ROLL Working Group   M. Robles
Internet-Draft   Ericsson
Intended status: Informational   M. Richardson
Expires: January 21, 2017   SSW
                                                          P. Thubert
                                                              Cisco
                                                      July 20, 2016

When to use RFC 6553, 6554 and IPv6-in-IPv6
draft-ietf-roll-useofrplinfo-07

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.

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 January 21, 2017.

Copyright Notice

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

   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 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], 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.

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. 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
[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.

```
+--------------+
| Upper Layers |
|              |
+--------------+
|    RPL       |
|              |
+--------------+
|    ICMPv6    |
|              |
+--------------+
|    IPv6      |
|              |
+--------------+
|    6LoWPAN   |
|              |
+--------------+
|    PHY-MAC   |
|              |
+--------------+
```

Figure 1: RPL Stack.

```
                                   +---------+
                             +---+Internet |
                             |   +---------+
                             |
                      +----+--+
                      | DODAG |  node:01
                +---------+ Root  +----------+
                |        | 6LBR  |          |
                |        +----+--+          |
                |            |              |
                |            |              |
               ...          ...            ...
                |            |              |
           +-----+-+      +--+---+      +--+---+
           |6LR    |      |      |      |      |
       +-----+     |      |      |      |      |
       |     |  11 |      |  12 |      |  13 +------+
       |     +-----+-+    +-+----+      +-+----+    |
       |         |            |            |        |
       |         |            |            |        |
       | 21      | 22         | 23         | 24     | 25
      +-+---+   +-+---+     +--+--+      +- --+    +---+-+
      |Leaf |   |     |     |     |      |Leaf|    |Leaf |
      | 6LN |   |     |     |     |      | 6LN|    | 6LN |
      +-----+   +-----+     +-----+      +----+    +-----+
```
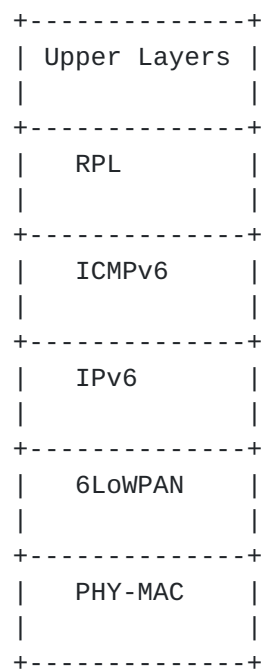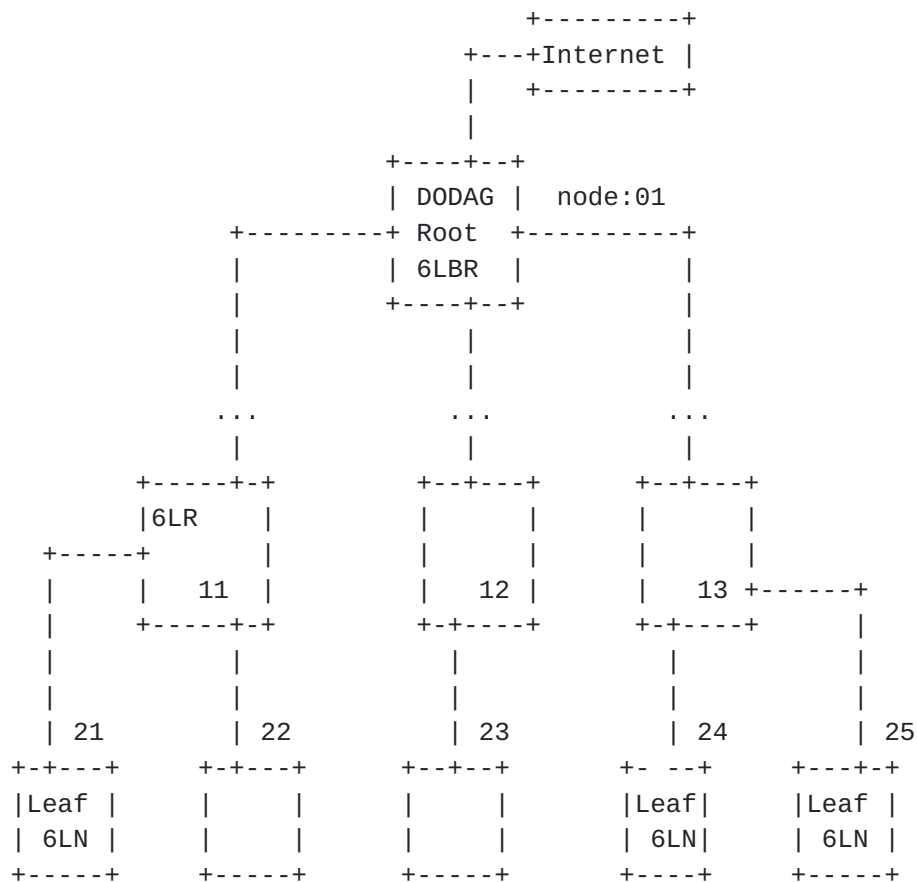
Figure 2: A reference RPL Topology.

In Figure 2 is showed the reference RPL Topology for this document.
The numbers in or above the nodes are there so that they may be
referenced in subsequent sections. In the figure, a 6LN can be a
router or a host. The 6LN leafs marked as (21) and (25) are routers.
The leaf marked 6LN (24) is a device which does 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 this leaf (24) is often named IPv6 node. The 6LBR in the
figure is the root of the Global DODAG.

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.

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.

This version of the document assumes the changes in [I-D.ietf-6man-rfc2460bis] are passed.

> 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 (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] is that Extensions may not be added or removed except by the sender or the receiver. (A revision to RFC2460 considered changing this rule, but has kept it)

But, options in the Hop-by-Hop option which are marked with option type 01 ([RFC2460] section 4.2 and [I-D.ietf-6man-rfc2460bis]) SHOULD be ignored when 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) will NOT be discarded, even if it

has the [RFC6553] RPL Option Header known as the RPI or [RFC6554]
SRH3 Extension Header (S)RH3.

With abolition of one of these rules it means that the RPI Hop-by-Hop
option MAY be left in place even if the end host does host understand
it. This collapses many of the cases above (where it says "or")

An intermediate router that needs to add an extension header (SHR3 or
RPI Option) 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
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. 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.

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 consise.

5. Storing mode

In storing mode (fully stateful), the sender cannot determine whether
the destination is RPL-capable and thus would need an IP-in-IP
header. The IP-in-IP header needs to be addressed on a hop-by-hop
basis so that the last 6LR can remove the RPI header. Additionally,
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 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 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.

Use Case	RPI	RH3	IP-in-IP	IP-in-IP 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

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

In this case the flow comprises:

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

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

The 6LN inserts the RPI header, and sends the packet to 6LR which
decrements the rank in RPI and sends the packet up. When the packet
arrives at 6LBR, 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 | 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 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   | 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)

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

```
+-------------------+------+-----+------+
| Header            | 6LBR | 6LR | IPv6 |
+-------------------+------+-----+------+
| Inserted headers  | --   | --  | --   |
| Removed headers   | --   | --  | --   |
| 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 processes the packet.

```
+-------------------+------+---------------+--------------+
| Header            | IPv6 | 6LR           | 6LBR         |
+-------------------+------+---------------+--------------+
| Inserted headers  | --   | IP-in-IP(RPI) | --           |
| Removed headers   | --   | --            | IP-in-IP(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 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 --> root (6LBR) --> Internet

   No IP-in-IP header is required.

```
+-------------------+------+------+------+----------+
| Header            | 6LN  | 6LR  | 6LBR | Internet |
+-------------------+------+------+------+----------+
| Inserted headers  | RPI  | --   | --   | --       |
| Removed headers   | --   | --   | --   | --       |
| 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 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.

```
+-----------------+---------+--------------+------+--------------+
| Header          | Internet | 6LBR        | 6LR  | 6LN          |
+-----------------+---------+--------------+------+--------------+
| Inserted        | --      | IP-in-IP(RPI) | --  | --           |
| headers         |         |              |      |              |
| Removed headers | --      | --           | --   | IP-in-IP(RPI) |
| Re-added        | --      | --           | --   | --           |
| headers         |         |              |      |              |
| Modified        | --      | --           | RPI  | --           |
| headers         |         |              |      |              |
| 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 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 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        | --   | IP-in-IP(RPI) | --          | --       |
| headers         |      |              |              |          |
| Removed headers | --   | --           | IP-in-IP(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 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, as the packet will not emerge again
   from the LLN.

```
+-----------------+----------+--------------+--------------+------+
| Header          | Internet | 6LBR         | 6LR          | IPv6 |
+-----------------+----------+--------------+--------------+------+
| Inserted        | --       | IP-in-IP(RPI)| --           | --   |
| headers         |          |              |              |      |
| Removed headers | --       | --           | IP-in-IP(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 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   | 6LR          | 6LR (common  | 6LR | 6LN   |
|             | src   |              | parent)      |     | dst   |
+-------------+-------+--------------+--------------+-----+-------+
| Inserted    | RPI   | --           | --           | --  | --    |
| headers     |       |              |              |     |       |
| Removed     | --    | --           | --           | --  | RPI   |
| headers     |       |              |              |     |       |
| Re-added    | --    | --           | --           | --  | --    |
| headers     |       |              |              |     |       |
| Modified    | --    | RPI          | RPI          | --  | --    |
| headers     |       | (decreasing  | (increasing  |     |       |
|             |       | rank)        | 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

This situation is identical to the situation Section 5.9

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 IP-in-IP
header is addressed to the destinion 6LN.

Header	IPv6	6LR	common parent (6LR)	6LR	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

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 is identical to Section 5.11

6. Non Storing mode

```
+--------------+------+------+----------+--------------+
| Use Case     | RPI  | RH3  | IP-in-IP | IP-in-IP 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 IP-in-IP header is necessary as the traffic originates with an RPL
aware node, the 6LBR. The destination is known to 6LBR because, the
root knows the whole topology in non-storing mode.

```
+-------------------+----------------+------+----------+
| 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, modified in each intermediate 6LR and it is
   fully consumed in the last 6LR, 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 IP-in-IP 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 IP-in-IP
   header to the last 6LR.

   Omitting the RPI entirely is therefore a better solution, as no IP-
   in-IP 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
   IP-in-IP 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  | --   | IP-in-IP(RPI) | --   | --            |
| Removed headers   | --   | --            | --   | IP-in-IP(RPI) |
| Re-added headers  | --   | --            | --   | --            |
| Modified headers  | --   | --            | --   | --            |
| Untouched headers | --   | IP-in-IP(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 removed by the 6LBR.
   The 6LN must therefore add the RPI inside an IP-in-IP 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        | IP-in-IP(RPI)| --   | --           | --       |
| headers         |              |      |              |          |
| Removed headers | --           | --   | IP-in-IP(RPI)| --       |
| Re-added        | --           | --   | --           | --       |
| headers         |              |      |              |          |
| Modified        | --           | --   | --           | --       |
| headers         |              |      |              |          |
| Untouched       | --           | RPI  | --           | --       |
| headers         |              |      |              |          |
+-----------------+--------------+------+--------------+----------+
```

          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 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.

```
+----------+----------+----------------------+--------------+-----+
| Header   | Internet | 6LBR                 | 6LR          | 6LN |
+----------+----------+----------------------+--------------+-----+
| Inserted | --       | IP-in-IP(RH3,opt:RPI)| --           | --  |
| headers  |          |                      |              |     |
| Removed  | --       | --                   | IP-in-IP(RH3)| --  |
| headers  |          |                      |              |     |
| Re-added | --       | --                   | --           | --  |
| headers  |          |                      |              |     |
| Modified | --       | --                   | IP-in-IP(RH3)| --  |
| headers  |          |                      |              |     |
| Untouche | --       | --                   | --           | --  |
| d        |          |                      |              |     |
| 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 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          | 6LBR          | Internet |
+-----------------+------+--------------+---------------+----------+
| Inserted        | --   | IP-in-IP(RPI)| --            | --       |
| headers         |      |              |               |          |
| Removed headers | --   | --           | IP-in-IP(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 (IPv6 node)

   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   | Interne | 6LBR                 | 6LR          | IPv6 |
|          | t       |                      |              |      |
+----------+---------+----------------------+--------------+------+
| Inserted | --      | IP-in-IP(RH3,opt:RPI)| --           | --   |
| headers  |         |                      |              |      |
| Removed  | --      | --                   | IP-in-IP(RH3,| --   |
| headers  |         |                      | RPI)         |      |
| Re-added | --      | --                   | --           | --   |
| headers  |         |                      |              |      |
| Modified | --      | --                   | --           | --   |
| headers  |         |                      |              |      |
| Untouche | --      | --                   | --           | --   |
| d        |         |                      |              |      |
| 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 IP-in-IP 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 IP-in-IP header. It may be able to remove the RPI if 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.

```
+----------+--------------+-------------+-----+------------------+
| Header   | 6LN src      | 6LBR        | 6LR | 6LN dst          |
+----------+--------------+-------------+-----+------------------+
| Inserted | IP-in-IP(RPI)| IP-in-IP(RH3| --  | --               |
| headers  |              | to 6LN,RPI) |     |                  |
| Removed  | --           | --          | --  | IP-in-IP(RH3,RPI) |
| headers  |              |             |     |                  |
| Re-added | --           | --          | --  | --               |
| headers  |              |             |     |                  |
| Modified | --           | --          | --  | --               |
| headers  |              |             |     |                  |
| Untouche | --           | --          | --  | --               |
| d        |              |             |     |                  |
| 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 (IPv6 node)

   As in the previous case, the 6LN will insert an RPI 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 above the destination node.

```
+----------+--------------+--------------+---------------+------+
| Header   | 6LN          | 6LBR         | 6LR           | IPv6 |
+----------+--------------+--------------+---------------+------+
| Inserted | IP-in-IP(RPI)| IP-in-IP(RH3,| --            | --   |
| headers  |              | opt RPI)     |               |      |
| Removed  | --           | IP-in-IP(RPI)| IP-in-IP(RH3, | --   |
| headers  |              |              | 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 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.

```
+------------+------+--------------+--------------+--------------+
| Header     | IPv6 | 6LR          | 6LBR         | 6LN          |
+------------+------+--------------+--------------+--------------+
| Inserted   | --   | IP-in-IP(RPI)| IP-in-IP(RH3)| --           |
| headers    |      |              |              |              |
| Removed    | --   | IP-in-IP(RPI)| --           | IP-in-IP(RH3)|
| headers    |      |              |              |              |
| 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
   (IPv6 node)

   This scenario is the combination of the previous two cases.

Header	IPv6	6LR	6LBR	6LR	IPv6
Inserted headers	--	IP-in-IP(RPI)	IP-in-IP(RH3)	--	--
Removed headers	--	--	IP-in-IP(RPI)	IP-in-IP(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
IP-in-IP encapsulation, and it can never omit the IP-in-IP
encapsulation. See table Table 1

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

There are potential significant advantages to having a single code
path that always processes IP-in-IP 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 IP-in-IP 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 IP-in-IP
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 IP-in-IP vs never use IP-in-IP)
should be signaled in the RPL protocol itself.

7.2. 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.

It is unclear what it would mean for an RH3 header to be present in a
hop-by-hop IP-in-IP header. The receiving node ought to consume the
IP-in-IP 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 (because they do not save the
sate), 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 IP-in-IP header can NOT
be used.

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

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.

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.

```
+--+-----+---+-------------+----------+------------+-------------+
|1 | 0|0 |TSE| 6LoRH Type 6 | Hop Limit | RPI - 6LoRH | LOWPAN IPHC |
+--+-----+---+-------------+----------+------------+-------------+
```

Figure 3: 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

12. References

12.1. Normative References

[I-D.ietf-6man-rfc2460bis]
           Deering, D. and R. Hinden, "Internet Protocol, Version 6
           (IPv6) Specification", draft-ietf-6man-rfc2460bis-05 (work
           in progress), June 2016.

[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-10 (work
           in progress), June 2016.

[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.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet
           Control Message Protocol (ICMPv6) for the Internet
           Protocol Version 6 (IPv6) Specification", RFC 4443,
           DOI 10.17487/RFC4443, March 2006,
           <http://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,
           <http://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,
           <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