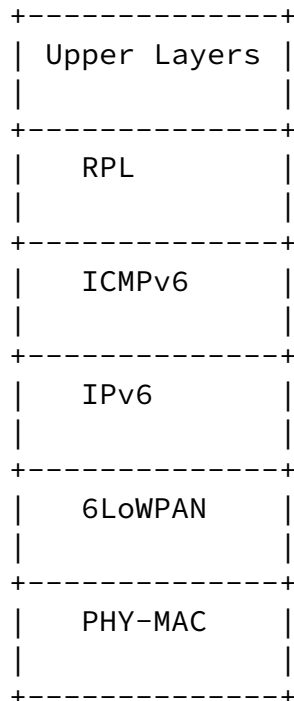


Figure 5: RPL Stack.

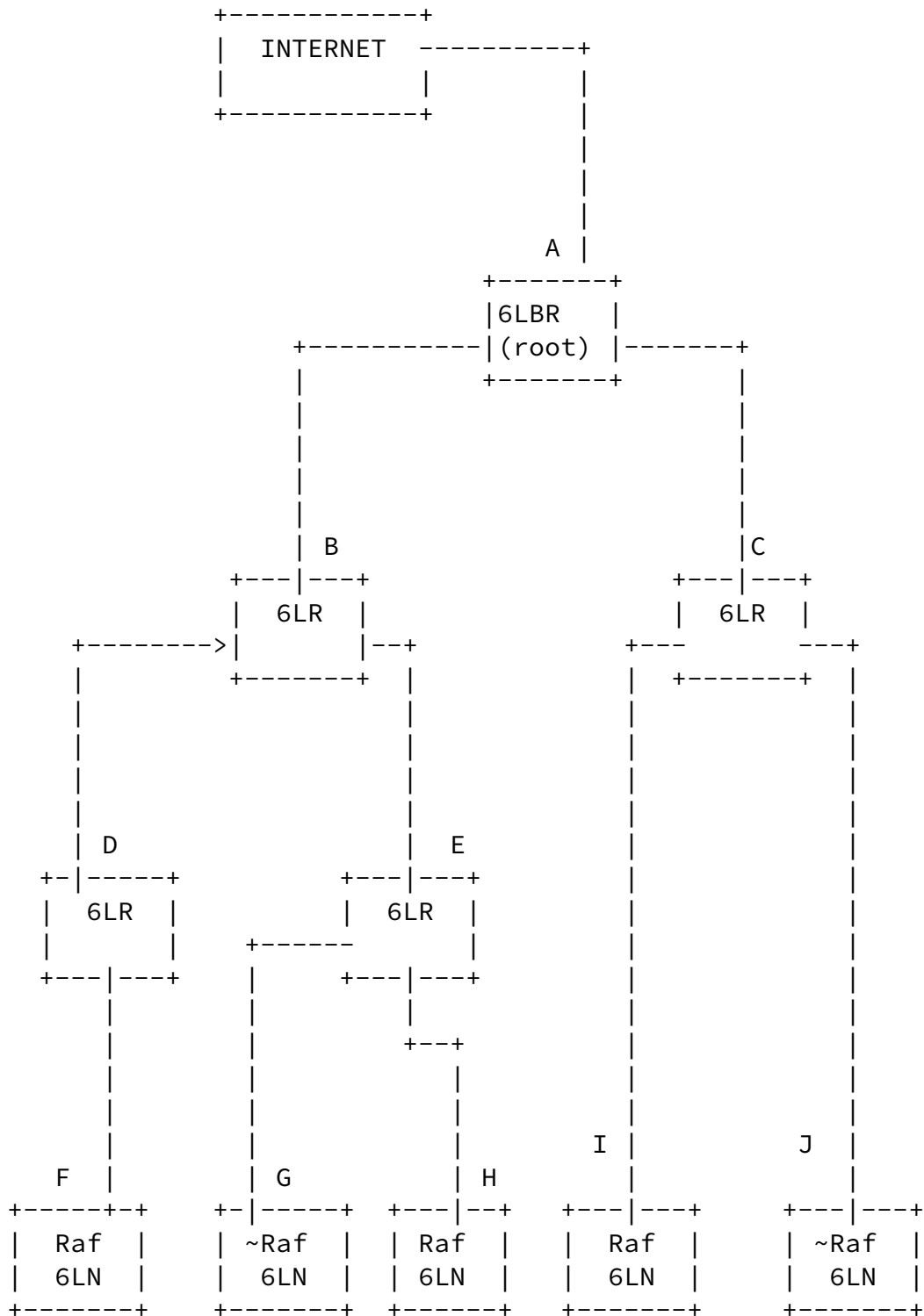


Figure 6: A reference RPL Topology.

Figure 2 shows the reference RPL Topology for this document. The

letters above the nodes are there so that they may be referenced in subsequent sections. In the figure, a 6LR is a router. A 6LN can be

a router or a host. The 6LN leaves (Raf - "RPL aware leaf"-) marked as (F and I) are RPL hosts that does not have forwarding capability. The 6LN leaf (H) is a RPL router. The leaves marked as ~Raf "not-RPL aware leaf" (G and J) are devices which do not speak RPL at all (not-RPL-aware), but uses Router-Advertisements, 6LowPAN DAR/DAC and efficient-ND only to participate in the network [[RFC6775](#)]. In the document these leaves (G and J) are often named IPv6 node. The 6LBR in the figure is the root of the Global DODAG.

5. Use cases

In the data plane a combination of [RFC6553](#), [RFC6554](#) and IPv6-in-IPv6 encapsulation is going to be analyzed for a number of representative traffic flows.

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

The uses cases describe the communication between RPL-aware-nodes, with the root (6LBR), and with Internet. This document also describe the communication between nodes acting as leaf that does not understand RPL and they are part of hte LLN. We name these nodes as not-RPL-aware-leaf.(e.g. [section 5.4](#)- Flow from not-RPL-aware-leaf to root) We describe also how is the communication inside of the LLN when it has the final destination addressed outside of the LLN e.g. with destination to Internet. (e.g. [section 5.7](#)- Flow from not-RPL-aware-leaf to Internet)

The uses cases comprise as follow:

Interaction between Leaf and Root:

RPL-aware-leaf to root

root to RPL-aware-leaf

not-RPL-aware-leaf to root

root to not-RPL-aware-leaf

Interaction between Leaf and Internet:

RPL-aware-leaf to Internet

Internet to RPL-aware-leaf

not-RPL-aware-leaf to Internet

Robles, et al.

Expires April 30, 2018

[Page 10]

Internet-Draft

Useof6553

October 2017

Internet to not-RPL-aware-leaf

Interaction between Leafs:

RPL-aware-leaf to RPL-aware-leaf (storing and non-storing)

RPL-aware-leaf to not-RPL-aware-leaf (non-storing)

not-RPL-aware-leaf to RPL-aware-leaf (storing and non-storing)

not-RPL-aware-leaf to not-RPL-aware-leaf (non-storing)

This document assumes 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 [[RFC2460](#)]. Extensions may not be added or removed except by the sender or the receiver.

But, options in the Hop-by-Hop Option Header whose option type has the first two bits set to '00' MUST be ignored when received by a host or router that does not understand that option ([Section 4.2](#) [[I-D.ietf-6man-rfc2460bis](#)]).

This means that when the no-drop RPI option code 0x23 is used, a packet that leaves the RPL domain of an LLN (or that leaves the LLN entirely) will not be discarded when it contains the [[RFC6553](#)] RPL Hop-by-Hop option known as RPI. Thus, the RPI Hop-by-Hop option MAY be left in place even if the end host does not understand it.

NOTE: There is some possible security risk when the RPI information is released to the Internet. At this point this is a theoretical situation. It is clear that the RPI option would waste some network

bandwidth when it escapes.

An intermediate router that needs to add an extension header (SHR3 or RPI Option) must encapsulate the packet in an (additional) outer IP header. The new header can be placed is placed after this new outer IP header.

A corollary is that an SHR3 or RPI 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. When it does so, the whole encapsulating header must be removed. (A replacement may be added). This sometimes can result in outer IP headers being addressed to the next hop router using link-local addresses.

Both RPI and RH3 headers may be modified in very specific ways 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 can not secure the values which mutate.

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 root. 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 (as the root has already made all routing decisions). There still may be cases (such as in 6tisch) where the instanceID portion of the RPI header may still be needed to pick an appropriate priority or channel at each hop.

In the tables present in this document, 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 extensive to make sure that the process is clear, while later examples are more concise.

6. Storing mode

In storing mode (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 PIO option.

The following table summarizes what headers are needed in the following scenarios, and indicates when the IP-in-IP header must be inserted on a hop-by-hop basis, and when it can target the destination node directly. There are these 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 IP-in-IP header is needed, the column is left blank.

In all cases the RPI headers are needed, since it identifies inconsistencies (loops) in the routing topology. In all cases the RH3 is not need because we do not indicate the route in storing mode.

The leaf can be a router 6LR or a host, both indicated as 6LN (Figure 2).

Interaction between	Use Case	IP-in-IP	IP-in-IP dst
Leaf - Root	Raf to root	No	--
	root to Raf	No	--
	root to ~Raf	No	--
	~Raf to root	Yes	root
Leaf - Internet	Raf to Int	No	--
	Int to Raf	Yes	Raf
	~Raf to Int	Yes	root
	Int to ~Raf	Yes	hop

	Raf to Raf	No	--
	Raf to ~Raf	No	--
Leaf - Leaf	~Raf to Raf	Yes	dst
	~Raf to ~Raf	Yes	hop

Figure 7: IP-in-IP encapsulation in Storing mode.

6.1. Storing Mode: Interaction between Leaf and Root

In this section we are going to describe the communication flow in storing mode (SM) between,

RPL-aware-leaf to root

root to RPL-aware-leaf

not-RPL-aware-leaf to root

root to not-RPL-aware-leaf

6.1.1. SM: Example of Flow from RPL-aware-leaf to root

In storing mode, [RFC 6553](#) (RPI) is used to send RPL Information instanceID and rank information.

As stated 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 this case the flow comprises:

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

For example, the communication flow would be: Node F --> Node E --> Node B --> Node A root(6LBR)

6LR_i (Node E and Node B) are the intermediate routers from source to destination. In this case, " $1 \leq i \leq n$ ", n is the number of routers (6LR) that the packet go through from source (6LN) to destination (6LBR).

As it was mentioned In this document 6LRs, 6LBR are always full-fledge RPL routers.

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

No IP-in-IP header is required.

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 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 | 6LRi | 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

6.1.2. SM: Example of Flow from root to RPL-aware-leaf

In this case the flow comprises:

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

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

6LR_i are the intermediate routers from source to destination. In this case, " $1 \leq i \leq n$ ", n is the number of routers (6LR) that the packet go through from source (6LBR) to destination (6LN).

In this case the 6LBR inserts RPI header and sends 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 IP-in-IP header is required.

Header	6LBR	6LR_i	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

6.1.3. SM: Example of Flow from root to not-RPL-aware-leaf

In this case the flow comprises:

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

For example, the communication flow would be: Node A root(6LBR) --> Node B --> Node E --> Node G

6LR_i are the intermediate routers from source to destination. In this case, "1 <= i >= n", n is the number of routers (6LR) that the packet go through from source (6LBR) to destination (IPv6).

As the RPI extension can be ignored by the not-RPL-aware leaf, this situation is identical to the previous scenario.

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

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

6.1.4. SM: Example of Flow from not-RPL-aware-leaf to root

In this case the flow comprises:

not-RPL-aware-leaf (IPv6) --> 6LR₁ --> 6LR_i --> root (6LBR)

For example, the communication flow would be: Node G --> Node E --> Node B --> Node A root(6LBR)

6LR_i are the intermediate routers from source to destination. In this case, "1 < i >= n", n is the number of routers (6LR) that the packet go through from source (IPv6) to destination (6LBR). For example, 6LR₁ (i=1) is the router that receives the packets from the IPv6 node (Node G).

When the packet arrives from IPv6 node (Node G) to 6LR₁ (Node E), the 6LR₁ will insert a RPI header, encapsuladed in a IPv6-in-IPv6 header. The IPv6-in-IPv6 header can be addressed to the next hop (Node B), or to the root (Node A). The root removes the header and processes the packet.

Internet-Draft

Useof6553

October 2017

Header	IPv6	6LR_1	6LR_i	6LBR
Inserted headers	--	IP-in-IP(RPI)	--	--
Removed headers	--	--	--	IP-in-IP(RPI)
Re-added headers	--	--	--	--
Modified headers	--	--	IP-in-IP(RPI)	--
Untouched headers	--	--	--	--

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

[6.2.](#) Storing Mode: Interaction between Leaf and Internet

In this section we are going to describe the communication flow in storing mode (SM) between,

RPL-aware-leaf to Internet

Internet to RPL-aware-leaf

not-RPL-aware-leaf to Internet

Internet to not-RPL-aware-leaf

[6.2.1.](#) SM: Example of Flow from RPL-aware-leaf to Internet

RPL information from [RFC 6553](#) MAY go out to Internet as it will be ignored by nodes which have not been configured to be RPI aware.

In this case the flow comprises:

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

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

6LR_i are the intermediate routers from source to destination. In this case, "1 ≤ i ≤ n", n is the number of routers (6LR) that the packet go through from source (6LN) to 6LBR.

No IP-in-IP header is required.

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

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

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

6.2.2. SM: Example of Flow from Internet to RPL-aware-leaf

In this case the flow comprises:

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

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

6LR_i are the intermediate routers from source to destination. In this case, "1 ≤ i ≤ n", n is the number of routers (6LR) that the packet go through from 6LBR to destination(6LN).

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

Internet-Draft

Useof6553

October 2017

Header	Internet	6LBR	6LR_i	6LN
Inserted headers	--	IP-in-IP(RPI)	--	--
Removed headers	--	--	--	IP-in-IP(RPI)
Re-added headers	--	--	--	--
Modified headers	--	--	IP-in-IP(RPI)	--
Untouched headers	--	--	--	--

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

[6.2.3.](#) SM: Example of Flow from not-RPL-aware-leaf to Internet

In this case the flow comprises:

not-RPL-aware-leaf (IPv6) --> 6LR_1 --> 6LR_i -->root (6LBR) --> Internet

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

6LR_i are the intermediate routers from source to destination. In this case, " $1 < i \leq n$ ", n is the number of routers (6LR) that the packet go through from source(IPv6) to 6LBR.

The 6LR₁ (i=1) node will add an IP-in-IP(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 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.

Header	IPv6	6LR ₁	6LR _i [i=2,..,n]	6LBR	Internet
Inserted headers	--	IP-in-IP(RPI)	--	--	--
Removed headers	--	--	--	IP-in-IP(RPI)	--
Re-added headers	--	--	--	--	--
Modified headers	--	--	IP-in-IP(RPI)	--	--
Untouched headers	--	--	--	--	--

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

6.2.4. SM: Example of Flow from Internet to non-RPL-aware-leaf

In this case the flow comprises:

Internet --> root (6LBR) --> 6LR_i --> not-RPL-aware-leaf (IPv6)

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

6LR_i are the intermediate routers from source to destination. In this case, " $1 < i \leq n$ ", n is the number of routers (6LR) that the packet go through from 6LBR to not-RPL-aware-leaf (IPv6). 6LR_i updates the rank in the RPI.

The 6LBR will have to add an RPI header within an IP-in-IP header. The IP-in-IP can be addressed to the not-RPL-aware-leaf, leaving the RPI inside.

The 6LBR MAY set the flow label on the inner IP-in-IP header to zero in order to aid in compression.

Header	Internet	6LBR	6LR_i	IPv6
Inserted headers	--	IP-in-IP(RPI)	--	--
Removed headers	--	--	--	--
Re-added headers	--	--	--	--
Modified headers	--	--	IP-in-IP(RPI)	--
Untouched	--	--	--	RPI

headers			(Ignored)
---------	--	--	-----------

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

6.3. Storing Mode: Interaction between Leaf and Leaf

In this section we are going to describe the communication flow in storing mode (SM) between,

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

6.3.1. SM: 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_{ia} --> common parent (6LR_x) --> 6LR_{id} --> 6LN

For example, the communication flow could be: Node F --> Node D --> Node B --> Node E --> Node H

6LR_{ia} (Node D) are the intermediate routers from source to the common parent (6LR_x) (Node B) In this case, " $1 \leq ia \leq n$ ", n is the number of routers (6LR) that the packet go through from 6LN (Node F) to the common parent (6LR_x).

6LR_{id} (Node E) are the intermediate routers from the common parent (6LR_x) (Node B) to destination 6LN (Node H). In this case, " $1 \leq id \leq m$ ", m is the number of routers (6LR) that the packet go through from the common parent (6LR_x) to destination 6LN.

This case is assumed in the same RPL Domain. In the common parent (Node B), 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 IP-in-IP headers are necessary. This may be done regardless of where the destination is, as the included RPI will be ignored by the receiver.

Header	6LN src	6LR_ia	6LR_x (common parent)	6LR_id	6LN dst
Inserted headers	RPI	--	--	--	--
Removed headers	--	--	--	--	RPI
Re-added headers	--	--	--	--	--
Modified headers	--	RPI	RPI	RPI	--
Untouched headers	--	--	--	--	--

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

6.3.2. SM: Example of Flow from RPL-aware-leaf to non-RPL-aware-leaf

In this case the flow comprises:

6LN --> 6LR_ia --> common parent (6LR_x) --> 6LR_id --> not-RPL-aware 6LN (IPv6)

For example, the communication flow could be: Node F --> Node D --> Node B --> Node E --> Node G

6LR_ia are the intermediate routers from source (6LN) to the common parent (6LR_x) In this case, " $1 \leq ia \leq n$ ", n is the number of routers (6LR) that the packet go through from 6LN to the common parent (6LR_x).

6LR_id (Node E) are the intermediate routers from the common parent (6LR_x) (Node B) to destination not-RPL-aware 6LN (IPv6) (Node G). In this case, " $1 \leq id \leq m$ ", m is the number of routers (6LR) that the packet go through from the common parent (6LR_x) to destination 6LN.

This situation is identical to the previous situation [Section 6.3.1](#)

Header	6LN src	6LR_ia	6LR_x(common parent)	6LR_id	IPv6
Inserted headers	RPI	--	--	--	--
Removed headers	--	--	--	--	RPI
Re-added headers	--	--	--	--	--
Modified headers	--	RPI	RPI	RPI	--
Untouched headers	--	--	--	--	RPI(Ignored)

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

6.3.3. SM: Example of Flow from not-RPL-aware-leaf to RPL-aware-leaf

In this case the flow comprises:

not-RPL-aware 6LN (IPv6) --> 6LR_ia --> common parent (6LR_x) --> 6LR_id --> 6LN

For example, the communication flow could be: Node G --> Node E --> Node B --> Node D --> Node F

6LR_ia (Node E) are the intermediate routers from source (not-RPL-aware 6LN (IPv6)) (Node G) to the common parent (6LR_x) (Node B) In this case, " $1 \leq ia \leq n$ ", n is the number of routers (6LR) that the packet go through from source to the common parent.

6LR_id (Node D) are the intermediate routers from the common parent (6LR_x) (Node B) to destination 6LN (Node F). In this case, " $1 \leq id \leq m$ ", m is the number of routers (6LR) that the packet go through from the common parent (6LR_x) to destination 6LN.

Internet-Draft

Useof6553

October 2017

The 6LR_ia (ia=1) (Node E) receives the packet from the the IPv6 node (Node G) and inserts and the RPI header encapsulated in IPv6-in-IPv6 header. The IP-in-IP header is addressed to the destination 6LN (Node F).

Header	IPv6	6LR_ia	common parent (6LRx)	6LR_id	6LN
Inserted headers	--	IP-in-IP(RPI)	--	--	--
Removed headers	--	--	--	--	IP-in-IP(RPI)
Re-added headers	--	--	--	--	--
Modified headers	--	--	IP-in-IP(RPI)	IP-in-IP(RPI)	--
Untouched headers	--	--	--	--	--

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

[6.3.4.](#) SM: Example of Flow from not-RPL-aware-leaf to not-RPL-aware-leaf

In this case the flow comprises:

not-RPL-aware 6LN (IPv6 src)--> 6LR_1--> 6LR_ia --> root (6LBR) --> 6LR_id --> not-RPL-aware 6LN (IPv6 dst)

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

Internet 6LR_{ia} (Node E and Node B) are the intermediate routers from source (not-RPL-aware 6LN (IPv6 src))(Node G) to the root (6LBR) (Node A) In this case, " $1 < ia \leq n$ ", n is the number of routers (6LR) that the packet go through from IPv6 src to the root.

6LR_{id} (C) are the intermediate routers from the root (Node A) to destination (IPv6 dst) (Node J). In this case, " $1 \leq id \leq m$ ", m is the number of routers (6LR) that the packet go through from the root to destination (IPv6 dst).

This flow is identical to [Section 6.3.3](#)

The 6LR₁ (Node E) receives the packet from the the IPv6 node (Node G) and inserts the RPI header (RPI_a) encapsulated in IPv6-in-IPv6 header. The IPv6-in-IPv6 header is addressed to the 6LBR. The 6LBR remove the IPv6-in-IPv6 header and insert another one (RPI_b) with destination to 6LR_m (Node C) node.

One of the side-effects of inserting IP-in-IP RPI header at 6LR₁, is that now all the packets will go through the 6LBR, even though there exists a shorter P2P path to the destination 6LN in storing mode. One possible solution is given by the work in [\[I-D.ietf-roll-dao-projection\]](#). Once we have route projection, the root can find that this traffic deserves optimization (based on volume and path length, or additional knowledge on that particular flow) and project a DAO into 6LR₁.

Internet-Draft

Useof6553

October 2017

Header	IPv6 src	6LR_1	6LR_ia	6LBR	6LR_m	IPv6 dst
Inserted headers	--	IP-in-IP(RPI_a)	--	IP-in-IP(RPI_b)	--	--
Removed headers	--	--	--	--	--	--
Re-added headers	--	--	--	--	IP-in-IP(RPI_b)	--
Modified headers	--	--	IP-in-IP(RPI_a)	--	IP-in-IP(RPI_b)	--
Untouched headers	--	--	--	--	--	--

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

7. 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 non-RPL aware nodes. Only the 6LBR needs to change when there are non-RPL aware nodes.

The following table summarizes what headers are needed in the following scenarios, and indicates when the RPI, RH3 and IP-in-IP header must be inserted. There are these possible situations: destination address possible (indicated by "dst"), to a 6LR, to a 6LN or to the root. In cases where no IP-in-IP header is needed, the column is left blank.

The leaf can be a router 6LR or a host, both indicated as 6LN (Figure 3).

Interaction between	Use Case	RPI	RH3	IP-in-IP	IP-in-IP dst
Leaf - Root	Raf to root	Yes	No	No	--
	root to Raf	Opt	Yes	No	--
	root to ~Raf	No	Yes	Yes	6LR
	~Raf to root	Yes	No	Yes	root
Leaf - Internet	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
Leaf - Leaf	~Raf to Raf	Yes	Yes	Yes	root/6LN
	~Raf to ~Raf	Yes	Yes	Yes	root/6LR

Figure 8: Headers needed in Non-Storing mode: RPI, RH3, IP-in-IP encapsulation.

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

In this section we are going to describe the communication flow in Non Storing Mode (Non-SM) between,

RPL-aware-leaf to root

root to RPL-aware-leaf

not-RPL-aware-leaf to root

root to not-RPL-aware-leaf

7.1.1. Non-SM: 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_i --> root(6LBR)

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

6LR_i are the intermediate routers from source to destination. In this case, "1 <= i >= n", n is the number of routers (6LR) that the packet go through from source (6LN) to destination (6LBR).

This situation is the same case as storing mode.

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

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

[7.1.2.](#) on-SM: Example of Flow from root to RPL-aware-leaf

In this case the flow comprises:

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

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

6LR_i are the intermediate routers from source to destination. In this case, " $1 \leq i \leq n$ ", n is the number of routers (6LR) that the packet go through from source (6LBR) to destination (6LN).

The 6LBR will insert an RH3, and may optionally insert an RPI header. No IP-in-IP header is necessary as the traffic originates with an RPL aware node, the 6LBR. The destination is known to RPL-aware because, the root knows the whole topology in non-storing mode.

Header	6LBR	6LR_i	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

[7.1.3.](#) Non-SM: Example of Flow from root to not-RPL-aware-leaf

In this case the flow comprises:

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

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

6LR_i are the intermediate routers from source to destination. In this case, " $1 \leq i \leq n$ ", n is the number of routers (6LR) that the packet go through from source (6LBR) to destination (IPv6).

In 6LBR the RH3 is added, modified in each intermediate 6LR (6LR_1 and so on) and it is fully consumed in the last 6LR (6LR_n), but left there. If RPI is left present, the IPv6 node which does not understand it will ignore it (following 2460bis), thus encapsulation is not necessary. Due the complete knowledge of the topology at the root, the 6LBR is able to address the IP-in-IP header to the last 6LR.

Header	6LBR	6LR _i (i=1)	6LR _n (i=n)	IPv6
Inserted headers	(opt: RPI), RH3	--	--	--
Removed headers	--	RH3	--	--
Re-added headers	--	--	--	--
Modified headers	--	(opt: RPI), RH3	(opt: RPI), RH3	--
Untouched headers	--	--	--	RPI

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

7.1.4. Non-SM: Example of Flow from not-RPL-aware-leaf to root

In this case the flow comprises:

not-RPL-aware-leaf (IPv6) --> 6LR₁ --> 6LR_i --> root (6LBR)

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

6LR_i are the intermediate routers from source to destination. In this case, " $1 < i \leq n$ ", n is the number of routers (6LR) that the packet go through from source (IPv6) to destination (6LBR). For example, 6LR₁ (i=1) is the router that receives the packets from the IPv6 node.

In this case the RPI is added by the first 6LR (6LR₁) (Node E), encapsulated in an IP-in-IP header, and is modified in the followings 6LRs. The RPI and entire packet is consumed by the root.

Internet-Draft

Useof6553

October 2017

Header	IPv6	6LR_1	6LR_i	6LBR
Inserted headers	--	IP-in-IP(RPI)	--	--
Removed headers	--	--	--	IP-in-IP(RPI)
Re-added headers	--	--	--	--
Modified headers	--	--	IP-in-IP(RPI)	--
Untouched headers	--	--	--	--

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

[7.2.](#) Non-Storing Mode: Interaction between Leaf and Internet

In this section we are going to describe the communication flow in Non Storing Mode (Non-SM) between,

RPL-aware-leaf to Internet

Internet to RPL-aware-leaf

not-RPL-aware-leaf to Internet

Internet to not-RPL-aware-leaf

[7.2.1.](#) Non-SM: Example of Flow from RPL-aware-leaf to Internet

In this case the flow comprises:

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

For example, the communication flow could be: Node F --> Node D --> Node B --> Node A --> Internet

6LR_i are the intermediate routers from source to destination. In this case, " $1 \leq i \leq n$ ", n is the number of routers (6LR) that the packet go through from source (6LN) to 6LBR.

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_i	6LBR	Internet
Inserted headers	RPI	--	--	--
Removed headers	--	--	--	--
Re-added headers	--	--	--	--
Modified headers	--	RPI	--	--
Untouched headers	--	--	RPI	RPI (Ignored)

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

7.2.2. Non-SM: Example of Flow from Internet to RPL-aware-leaf

In this case the flow comprises:

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

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

6LR_i are the intermediate routers from source to destination. In this case, "1 <= i >= n", n is the number of routers (6LR) that the packet go through from 6LBR to destination(6LN).

The 6LBR must add an RH3 header. As the 6LBR will know the path and address of the target node, it can address the IP-in-IP 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, it is optional.

Internet-Draft

Useof6553

October 2017

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

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

[7.2.3.](#) Non-SM: Example of Flow from not-RPL-aware-leaf to Internet

In this case the flow comprises:

not-RPL-aware-leaf (IPv6) --> 6LR_1 --> 6LR_i -->root (6LBR) --> Internet

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

6LR_i are the intermediate routers from source to destination. In this case, " $1 < i \leq n$ ", n is the number of routers (6LR) that the packet go through from source(IPv6) to 6LBR. e.g 6LR_1 (i=1).

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 6LR (6LR_1) will add an RPI header inside a new IP-in-IP header. The IP-in-IP header will be addressed to the root. This case is identical to the storing-mode case ([Section 5.7](#)).

Header	IPv6	6LR_1	6LR_i [i=2,..,n]	6LBR	Internet
Inserted headers	--	IP-in-IP(RPI)	--	--	--
Removed headers	--	--	--	IP-in-IP(RPI)	--
Re-added headers	--	--	--	--	--
Modified headers	--	--	IP-in-IP(RPI)	--	--
Untouched headers	--	--	--	--	--

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

[7.2.4.](#) Non-SM: Example of Flow from Internet to not-RPL-aware-leaf

In this case the flow comprises:

Internet --> root (6LBR) --> 6LR_i --> not-RPL-aware-leaf (IPv6)

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

6LR_i are the intermediate routers from source to destination. In this case, " $1 < i \leq n$ ", n is the number of routers (6LR) that the packet go through from 6LBR to not-RPL-aware-leaf (IPv6).

The 6LBR must add an RH3 header inside an IP-in-IP 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 IP-in-IP header destination be the last 6LR. The 6LBR will set to zero the flow label upon entry in order to aid compression.

Header	Internet	6LBR	6LR ₁	6LR _i ($i=2, \dots, n$)	IPv6
Insert ed headers	--	IP-in-IP(RH3, opt:RPI)	--	--	--
Removed headers	--	--	--	IP-in-IP(RH3, RPI)	--
Re-added headers	--	--	--	--	--
Modified	--	--	IP-in-	IP-in-	--

ed headers			IP(RH3, RPI)	IP(RH3, RPI)	
Untouched headers	--	--	--	--	RPI

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

7.3. Non-Storing Mode: Interaction between Leafs

In this section we are going to describe the communication flow in Non Storing Mode (Non-SM) between,

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

7.3.1. Non-SM: Example of Flow from RPL-aware-leaf to RPL-aware-leaf

In this case the flow comprises:

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

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

6LR_ia are the intermediate routers from source to the root In this case, " $1 \leq ia \leq n$ ", n is the number of routers (6LR) that the packet go through from 6LN to the root.

6LR_id are the intermediate routers from the root to the destination. In this case, " $1 \leq id \leq m$ ", m is the number of the intermediate routers (6LR).

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 SHOULD put the RPI into an IP-in-IP header addressed to the root, so that the 6LBR can remove that header. If it does not, then additional resources are wasted on the way down to carry the useless RPI option.

The 6LBR will need to insert an RH3 header, which requires that it add an IP-in-IP header. It SHOULD be able to remove the RPI, as it was contained in an IP-in-IP 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 or scenario [Section 7.1.2](#), with the originating node acting as 6LBR.

Header	6LN src	6LR_ia	6LBR	6LR_id	6LN dst
Inserted headers	IP-in-IP(RPI1)	--	IP-in-IP(RH3 to 6LN, opt RPI2)	--	--
Removed headers	--	--	IP-in-IP(RPI1)	--	IP-in-IP(RH3, opt RPI2)
Re-added headers	--	--	--	--	--
Modified headers	--	RPI1	--	RPI2	--
Untouched headers	--	--	--	--	--

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

7.3.2. Non-SM: Example of Flow from RPL-aware-leaf to not-RPL-aware-leaf

In this case the flow comprises:

6LN --> 6LR_ia --> root (6LBR) --> 6LR_id --> not-RPL-aware (IPv6)

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

6LR_ia are the intermediate routers from source to the root. In this case, " $1 \leq ia \leq n$ ", n is the number of intermediate routers (6LR).

6LR_id are the intermediate routers from the root to the destination. In this case, " $1 \leq id \leq m$ ", m is the number of the intermediate routers (6LR).

As in the previous case, the 6LN will insert an RPI (RPI₁) header which MUST be in an IP-in-IP header addressed to the root so that the 6LBR can remove this RPI. The 6LBR will then insert an RH3 inside a new IP-in-IP header addressed to the 6LN destination node. The RPI is optional from 6LBR to 6LR_id (RPI₂).

Header	6LN	6LR_1	6LBR	6LR_id	IPv6
Insert ed headers	IP-in-IP(RPI1)	--	IP-in-IP(RH3, opt RPI_2)	--	--
Removed headers	--	--	IP-in-IP(RPI_1)	IP-in-IP(RH3, opt RPI_2)	--
Re-added headers	--	--	--	--	--
Modified headers	--	IP-in-IP(RPI_1)	--	IP-in-IP(RH3, opt RPI_2)	--
Untouched headers	--	--	--	--	opt RPI_2

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

[7.3.3.](#) Non-SM: Example of Flow from not-RPL-aware-leaf to RPL-aware-leaf

In this case the flow comprises:

not-RPL-aware 6LN (IPv6) --> 6LR_ia --> root (6LBR) --> 6LR_id --> 6LN

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

6LR_ia are the intermediate routers from source to the root In this case, " $1 \leq ia \leq n$ ", n is the number of intermediate routers (6LR)

6LR_id are the intermediate routers from the root to the destination. In this case, " $1 \leq id \leq m$ ", m is the number of the intermediate routers (6LR).

This scenario is mostly identical to the previous one. The RPI is added by the first 6LR (6LR_1) inside an IP-in-IP header addressed to the root. The 6LBR will remove this RPI, and add it's own IP-in-IP header containing an RH3 header and optional RPI (RPI_2).

Internet-Draft

Useof6553

October 2017

Header	IPv6	6LR_1	6LBR	6LR_id	6LN
Insert ed headers	--	IP-in-IP(RPI_1)	IP-in-IP(RH3, opt RPI_2)	--	--
Remove d headers	--	--	IP-in-IP(RPI_1)	--	IP-in-IP(RH3, opt RPI_2)
Re-added headers	--	--	--	--	--
Modified headers	--	--	--	IP-in-IP(RH3, opt RPI_2)	--
Untouched headers	--	--	--	--	--

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

[7.3.4.](#) Non-SM: Example of Flow from not-RPL-aware-leaf to not-RPL-aware-leaf

In this case the flow comprises:

not-RPL-aware 6LN (IPv6 src)--> 6LR_{ia} --> root (6LBR) --> 6LR_{id} --> not-RPL-aware (IPv6 dst)

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

6LR_{ia} are the intermediate routers from source to the root In this case, " $1 \leq ia \leq n$ ", n is the number of intermediate routers (6LR)

6LR_id are the intermediate routers from the root to the destination. In this case, " $1 \leq ia \leq m$ ", m is the number of the intermediate routers (6LR).

This scenario is the combination of the previous two cases.

Header	IPv6 src	6LR_1	6LBR	6LR_id	IPv6 dst
Inserted headers	--	IP-in-IP(RPI_1)	IP-in-IP(RH3)	--	--
Removed headers	--	--	IP-in-IP(RPI_1)	IP-in-IP(RH3, opt RPI_2)	--
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

8. Observations about the cases

8.1. Storing mode

[I-D.ietf-roll-routing-dispatch] shows that the hop-by-hop IP-in-IP header can be compressed using IP-in-IP 6LoRH (IP-in-IP-6LoRH) header as described in [Section 7](#) of the document.

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

Thanks to the change of the RPI option type from 0x63 to 0x23, there is no longer any uncertainty about when to use an IP-in-IP header in the storing mode. A Hop-by-Hop Options Header containing the RPI option SHOULD always be added when 6LRs originate packets (without IP-in-IP headers), and IP-in-IP headers should always be added (addressed to the root when on the way up, to the end-host when on the way down) when a 6LR find that it needs to insert a Hop-by-Hop Options Header containing the RPI option.

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

[8.2.](#) Non-Storing mode

In 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 DODAG 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 IP-in-IP headers.

Unlike in the storing mode case, there is 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.

[9.](#) 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.

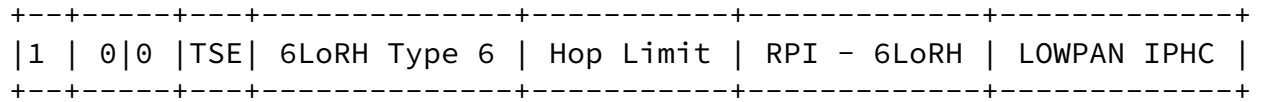


Figure 9: Critical IP-in-IP (RPI).

10. IANA Considerations

This document updates the registration made in [[RFC6553](#)] Destination Options and Hop-by-Hop Options registry from 0x63 to 0x23.

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

Figure 10: Option Type in RPL Option.

We propose to use DODAG Configuration Option Flags in the DODAG Configuration option as follows:

```

+-----+-----+-----+
| Bit number | Description | Reference |
+-----+-----+-----+
|      3     | RPI 0x23 enable | This document |
+-----+-----+-----+

```

Figure 11: DODAG Configuration Option Flag to indicate the RPI-flag-day.

11. Security Considerations

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

The IPIP 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.

Nodes outside of the LLN will need to pass IPIP traffic through the RPL root to perform this attack. To counter, the RPL root SHOULD either restrict ingress of IPIP packets (the simpler solution), or it SHOULD do a deep packet inspection wherein it walks the IP header extension chain until it can inspect the upper-layer-payload as described in [\[RFC7045\]](#). In particular, the RPL root SHOULD do [BCP38](#) ([\[RFC2827\]](#)) processing 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 described in [\[I-D.ietf-6lo-backbone-router\]](#). In this case the [BCP38](#) filtering needs to take this into account.

Nodes with the LLN can use the IPIP 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 Distributed Denial Of Service attack upon nodes elsewhere in the Internet. See [\[DDOS-KREBS\]](#) for an example of such attacks already seen in the real world.

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, they could still have a significant

impact, particularly if the attack was on another LLN! Additionally, some uses of RPL involve large backbone ISP scale equipment [[I-D.ietf-anima-autonomic-control-plane](#)], which may be equipped with multiple 100Gb/s interfaces.

Blocking or careful filtering of IPIP traffic entering the LLN as described above will make sure that any attack that is mounted must originate compromised nodes within the LLN. The use of [BCP38](#) filtering at the RPL root on egress traffic will both 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, [BCP38](#) filtering involves a single prefix comparison and should be trivial to automatically configure.

There are some scenarios where IPIP traffic SHOULD be allowed to pass through the RPL root, such as the IPIP mediated communications between a new Pledge and the Join Registrar/Coordinator (JRC) when using [[I-D.ietf-anima-bootstrapping-keyinfra](#)] and [[I-D.ietf-6tisch-dtsecurity-secure-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 IPIP tunnels will 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 IPIP headers using forged source addresses. If the attack requires bi-directional communication, then IPIP provides no advantages.

[RFC2473] suggests that tunnel entry and exit points can be secured, via the "Use IPsec". This solution has all the problems that [[RFC5406](#)] goes into. In an LLN such a solution would degenerate into every node having a tunnel with every other node. It would provide a small amount of origin address authentication at a very high cost;

doing [BCP38](#) at every node (linking layer-3 addresses to layer-2 addresses, and to already present layer-2 cryptographic mechanisms) would be cheaper should RPL be run in an environment where hostile nodes are likely to be a part of the LLN.

The RH3 header usage described here can be abused in equivalent ways with an IPIP header to add the needed RH3 header. As such, the attacker's RH3 header will not be seen by the network until it reaches the end host, which will decapsulate it. An end-host SHOULD be suspicious about a RH3 header which has additional hops which have not yet been processed, and SHOULD ignore such a second RH3 header.

In addition, the LLN will likely use [[I-D.ietf-roll-routing-dispatch](#)] to compress the IPIP and RH3 headers. As such, the compressor at the RPL-root will see the second RH3 header and MAY choose to discard the packet if the RH3 header has not been completely consumed. A consumed (inert) RH3 header could be present in a packet that flows from one LLN, crosses the Internet, and enters another LLN. As 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. It should just be noted that an incoming RH3 must be fully consumed, or very carefully inspected.

The RPI header, if permitted to enter the LLN, could be used by an attacker to change the priority of a packet by selecting a different RPL instanceID, 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 instanceID, but a change of instanceID would permit an attacker to bypass such filtering. Like the RH3, an RPI header is to be inserted by the RPL root on traffic entering the LLN by first inserting an IPIP header. The attacker's RPI header therefore will not be seen by the network. Upon reaching the destination node the RPI header has no further meaning and is just skipped; the presence of a second RPI header will have no meaning to the end node as the packet has already been identified as being at its final destination.

The RH3 and RPI headers could be abused by an attacker inside of the network to route packets on non-obvious ways, perhaps eluding observation. This usage is in fact part of [[RFC6997](#)] and can not 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 IPIP headers, and this document does not change that analysis.

12. 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 (alphabetical order): Robert Cragie, Simon Duquennoy, Ralph Droms, Cenk Guendogan, C. M. Heard, Rahul Jadhav, Matthias Kovatsch, Peter van der Stok, Xavier Vilajosana and Thomas Watteyne.

13. References

13.1. Normative References

- [I-D.ietf-6man-rfc2460bis]
Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", [draft-ietf-6man-rfc2460bis-13](#) (work in progress), May 2017.

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

- [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>>.

- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", [BCP 38](#), [RFC 2827](#), DOI 10.17487/RFC2827, May 2000, <<https://www.rfc-editor.org/info/rfc2827>>.

- [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>>.

- [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>>.

Internet-Draft

Useof6553

October 2017

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

13.2. Informative References

- [DDOS-KREBS]
Goodin, D., "Record-breaking DDoS reportedly delivered by >145k hacked cameras", September 2016, <<http://arstechnica.com/security/2016/09/botnet-of-145k-cameras-reportedly-deliver-internets-biggest-ddos-ever/>>.
- [I-D.ietf-6lo-backbone-router]
Thubert, P., "IPv6 Backbone Router", [draft-ietf-6lo-backbone-router-04](#) (work in progress), July 2017.

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

Thubert, P., "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4", [draft-ietf-6tisch-architecture-12](#) (work in progress), August 2017.

Robles, et al.

Expires April 30, 2018

[Page 46]

Internet-Draft

Useof6553

October 2017

[I-D.ietf-6tisch-dtsecurity-secure-join]

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

[I-D.ietf-anima-autonomic-control-plane]

Behringer, M., Eckert, T., and S. Bjarnason, "An Autonomic Control Plane (ACP)", [draft-ietf-anima-autonomic-control-plane-12](#) (work in progress), October 2017.

[I-D.ietf-anima-bootstrapping-keyinfra]

Pritikin, M., Richardson, M., Behringer, M., Bjarnason, S., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructures (BRSKI)", [draft-ietf-anima-bootstrapping-keyinfra-08](#) (work in progress), October 2017.

[I-D.ietf-roll-dao-projection]

Thubert, P. and J. Pylakutty, "Root initiated routing state in RPL", [draft-ietf-roll-dao-projection-02](#) (work in progress), September 2017.

[I-D.ietf-roll-routing-dispatch]

Thubert, P., Bormann, C., Toutain, L., and R. Cragie, "6LoWPAN Routing Header", [draft-ietf-roll-routing-dispatch-05](#) (work in progress), October 2016.

[RFC4192] Baker, F., Lear, E., and R. Droms, "Procedures for Renumbering an IPv6 Network without a Flag Day", [RFC 4192](#), DOI 10.17487/RFC4192, September 2005, <<https://www.rfc-editor.org/info/rfc4192>>.

[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>>.

[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>>.

Robles, et al.

Expires April 30, 2018

[Page 47]

Internet-Draft

Useof6553

October 2017

[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>>.

[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/mcr/>

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