Internet Engineering Task Force (IETF)
Request for Comments: 8138
Category: Standards Track
ISSN: 2070-1721

P. Thubert, Ed.
Cisco
C. Bormann
Uni Bremen TZI
L. Toutain
IMT Atlantique
R. Cragie
ARM

April 2017

# IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing Header

## Abstract

This specification introduces a new IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) dispatch type for use in 6LoWPAN route-over topologies, which initially covers the needs of Routing Protocol for Low-Power and Lossy Networks (RPL) data packet compression (RFC 6550). Using this dispatch type, this specification defines a method to compress the RPL Option (RFC 6553) information and Routing Header type 3 (RFC 6554), an efficient IP-in-IP technique, and is extensible for more applications.

## Status of This Memo

This is an Internet Standards Track document.

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

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc8138.

Table of Contents

# 1. Introduction

The design of Low-Power and Lossy Networks (LLNs) is generally focused on saving energy, a very constrained resource in most cases. The other constraints, such as the memory capacity and the duty cycling of the LLN devices, derive from that primary concern. Energy is often available from primary batteries that are expected to last for years, or it is scavenged from the environment in very limited quantities. Any protocol that is intended for use in LLNs must be designed with the primary concern of saving energy as a strict requirement.

Controlling the amount of data transmission is one possible venue to save energy. In a number of LLN standards, the frame size is limited to much smaller values than the guaranteed IPv6 Maximum Transmission Unit (MTU) of 1280 bytes. In particular, an LLN that relies on the classical Physical Layer (PHY) of IEEE 802.15.4 [IEEE.802.15.4] is limited to 127 bytes per frame. The need to compress IPv6 packets over IEEE 802.15.4 led to the writing of "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks" [RFC6282].

Innovative route-over techniques have been and still are being developed for routing inside an LLN. Generally, such techniques require additional information in the packet to provide loop prevention and to indicate information such as flow identification, source routing information, etc.

For reasons such as security and the capability to send ICMPv6 errors (see "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification" [RFC4443]) back to the source, an original packet must not be tampered with, and any information that must be inserted in or removed from an IPv6 packet must be placed in an extra IP-in-IP encapsulation.

This is the case when the additional routing information is inserted by a router on the path of a packet, for instance, the root of a mesh, as opposed to the source node, with the Non-Storing mode of the "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks" [RFC6550].

This is also the case when some routing information must be removed from a packet that flows outside the LLN.

"When to use RFC 6553, RFC 6554 and IPv6-in-IPv6" [RPL-INFO] details different cases where IPv6 headers defined in the RPL Option for Carrying RPL Information in Data-Plane Datagrams [RFC6553], Header for Source Routes with RPL [RFC6554], and IPv6-in-IPv6 encapsulation, are inserted or removed from packets in LLN environments operating RPL.

When using RFC 6282 [RFC6282], the outer IP header of an IP-in-IP encapsulation may be compressed down to 2 octets in stateless compression and down to 3 octets in stateful compression when context information must be added.

```
   0                                       1
   0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5
 +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 | 0 | 1 | 1 |  TF   |NH | HLIM  |CID|SAC|  SAM  | M |DAC|  DAM  |
 +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 1: LOWPAN_IPHC Base Encoding (RFC 6282)

The stateless compression of an IPv6 address can only happen if the IPv6 address can de deduced from the Media Access Control (MAC) addresses, meaning that the IP endpoint is also the MAC-layer endpoint.  This is usually not the case in a RPL network, which is generally a multi-hop route-over (i.e., operated at Layer 3) network. A better compression, which does not involve variable compressions depending on the hop in the mesh, can be achieved based on the fact that the outer encapsulation is usually between the source (or destination) of the inner packet and the root.  Also, the inner IP header can only be compressed by RFC 6282 [RFC6282] if all the fields preceding it are also compressed.  This specification makes the inner IP header the first header to be compressed by RFC 6282 [RFC6282], and it keeps the inner packet encoded the same way whether or not it is encapsulated, thus preserving existing implementations.

As an example, RPL [RFC6550] is designed to optimize the routing operations in constrained LLNs.  As part of this optimization, RPL requires the addition of RPL Packet Information (RPI) in every packet, as defined in Section 11.2 of RFC 6550 [RFC6550].

"The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams" [RFC6553] specification indicates how the RPI can be placed in a RPL Option (RPL-OPT) that is placed in an IPv6 Hop-by-Hop header.

This representation demands a total of 8 bytes, while, in most cases, the actual RPI payload requires only 19 bits.  Since the Hop-by-Hop header must not flow outside of the RPL domain, it must be inserted in packets entering the domain and be removed from packets that leave the domain. In both cases, this operation implies an IP-in-IP encapsulation.

Additionally, in the case of the Non-Storing Mode of Operation (MOP), RPL requires a Source Routing Header (SRH) in all packets that are routed down a RPL graph. For that purpose, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)" [RFC6554] defines the type 3 Routing Header for IPv6 (RH3).

```
          ------+---------                           ^
                |          Internet                  |
                |                                    | Native IPv6
             +-----+                                 |
             |     | Border Router (RPL Root)      + | +
             |     |                               | | |
             +-----+                               | | | tunneled
                |                                  | | | using
          o    o   o    o                          | | | IPv6-in-
      o o   o  o   o  o  o o   o                   | | | IPv6 and
     o  o o  o o    o   o   o  o  o                | | | RPL SRH
     o   o    o  o     o  o    o  o  o             | | |
    o  o    o o   o         o   o o                | | |
    o          o             o     o               + v +
                      LLN
```

Figure 2: IP-in-IP Encapsulation within the LLN

With Non-Storing RPL, even if the source is a node in the same LLN, the packet must first reach up the graph to the root so that the root can insert the SRH to go down the graph. In any fashion, whether the packet was originated in a node in the LLN or outside the LLN, and regardless of whether or not the packet stays within the LLN, as long as the source of the packet is not the root itself, the source-routing operation also implies an IP-in-IP encapsulation at the root in order to insert the SRH.

"An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4" [IPv6-ARCH] specifies the operation of IPv6 over the mode of operation described in "Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement" [RFC7554]. The architecture requires the use of both RPL and the 6lo adaptation layer over IEEE 802.15.4. Because it inherits the constraints on frame size from the MAC layer, 6TiSCH cannot afford to allocate 8 bytes per packet on the RPI, hence the requirement for 6LoWPAN header compression of the RPI.

An extensible compression technique is required that simplifies IP-in-IP encapsulation when it is needed and optimally compresses existing routing artifacts found in RPL LLNs.

This specification extends the 6lo adaptation layer framework ([RFC4944] [RFC6282]) so as to carry routing information for route-over networks based on RPL. It includes the formats necessary for RPL and is extensible for additional formats.

## 2. Terminology

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

This document uses the terms from, and is consistent with, "Terms Used in Routing for Low-Power and Lossy Networks" [RFC7102] and RPL [RFC6550].

The terms "route-over" and "mesh-under" are defined in RFC 6775 [RFC6775].

Other terms in use in LLNs are found in "Terminology for Constrained-Node Networks" [RFC7228].

The term "byte" is used in its now customary sense as a synonym for "octet".

## 3. Using the Page Dispatch

The "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Paging Dispatch" [RFC8025] specification extends the 6lo adaptation layer framework ([RFC4944] [RFC6282]) by introducing a concept of "context" in the 6LoWPAN parser, a context being identified by a Page number. The specification defines 16 Pages.

This document operates within Page 1, which is indicated by a dispatch value of binary 11110001.

### 3.1. New Routing Header Dispatch (6LoRH)

This specification introduces a new 6LoWPAN Routing Header (6LoRH) to carry IPv6 routing information. The 6LoRH may contain source routing information such as a compressed form of SRH, as well as other sorts of routing information such as the RPI and IP-in-IP encapsulation.

The 6LoRH is expressed in a 6loWPAN packet as a Type-Length-Value (TLV) field, which is extensible for future use.

It is expected that a router that does not recognize the 6LoRH general format detailed in Section 4 will drop the packet when a 6LoRH is present.

This specification uses the bit pattern 10xxxxxx in Page 1 for the new 6LoRH Dispatch. Section 4 describes how RPL artifacts in data packets can be compressed as 6LoRH headers.

## 3.2. Placement of 6LoRH Headers

### 3.2.1. Relative to Non-6LoRH Headers

In a zone of a packet where Page 1 is active (that is, once the Page 1 Paging Dispatch is parsed, and until another Paging Dispatch is parsed as described in the 6LoWPAN Paging Dispatch specification [RFC8025]), the parsing of the packet MUST follow this specification if the 6LoRH Bit Pattern (see Section 3.1) is found.

With this specification, the 6LoRH Dispatch is only defined in Page 1, so it MUST be placed in the packet in a zone where the Page 1 context is active.

Because a 6LoRH header requires a Page 1 context, it MUST always be placed after any Fragmentation Header and/or Mesh Header as defined in RFC 4944 [RFC4944].

A 6LoRH header MUST always be placed before the LOWPAN_IPHC as defined in RFC 6282 [RFC6282]. It is designed in such a fashion that placing or removing a header that is encoded with 6LoRH does not modify the part of the packet that is encoded with LOWPAN_IPHC, whether or not there is an IP-in-IP encapsulation. For instance, the final destination of the packet is always the one in the LOWPAN_IPHC, whether or not there is a Routing Header.

### 3.2.2. Relative to Other 6LoRH Headers

The "Internet Protocol, Version 6 (IPv6) Specification" [RFC2460] defines chains of headers that are introduced by an IPv6 header and terminated by either another IPv6 header (IP-in-IP) or an Upper-Layer Protocol (ULP) header. When an outer header is stripped from the packet, the whole chain goes with it. When one or more headers are inserted by an intermediate router, that router normally chains the headers and encapsulates the result in IP-in-IP.

With this specification, the chains of headers MUST be compressed in the same order as they appear in the uncompressed form of the packet. This means that if there is more than one nested IP-in-IP encapsulation, the first IP-in-IP encapsulation, with all its chain of headers, is encoded first in the compressed form.

In the compressed form of a packet that has a Source Route or a Hop-by-Hop (HbH) Options Header [RFC2460] after the inner IPv6 header (e.g., if there is no IP-in-IP encapsulation), these headers are placed in the 6LoRH form before the 6LOWPAN_IPHC that represents the IPv6 header (see Section 3.2.1). If this packet gets encapsulated and some other SRH or HbH Options Headers are added as part of the encapsulation, placing the 6LoRH headers next to one another may present an ambiguity on which header belongs to which chain in the uncompressed form.

In order to disambiguate the headers that follow the inner IPv6 header in the uncompressed form from the headers that follow the outer IP-in-IP header, it is REQUIRED that the compressed IP-in-IP header is placed last in the encoded chain. This means that the 6LoRH headers that are found after the last compressed IP-in-IP header are to be inserted after the IPv6 header that is encoded with the 6LOWPAN_IPHC when decompressing the packet.

With regard to the relative placement of the SRH and the RPI in the compressed form, it is a design point for this specification that the SRH entries are consumed as the packet progresses down the LLN (see Section 5.3). In order to make this operation simpler in the compressed form, it is REQUIRED that in the compressed form, the addresses along the source route path are encoded in the order of the path, and that the compressed SRH are placed before the compressed RPI.

## 4. 6LoWPAN Routing Header General Format

The 6LoRH uses the Dispatch Value Bit Pattern of 10xxxxxx in Page 1.

The Dispatch Value Bit Pattern is split in two forms of 6LoRH:

Elective (6LoRHE), which may skipped if not understood

Critical (6LoRHC), which may not be ignored

For each form, a Type field is used to encode the type of 6LoRH.

Note that there is a different registry for the Type field of each form of 6LoRH.

This means that a value for the Type that is defined for one form of 6LoRH may be redefined in the future for the other form.

### 4.1. Elective Format

The 6LoRHE uses the Dispatch Value Bit Pattern of 101xxxxx. A 6LoRHE may be ignored and skipped in parsing. If it is ignored, the 6LoRHE is forwarded with no change inside the LLN.

```
  0                   1
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-       ...        -+
 |1|0|1| Length  |      Type     |                    |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-       ...        -+
                                 <--    Length    -->
```

Figure 3: Elective 6LoWPAN Routing Header

Length: Length of the 6LoRHE expressed in bytes, excluding the first 2 bytes. This enables a node to skip a 6LoRHE header that it does not support and/or cannot parse, for instance, if the Type is not recognized.

Type: Type of the 6LoRHE

### 4.2. Critical Format

The 6LoRHC uses the Dispatch Value Bit Pattern of 100xxxxx.

A node that does not support the 6LoRHC Type MUST silently discard the packet.

Note: A situation where a node receives a message with a Critical 6LoWPAN Routing Header that it does not understand should not occur and is an administrative error, see Section 8.

```
  0                   1
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-              ...              -+
 |1|0|0|   TSE   |      Type     |                                 |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-              ...              -+
                                 <-- Length implied by Type/TSE -->
```

Figure 4: Critical 6LoWPAN Routing Header

Type-Specific Extension (TSE): The meaning depends on the Type, which must be known in all of the nodes. The interpretation of the TSE depends on the Type field that follows. For instance, it may be used to transport control bits, the number of elements in an array, or the length of the remainder of the 6LoRHC expressed in a unit other than bytes.

Type: Type of the 6LoRHC

### 4.3. Compressing Addresses

The general technique used in this document to compress an address is first to determine a reference that has a long prefix match with this address and then elide that matching piece. In order to reconstruct the compressed address, the receiving node will perform the process of coalescence described in Section 4.3.1.

One possible reference is the root of the RPL Destination-Oriented Directed Acyclic Graph (DODAG) that is being traversed. It is used by 6LoRH as the reference to compress an outer IP header in case of an IP-in-IP encapsulation. If the root is the source of the packet, this technique allows one to fully elide the source address in the compressed form of the IP header. If the root is not the encapsulator, then the Encapsulator Address may still be compressed using the root as a reference. How the address of the root is determined is discussed in Section 4.3.2.

Once the address of the source of the packet is determined, it becomes the reference for the compression of the addresses that are located in compressed SRH headers that are present inside the IP-in-IP encapsulation in the uncompressed form.

#### 4.3.1. Coalescence

An IPv6 compressed address is coalesced with a reference address by overriding the N rightmost bytes of the reference address with the compressed address, where N is the length of the compressed address, as indicated by the Type of the SRH-6LoRH header in Figure 7.

The reference address MAY be a compressed address as well, in which case, it MUST be compressed in a form that is of an equal or greater length than the address that is being coalesced.

A compressed address is expanded by coalescing it with a reference address. In the particular case of a Type 4 SRH-6LoRH, the address is expressed in full and the coalescence is a complete override as illustrated in Figure 5.

```
   RRRRRRRRRRRRRRRRRRR  A reference address, which may be
                        compressed or not

               CCCCCCC  A compressed address, which may be
                        shorter or the same as the reference

   RRRRRRRRRRRRCCCCCCC  A coalesced address, which may be the
                        same compression as the reference
```

Figure 5: Coalescing Addresses

### 4.3.2. DODAG Root Address Determination

Stateful address compression requires that some state is installed in the devices to store the compression information that is elided from the packet. That state is stored in an abstract context table, and some form of index is found in the packet to obtain the compression information from the context table.

With RFC 6282 [RFC6282], the state is provided to the stack by the 6LoWPAN Neighbor Discovery Protocol (NDP) [RFC6775]. NDP exchanges the context through the 6LoWPAN Context Option in Router Advertisement (RA) messages. In the compressed form of the packet, the context can be signaled in a Context Identifier Extension.

With this specification, the compression information is provided to the stack by RPL, and RPL exchanges it through the DODAGID field in the DAG Information Object (DIO) messages, as described in more detail below. In the compressed form of the packet, the context can be signaled by the RPLInstanceID in the RPI.

With RPL [RFC6550], the address of the DODAG root is known from the DODAGID field of the DIO messages. For a Global Instance, the RPLInstanceID that is present in the RPI is enough information to identify the DODAG that this node participates with and its associated root. But, for a Local Instance, the address of the root MUST be explicit, either in some device configuration or signaled in the packet, as the source or the destination address, respectively.

When implicit, the address of the DODAG root MUST be determined as follows:

> If the whole network is a single DODAG, then the root can be well-known and does not need to be signaled in the packets. But, since RPL does not expose that property, it can only be known by a configuration applied to all nodes.

Else, the router that encapsulates the packet and compresses it with this specification MUST also place an RPI in the packet as prescribed by RPL to enable the identification of the DODAG. The RPI must be present even in the case when the router also places an SRH header in the packet.

It is expected that the RPL implementation maintains an abstract context table, indexed by Global RPLInstanceID, that provides the address of the root of the DODAG that this node participates in for that particular RPL Instance.

# 5. The SRH-6LoRH Header

## 5.1. Encoding

A Source Routing Header 6LoRH (SRH-6LoRH) provides a compressed form for the SRH, as defined in RFC 6554 [RFC6554], for use by RPL routers.

One or more SRH-6LoRH header(s) MAY be placed in a 6LoWPAN packet.

If a non-RPL router receives a packet with an SRH-6LoRH header, there was a routing or a configuration error (see Section 8).

The desired reaction for the non-RPL router is to drop the packet, as opposed to skipping the header and forwarding the packet.

The Dispatch Value Bit Pattern for the SRH-6LoRH header indicates it is Critical. Routers that understand the 6LoRH general format detailed in Section 4 cannot ignore a 6LoRH header of this type and will drop the packet if it is unknown to them.

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-    -+-    -+ ... +-    -+
|1|0|0|  Size   |6LoRH Type 0..4| Hop1 | Hop2 |     | HopN |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-    -+-    -+ ... +-    -+

          Where N = Size + 1
```

Figure 6: The SRH-6LoRH

The 6LoRH Type of an SRH-6LoRH header indicates the compression level used for that header.

The fields following the 6LoRH Type are compressed addresses indicating the consecutive hops and are ordered from the first to the last hop.

All the addresses in a given SRH-6LoRH header MUST be compressed in an identical fashion, so the Length of the compressed form is the same for all.

In order to get different degrees of compression, multiple consecutive SRH-6LoRH headers MUST be used.

Type 0 means that the address is compressed down to one byte, whereas Type 4 means that the address is provided in full in the SRH-6LoRH with no compression. The complete list of Types of SRH-6LoRH and the corresponding compression level are provided in Figure 7:

| 6LoRH Type | Length of compressed IPv6 address (bytes) |
|---|---|
| 0 | 1 |
| 1 | 2 |
| 2 | 4 |
| 3 | 8 |
| 4 | 16 |

Figure 7: The SRH-6LoRH Types

In the case of an SRH-6LoRH header, the TSE field is used as a Size, which encodes the number of hops minus 1; so a Size of 0 means one hop, and the maximum that can be encoded is 32 hops. (If more than 32 hops need to be expressed, a sequence of SRH-6LoRH elements can be employed.) The result is that the Length, in bytes, of an SRH-6LoRH header is:

2 + Length_of_compressed_IPv6_address * (Size + 1)

## 5.2. SRH-6LoRH General Operation

### 5.2.1. Uncompressed SRH Operation

In the uncompressed form, when the root generates or forwards a packet in Non-Storing mode, it needs to include a Source Routing Header [RFC6554] to signal a strict source route path to a final destination down the DODAG.

All the hops along the path, except the first one, are encoded in order in the SRH. The last entry in the SRH is the final destination; the destination in the IPv6 header is the first hop along the source route path. The intermediate hops perform a swap and the Segments Left field indicates the active entry in the Routing Header [RFC2460].

The current destination of the packet, which is the termination of the current segment, is indicated at all times by the destination address of the IPv6 header.

### 5.2.2. 6LoRH-Compressed SRH Operation

The handling of the SRH-6LoRH is different: there is no swap, and a forwarding router that corresponds to the first entry in the first SRH-6LoRH, upon reception of a packet, effectively consumes that entry when forwarding. This means that the size of a compressed source-routed packet decreases as the packet progresses along its path and that the routing information is lost along the way. This also means that an SRH encoded with 6LoRH is not recoverable and cannot be protected.

When compressed with this specification, all the remaining hops MUST be encoded in order in one or more consecutive SRH-6LoRH headers. Whether or not there is an SRH-6LoRH header present, the address of the final destination is indicated in the LOWPAN_IPHC at all times along the path. Examples of this are provided in Appendix A.

The current destination (termination of the current segment) for a compressed source-routed packet is indicated in the first entry of the first SRH-6LoRH. In strict source routing, that entry MUST match an address of the router that receives the packet.

The last entry in the last SRH-6LoRH is the last router on the way to the final destination in the LLN. This router can be the final destination if it is found desirable to carry a whole IP-in-IP encapsulation all the way. Else, it is the RPL parent of the final destination, or a router acting at 6LoWPAN Router (6LR) [RFC6775] for the destination host, and it is advertising the host as an external route to RPL.

If the SRH-6LoRH header is contained in an IP-in-IP encapsulation, the last router removes the whole chain of headers. Otherwise, it removes the SRH-6LoRH header only.

### 5.2.3. Inner LOWPAN_IPHC Compression

6LoWPAN ND [RFC6282] is designed to support more than one IPv6 address per node and per Interface Identifier (IID); an IID is typically derived from a MAC address to optimize the LOWPAN_IPHC compression.

Link-local addresses are compressed with stateless address compression (S/DAC=0). The other addresses are derived from different prefixes, and they can be compressed with stateful address compression based on a context (S/DAC=1).

But, stateless compression is only defined for the specific link-local prefix as opposed to the prefix in an encapsulating header. And with stateful compression, the compression reference is found in a context, as opposed to an encapsulating header.

The result is that, in the case of an IP-in-IP encapsulation, it is possible to compress an inner source (respective destination) IP address in a LOWPAN_IPHC based on the encapsulating IP header only if stateful (context-based) compression is used. The compression will operate only if the IID in the source (respective destination) IP address in the outer and inner headers match, which usually means that they refer to the same node. This is encoded as S/DAC = 1 and S/AM=11. It must be noted that the outer destination address that is used to compress the inner destination address is the last entry in the last SRH-6LoRH header.

## 5.3. The Design Point of Popping Entries

In order to save energy and to optimize the chances of transmission success on lossy media, it is a design point for this specification that the entries in the SRH that have been used are removed from the packet. This creates a discrepancy from the art of IPv6, where Routing Headers are mutable but recoverable.

With this specification, the packet can be expanded at any hop into a valid IPv6 packet, including an SRH, and compressed back. But the packet, as decompressed along the way, will not carry all the consumed addresses that packet would have if it had been forwarded in the uncompressed form.

It is noted that:

> The value of keeping the whole RH in an IPv6 header is for the receiver to reverse it to use the symmetrical path on the way back.
>
> It is generally not a good idea to reverse a Routing Header. The RH may have been used to stay away from the shortest path for some reason that is only valid on the way in (segment routing).
>
> There is no use in reversing an RH in the present RPL specifications.

Point-to-Point (P2P) RPL reverses a path that was learned reactively as a part of the protocol operation, which is probably a cleaner way than a reversed echo on the data path.

Reversing a header is discouraged (by RFC 2460 [RFC2460]) for Redirected Header Option (RHO) unless it is authenticated, which requires an Authentication Header (AH). There is no definition of an AH operation for SRH, and there is no indication that the need exists in LLNs.

AH does not protect the RH on the way. AH is a validation at the receiver with the sole value of enabling the receiver to reverse it.

A RPL domain is usually protected by L2 security, which secures both RPL itself and the RH in the packets at every hop. This is a better security than that provided by AH.

In summary, the benefit of saving energy and lowering the chances of loss by sending smaller frames over the LLN are seen as overwhelming compared to the value of possibly reversing the header.

## 5.4. Compression Reference for SRH-6LoRH Header Entries

In order to optimize the compression of IP addresses present in the SRH headers, this specification requires that the 6LoWPAN layer identifies an address that is used as a reference for the compression.

With this specification, the Compression Reference for the first address found in an SRH header is the source of the IPv6 packet, and then the reference for each subsequent entry is the address of its predecessor once it is uncompressed.

With RPL [RFC6550], an SRH header may only be present in Non-Storing mode, and it may only be placed in the packet by the root of the DODAG, which must be the source of the resulting IPv6 packet [RFC2460]. In this case, the address used as Compression Reference is the address of the root.

The Compression Reference MUST be determined as follows:

The reference address may be obtained by configuration. The configuration may indicate either the address in full or the identifier of a 6LoWPAN Context that carries the address [RFC6775], for instance, one of the 16 Context Identifiers used in LOWPAN_IPHC [RFC6282].

Else, if there is no IP-in-IP encapsulation, the source address in the IPv6 header that is compressed with LOWPAN_IPHC is the reference for the compression.

Else, if the IP-in-IP compression specified in this document is used and the Encapsulator Address is provided, then the Encapsulator Address is the reference.

Else, meaning that the IP-in-IP compression specified in this document is used and the encapsulator is implicitly the root, the address of the root is the reference.

### 5.5. Popping Headers

Upon reception, the router checks whether the address in the first entry of the first SRH-6LoRH is one of its own addresses. If that is the case, the router MUST consume that entry before forwarding, which is an action of popping from a stack, where the stack is effectively the sequence of entries in consecutive SRH-6LoRH headers.

Popping an entry of an SRH-6LoRH header is a recursive action performed as follows:

If the Size of the current SRH-6LoRH header is 1 or more (indicating that there are at least 2 entries in the header), the router removes the first entry and decrements the Size by 1.

If the Size of the current SRH-6LoRH header is 0 (indicating that there is only 1 entry in the header) and there is no subsequent SRH-6LoRH after this, then the current SRH-6LoRH is removed.

If the Size of the current SRH-6LoRH header is 0 and there is a subsequent SRH-6LoRH and the Type of the subsequent SRH-6LoRH is equal to or greater than the Type of the current SRH-6LoRH header (indicating the same or lesser compression yielding the same or larger compressed forms), then the current SRH-6LoRH is removed.

If the Size of the current SRH-6LoRH header is 0 and there is a subsequent SRH-6LoRH and the Type of the subsequent SRH-6LoRH is less the Type of the current SRH-6LoRH header, the first entry of the subsequent SRH-6LoRH is removed and coalesced with the first entry of the current SRH-6LoRH.

At the end of the process, if there are no more SRH-6LoRH in the packet, then the processing node is the last router along the source route path.

An example of this operation is provided in Appendix A.3.

### 5.6. Forwarding

When receiving a packet with an SRH-6LoRH, a router determines the IPv6 address of the current segment endpoint.

If strict source routing is enforced and this router is not the segment endpoint for the packet, then this router MUST drop the packet.

If this router is the current segment endpoint, then the router pops its address as described in Section 5.5 and continues processing the packet.

If there is still an SRH-6LoRH, then the router determines the new segment endpoint and routes the packet towards that endpoint.

Otherwise, the router uses the destination in the inner IP header to forward or accept the packet.

The segment endpoint of a packet MUST be determined as follows:

The router first determines the Compression Reference as discussed in Section 4.3.1.

The router then coalesces the Compression Reference with the first entry of the first SRH-6LoRH header as discussed in Section 5.4. If the SRH-6LoRH header is Type 4, then the coalescence is a full override.

Since the Compression Reference is an uncompressed address, the coalesced IPv6 address is also expressed in the full 128 bits.

## 6. The RPL Packet Information 6LoRH (RPI-6LoRH)

Section 11.2 of the RPL document [RFC6550] specifies the RPL Packet Information (RPI) as a set of fields that are placed by RPL routers in IP packets to identify the RPL Instance, detect anomalies, and trigger corrective actions.

In particular, the SenderRank, which is the scalar metric computed by a specialized Objective Function such as described in RFC 6552 [RFC6552], indicates the Rank of the sender and is modified at each hop. The SenderRank field is used to validate that the packet progresses in the expected direction, either upwards or downwards, along the DODAG.

RPL defines the "The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams" [RFC6553] to transport the RPI, which is carried in an IPv6 Hop-by-Hop Options Header [RFC2460], typically consuming 8 bytes per packet.

With RFC 6553 [RFC6553], the RPL Option is encoded as 6 octets, which must be placed in a Hop-by-Hop header that consumes two additional octets for a total of 8 octets. To limit the header's range to just the RPL domain, the Hop-by-Hop header must be added to (or removed from) packets that cross the border of the RPL domain.

The 8-byte overhead is detrimental to LLN operation, particularly with regard to bandwidth and battery constraints. These bytes may cause a containing frame to grow above maximum frame size, leading to Layer 2 or 6LoWPAN [RFC4944] fragmentation, which in turn leads to even more energy expenditure and issues discussed in "LLN Fragment Forwarding and Recovery" [FORWARD-FRAG].

An additional overhead comes from the need, in certain cases, to add an IP-in-IP encapsulation to carry the Hop-by-Hop header. This is needed when the router that inserts the Hop-by-Hop header is not the source of the packet so that an error can be returned to the router. This is also the case when a packet originated by a RPL node must be stripped from the Hop-by-Hop header to be routed outside the RPL domain.

For that reason, this specification defines an IP-in-IP-6LoRH header in Section 7, but it must be noted that removal of a 6LoRH header does not require manipulation of the packet in the LOWPAN_IPHC, and thus, if the source address in the LOWPAN_IPHC is the node that inserted the IP-in-IP-6LoRH header, then this situation alone does not mandate an IP-in-IP-6LoRH header.

Note: It was found that some implementations omit the RPI for packets going down the RPL graph in Non-Storing mode, even though RPL indicates that the RPI should be placed in the packet. With this specification, the RPI is important to indicate the RPLInstanceID, so the RPI should not be omitted.

As a result, a RPL packet may bear only an RPI-6LoRH header and no IP-in-IP-6LoRH header. In that case, the source and destination of the packet are specified by the LOWPAN_IPHC.

As with RFC 6553 [RFC6553], the fields in the RPI include an 'O', an 'R', and an 'F' bit, an 8-bit RPLInstanceID (with some internal structure), and a 16-bit SenderRank.

The remainder of this section defines the RPI-6LoRH header, which is a Critical 6LoWPAN Routing Header that is designed to transport the RPI in 6LoWPAN LLNs.

## 6.1. Compressing the RPLInstanceID

RPL Instances are discussed in Section 5 of the RPL specification [RFC6550]. A number of simple use cases do not require more than one RPL Instance, and in such cases, the RPL Instance is expected to be the Global Instance 0. A global RPLInstanceID is encoded in a RPLInstanceID field as follows:

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|0|     ID      |  Global RPLInstanceID in 0..127
+-+-+-+-+-+-+-+-+
```

Figure 8: RPLInstanceID Field Format for Global Instances

For the particular case of the Global Instance 0, the RPLInstanceID field is all zeros. This specification allows the compressor to elide a RPLInstanceID field that is all zeros and defines an I flag that, when set, signals that the field is elided.

## 6.2. Compressing the SenderRank

The SenderRank is the result of the DAGRank operation on the Rank of the sender; here, the DAGRank operation is defined in Section 3.5.1 of the RPL specification [RFC6550] as:

```
DAGRank(rank) = floor(rank/MinHopRankIncrease)
```

If MinHopRankIncrease is set to a multiple of 256, the least significant eight bits of the SenderRank will be all zeroes; by eliding those, the SenderRank can be compressed into a single byte. This idea is used in RFC 6550 [RFC6550], by defining DEFAULT_MIN_HOP_RANK_INCREASE as 256, and in RFC 6552 [RFC6552], which defaults MinHopRankIncrease to DEFAULT_MIN_HOP_RANK_INCREASE.

This specification allows for the SenderRank to be encoded as either 1 or 2 bytes and defines a K flag that, when set, signals that a single byte is used.

## 6.3. The Overall RPI-6LoRH Encoding

The RPI-6LoRH header provides a compressed form for the RPL RPI. Routers that need to forward a packet with a RPI-6LoRH header are expected to be RPL routers that support this specification.

If a non-RPL router receives a packet with a RPI-6LoRH header, there was a routing or a configuration error (see Section 8).

The desired reaction for the non-RPL router is to drop the packet as opposed to skip the header and forward the packet, which could end up forming loops by reinjecting the packet in the wrong RPL Instance.

The Dispatch Value Bit Pattern for the SRH-6LoRH header indicates it is Critical. Routers that understand the 6LoRH general format detailed in Section 4 cannot ignore a 6LoRH header of this type and will drop the packet if it is unknown to them.

Since the RPI-6LoRH header is a Critical header, the TSE field does not need to be a length expressed in bytes. Here, the field is fully reused for control bits that encode the O, R, and F flags from the RPI, as well as the I and K flags that indicate the compression format.

The RPI-6LoRH is Type 5.

The RPI-6LoRH header is immediately followed by the RPLInstanceID field, unless that field is fully elided, and then the SenderRank, which is either compressed into one byte or fully in-lined as 2 bytes. The I and K flags in the RPI-6LoRH header indicate whether the RPLInstanceID is elided and/or the SenderRank is compressed. Depending on these bits, the Length of the RPI-6LoRH may vary as described hereafter.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  ...  -+-+-+
|1|0|0|O|R|F|I|K| 6LoRH Type=5  |   Compressed fields  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  ...  -+-+-+
```

Figure 9: The Generic RPI-6LoRH Format

O, R, and F bits: The O, R, and F bits are defined in Section 11.2 of RFC 6550 [RFC6550].

I flag: If it is set, the RPLInstanceID is elided and the RPLInstanceID is the Global RPLInstanceID 0. If it is not set, the octet immediately following the Type field contains the RPLInstanceID as specified in Section 5.1 of RFC 6550 [RFC6550].

K flag: If it is set, the SenderRank is compressed into 1 octet, with the least significant octet elided. If it is not set, the SenderRank is fully inlined as 2 octets.

In Figure 10, the RPLInstanceID is the Global RPLInstanceID 0, and the MinHopRankIncrease is a multiple of 256, so the least significant byte is all zeros and can be elided:

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1|0|0|O|R|F|1|1| 6LoRH Type=5  | SenderRank    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          I=1, K=1
```

Figure 10: The Most Compressed RPI-6LoRH

In Figure 11, the RPLInstanceID is the Global RPLInstanceID 0, but both bytes of the SenderRank are significant so it cannot be compressed:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1|0|0|O|R|F|1|0| 6LoRH Type=5  |        SenderRank             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          I=1, K=0
```

Figure 11: Eliding the RPLInstanceID

In Figure 12, the RPLInstanceID is not the Global RPLInstanceID 0, and the MinHopRankIncrease is a multiple of 256:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1|0|0|O|R|F|0|1| 6LoRH Type=5  | RPLInstanceID |  SenderRank   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          I=0, K=1
```

Figure 12: Compressing SenderRank

In Figure 13, the RPLInstanceID is not the Global RPLInstanceID 0, and both bytes of the SenderRank are significant:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |1|0|0|O|R|F|0|0| 6LoRH Type=5  | RPLInstanceID |    Sender-...
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     ...-Rank      |
   +-+-+-+-+-+-+-+-+
            I=0, K=0
```

Figure 13: The Least Compressed Form of RPI-6LoRH

## 7. The IP-in-IP 6LoRH Header

The IP-in-IP 6LoRH (IP-in-IP-6LoRH) header is an Elective 6LoWPAN Routing Header that provides a compressed form for the encapsulating IPv6 Header in the case of an IP-in-IP encapsulation.

An IP-in-IP encapsulation is used to insert a field such as a Routing Header or an RPI at a router that is not the source of the packet. In order to send an error back regarding the inserted field, the address of the router that performs the insertion must be provided.

The encapsulation can also enable the last router prior to the Destination to remove a field such as the RPI, but this can be done in the compressed form by removing the RPI-6LoRH, so an IP-in-IP-6LoRH encapsulation is not required for that sole purpose.

The Dispatch Value Bit Pattern for the SRH-6LoRH header indicates it is Elective. This field is not Critical for routing since it does not indicate the destination of the packet, which is either encoded in an SRH-6LoRH header or in the inner IP header. A 6LoRH header of this type can be skipped if not understood (per Section 4), and the 6LoRH header indicates the Length in bytes.

```
   0                   1                   2
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-       ...      -+
  |1|0|1| Length  | 6LoRH Type 6  |  Hop Limit    | Encaps. Address |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-       ...      -+
```

Figure 14: The IP-in-IP-6LoRH

The Length of an IP-in-IP-6LoRH header is expressed in bytes and MUST be at least 1, to indicate a Hop Limit (HL) that is decremented at each hop. When the HL reaches 0, the packet is dropped per RFC 2460 [RFC2460].

If the Length of an IP-in-IP-6LoRH header is exactly 1, then the Encapsulator Address is elided, which means that the encapsulator is a well-known router, for instance, the root in a RPL graph.

The most efficient compression of an IP-in-IP encapsulation that can be achieved with this specification is obtained when an endpoint of the packet is the root of the RPL DODAG associated to the RPL Instance that is used to forward the packet, and the root address is known implicitly as opposed to signaled explicitly in the data packets.

If the Length of an IP-in-IP-6LoRH header is greater than 1, then an Encapsulator Address is placed in a compressed form after the Hop Limit field. The value of the Length indicates which compression is performed on the Encapsulator Address. For instance, a Length of 3 indicates that the Encapsulator Address is compressed to 2 bytes. The reference for the compression is the address of the root of the DODAG. The way the address of the root is determined is discussed in Section 4.3.2.

With RPL, the destination address in the IP-in-IP header is implicitly the root in the RPL graph for packets going upwards; in Storing mode, it is the destination address in the LOWPAN_IPHC for packets going downwards. In Non-Storing mode, there is no implicit value for packets going downwards.

If the implicit value is correct, the destination IP address of the IP-in-IP encapsulation can be elided. Else, the destination IP address of the IP-in-IP header is transported in an SRH-6LoRH header as the first entry of the first of these headers.

If the final destination of the packet is a leaf that does not support this specification, then the chain of 6LoRH headers must be stripped by the RPL/6LR router to which the leaf is attached. In that example, the destination IP address of the IP-in-IP header cannot be elided.

In the special case where a 6LoRH header is used to route 6LoWPAN fragments, the destination address is not accessible in the LOWPAN_IPHC on all fragments and can be elided only for the first fragment and for packets going upwards.

## 8. Management Considerations

Though it is possible to decompress a packet at any hop, this specification is optimized to enable that a packet is forwarded in its compressed form all the way, and it makes sense to deploy homogeneous networks where all nodes, or no nodes at all, use the compression technique detailed therein.

This specification aims at a simple implementation running in constrained nodes, so it does indeed expect a homogeneous network and, as a consequence, it does not provide a method to determine the level of support by the next hops at forwarding time.

Should an extension to this specification provide such a method, forwarding nodes could compress or decompress the RPL artifacts appropriately and enable a backward compatibility between nodes that support this specification and nodes that do not.

It results that this specification does not attempt to enable such backwards compatibility. It does not require extraneous code to exchange and handle error messages to automatically correct mismatch situations either.

When a packet is expected to carry a 6LoRH header but does not, the node that discovers the issue is expected to send an ICMPv6 error message to the root. It should be sent at an adapted rate-limitation and with a type 4 (indicating a "Parameter Problem") and a code 0 (indicating an "Unrecognized Next Header field encountered"). The relevant portion of the received packet should be embedded and the offset therein where the 6LoRH header was expected should be pointed out.

When a packet is received with a 6LoRH header that is not recognized, the node that discovers the issue is expected to send an ICMPv6 error message to the root. It should be sent at an adapted rate-limitation and with a type 4 (indicating a "Parameter Problem") and a code 1 (indicating an "Unrecognized Next Header type encountered"). The relevant portion of the received packet should be embedded and the offset therein where the 6LoRH header was expected should be pointed out.

In both cases, the node SHOULD NOT place a 6LoRH header as defined in this specification in the resulting message, and the node should either omit the RPI or place it uncompressed after the IPv6 header.

Additionally, in both cases, an alternate management method may be preferred in order to notify the network administrator that there is a configuration error.

Keeping the network homogeneous is either a deployment issue, by deploying only devices with a same capability, or a management issue, by configuring all devices to either use or not use a certain level of this compression technique and its future additions.

In particular, the situation where a node receives a message with a Critical 6LoWPAN Routing Header that it does not understand is an administrative error whereby the wrong device is placed in a network, or the device is misconfigured.

When a mismatch situation is detected, it is expected that the device raises some management alert indicating the issue, e.g., that it has to drop a packet with a Critical 6LoRH.

# 9. Security Considerations

The security considerations of RFC 4944 [RFC4944], RFC 6282 [RFC6282], and RFC 6553 [RFC6553] apply.

Using a compressed format as opposed to the full in-line format is logically equivalent and is believed not to create an opening for a new threat when compared to RFC 6550 [RFC6550], RFC 6553 [RFC6553], and RFC 6554 [RFC6554], noting that, even though intermediate hops are removed from the SRH header as they are consumed, a node may still identify that the rest of the source-routed path includes a loop or not (see the "Security" section of RFC 6554). It must be noted that if the attacker is not part of the loop, then there is always a node at the beginning of the loop that can detect it and remove it.

# 10. IANA Considerations

## 10.1. Reserving Space in 6LoWPAN Dispatch Page 1

This specification reserves Dispatch Value Bit Patterns within the 6LoWPAN Dispatch Page 1 as follows:

10 1xxxxx: for Elective 6LoWPAN Routing Headers

10 0xxxxx: for Critical 6LoWPAN Routing Headers

Additionally, this document creates two IANA registries: one for the Critical 6LoWPAN Routing Header Type and one for the Elective 6LoWPAN Routing Header Type, each with 256 possible values, from 0 to 255, as described below.

Future assignments are made by IANA using the "RFC Required" procedure [RFC5226].

### 10.2. New Critical 6LoWPAN Routing Header Type Registry

This document creates an IANA registry titled "Critical 6LoWPAN Routing Header Type" and assigns the following values:

0-4: SRH-6LoRH [RFC8138]

5: RPI-6LoRH [RFC8138]

### 10.3. New Elective 6LoWPAN Routing Header Type Registry

This document creates an IANA registry titled "Elective 6LoWPAN Routing Header Type" and assigns the following value:

6: IP-in-IP-6LoRH [RFC8138]

## 11. References

### 11.1. Normative References

[IEEE.802.15.4]
IEEE, "IEEE Standard for Low-Rate Wireless Networks", IEEE 802.15.4-2015, DOI 10.1109/IEEESTD.2016.7460875, <http://ieeexplore.ieee.org/document/7460875/>.

[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, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[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>.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <http://www.rfc-editor.org/info/rfc4944>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <http://www.rfc-editor.org/info/rfc6282>.

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

[RFC6552] Thubert, P., Ed., "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6552, DOI 10.17487/RFC6552, March 2012, <http://www.rfc-editor.org/info/rfc6552>.

[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>.

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

## 11.2. Informative References

[FORWARD-FRAG] Thubert, P., Ed. and J. Hui, "LLN Fragment Forwarding and Recovery", Work in Progress, draft-thubert-6lo-forwarding-fragments-05, April 2017.

[IPv6-ARCH]
Thubert, P., Ed., "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4", Work in Progress, draft-ietf-6tisch-architecture-11, January 2017.

[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>.

[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>.

[RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, DOI 10.17487/RFC7228, May 2014, <http://www.rfc-editor.org/info/rfc7228>.

[RFC7554] Watteyne, T., Ed., Palattella, M., and L. Grieco, "Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement", RFC 7554, DOI 10.17487/RFC7554, May 2015, <http://www.rfc-editor.org/info/rfc7554>.

[RPL-INFO] Robles, M., Richardson, M., and P. Thubert, "When to use RFC 6553, 6554 and IPv6-in-IPv6", Work in Progress, draft-ietf-roll-useofrplinfo-14, April 2017.

## Appendix A.  Examples

### A.1.  Examples Compressing the RPI

The example in Figure 15 illustrates the 6LoRH compression of a classical packet in Storing mode in all directions, as well as in Non-Storing mode for a packet going up the DODAG following the default route to the root.  In this particular example, a fragmentation process takes place per RFC 4944 [RFC4944], and the fragment headers must be placed in Page 0 before switching to Page 1:

```
+-  ...  -+-  ...  -+-+ ... -+- ... +-+-+ ... -+-+-+-+-+-+-+-+-+-+...
|Frag type|Frag hdr |11110001|  RPI-   |IP-in-IP| LOWPAN_IPHC | ...
|RFC 4944 |RFC 4944 | Page 1 | 6LoRH   | 6LoRH  |             |
+-  ...  -+-  ...  -+-+ ... -+- ... +-+-+ ... -+-+-+-+-+-+-+-+-+-+...
                                                 <-  RFC 6282  ->
                                                  No RPL artifact

+-  ...  -+-  ...  -+-+ ... -+-+ ... -+- ... +-+-+-+-+-+-+-+-+-+-+...
|Frag type|Frag hdr |
|RFC 4944 |RFC 4944 |  Payload (cont)
+-  ...  -+-  ...  -+-+ ... -+-+ ... -+- ... +-+-+-+-+-+-+-+-+-+-+...

+-  ...  -+-  ...  -+-+ ... -+-+ ... -+- ... +-+-+-+-+-+-+-+-+-+-+...
|Frag type|Frag hdr |
|RFC 4944 |RFC 4944 |  Payload (cont)
+-  ...  -+-  ...  -+-+ ... -+-+ ... -+- ... +-+-+-+-+-+-+-+-+-+-+...
```

Figure 15: Example Compressed Packet with RPI

In Storing mode, if the packet stays within the RPL domain, then it is possible to save the IP-in-IP encapsulation, in which case, only the RPI is compressed with a 6LoRH, as illustrated in Figure 16 in the case of a non-fragmented ICMP packet:

```
+- ...  -+-+- ... -+-+-+-+ ... -+-+-+-+ ... -+-+-+-+-+-+-+-+-+-+-+...
|11110001| RPI-6LoRH |  NH = 0      | NH = 58  |  ICMP message ...
|Page 1  |  Type 5   | 6LOWPAN_IPHC | (ICMP)   |  (no compression)
+- ...  -+-+- ... -+-+-+-+ ... -+-+-+-+ ... -+-+-+-+-+-+-+-+-+-+-+...
                      <-      RFC 6282       ->
                          No RPL artifact
```

Figure 16: Example ICMP Packet with RPI in Storing Mode

The format in Figure 16 is logically equivalent to the uncompressed format illustrated in Figure 17:

```
+-+-+-+- ... -+-+-+-+ ... -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+...
|  IPv6 Header   |Hop-by-Hop |  RPI in         |  ICMP message ...
|  NH = 58       |Header     |  RPL Option     |
+-+-+-+- ... -+-+-+-+ ... -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+...
```

Figure 17: Uncompressed ICMP Packet with RPI

For a UDP packet, the transport header can be compressed with 6LoWPAN HC [RFC6282] as illustrated in Figure 18:

```
+-+ ... -+-+-...-+-+- ... -+-+-+-+ ... -+-+-+ ... -+-+-+-+-+...
|11110001| RPI-  | NH=1          |11110CPP| Compressed | UDP
|Page 1  | 6LoRH | LOWPAN_IPHC   | UDP    | UDP header | Payload
+-+ ... -+-+-...-+-+- ... -+-+-+-+ ... -+-+-+ ... -+-+-+-+-+...
                  <-        RFC 6282              ->
                            No RPL artifact
```

Figure 18: Uncompressed ICMP Packet with RPI

If the packet is received from the Internet in Storing mode, then the root is supposed to encapsulate the packet to insert the RPI.  The resulting format would be as represented in Figure 19:

```
+-+ ... -+-+-...-+-+-- ... -+-+-+-+- ... -+-+ ... -+-+-+ ... -+-+-+...
|11110001| RPI-  | IP-in-IP | NH=1          |11110CPP| Compressed | UDP
|Page 1  | 6LoRH |  6LoRH   | LOWPAN_IPHC   | UDP    | UDP header | Payld
+-+ ... -+-+-...-+-+-- ... -+-+-+-+- ... -+-+ ... -+-+-+ ... -+-+-+...
                             <-        RFC 6282              ->
                                       No RPL artifact
```

Figure 19: RPI Inserted by the Root in Storing Mode

## A.2. Example of a Downward Packet in Non-Storing Mode

The example illustrated in Figure 20 is a classical packet in Non-Storing mode for a packet going down the DODAG following a source-routed path from the root.  Say that we have four forwarding hops to reach a destination.  In the uncompressed form, when the root generates the packet, the last 3 hops are encoded in a Routing Header Type 3 (SRH) and the first hop is the destination of the packet.  The intermediate hops perform a swap; the hop count indicates the current active hop as defined in RFC 2460 [RFC2460] and RFC 6554 [RFC6554].

When compressed with this specification, the 4 hops are encoded in SRH-6LoRH when the root generates the packet, and the final destination is left in the LOWPAN_IPHC. There is no swap; the forwarding node that corresponds to the first entry effectively consumes it when forwarding, which means that the size of the encoded packet decreases and that the hop information is lost.

If the last hop in an SRH-6LoRH is not the final destination, then it removes the SRH-6LoRH before forwarding.

In the particular example illustrated in Figure 20, all addresses in the DODAG are assigned from the same /112 prefix and the last 2 octets encoding an identifier such as an IEEE 802.15.4 short address. In that case, all addresses can be compressed to 2 octets, using the root address as reference. There will be one SRH_6LoRH header with, in this example, three compressed addresses:

```
+-+ ... -+-+ ... +-+- ... -+-+- ... +-+-+-+ ... +-+-+ ... -+ ... +-...
|11110001|SRH-6LoRH| RPI-  | IP-in-IP | NH=1        |11110CPP| UDP | UDP
|Page 1  |Type1 S=2| 6LoRH |  6LoRH   |LOWPAN_IPHC| UDP     | hdr |Payld
+-+ ... -+-+ ... +-+- ... -+-+-- ... -+-+-+ ... +-+-+ ... -+ ... +-...
           <-8bytes->                   <-        RFC 6282      ->
                                                No RPL artifact
```

Figure 20: Example Compressed Packet with SRH

One may note that the RPI is provided. This is because the address of the root that is the source of the IP-in-IP header is elided and inferred from the RPLInstanceID in the RPI. Once found from a local context, that address is used as a Compression Reference to expand addresses in the SRH-6LoRH.

With the RPL specifications available at the time of writing, the root is the only node that may incorporate an SRH in an IP packet. When the root forwards a packet that it did not generate, it has to encapsulate the packet with IP-in-IP.

But, if the root generates the packet towards a node in its DODAG, then it should avoid the extra IP-in-IP as illustrated in Figure 21:

```
+- ...  -+-+-+ ... +-+-+-+ ... -+-+-+-+-+-+-+-++-+- ... -+-+-+-+-+...
|11110001| SRH-6LoRH | NH=1        | 11110CPP | Compressed | UDP
|Page 1  | Type1 S=3 | LOWPAN_IPHC| LOWPAN-NHC| UDP header | Payload
+- ...  -+-+-+ ... +-+-+-+ ... -+-+-+-+-+-+-+-++-+- ... -+-+-+-+-+...
                                <-        RFC 6282        ->
```

Figure 21: Compressed SRH 4*2bytes Entries Sourced by Root

Note: The RPI is not represented, though RPL [RFC6550] generally expects it. In this particular case, since the Compression Reference for the SRH-6LoRH is the source address in the LOWPAN_IPHC, and the routing is strict along the source route path, the RPI does not appear to be absolutely necessary.

In Figure 21, all the nodes along the source route path share the same /112 prefix. This is typical of IPv6 addresses derived from an IEEE802.15.4 short address, as long as all the nodes share the same PAN-ID. In that case, a Type 1 SRH-6LoRH header can be used for encoding. The IPv6 address of the root is taken as reference, and only the last 2 octets of the address of the intermediate hops are encoded. The Size of 3 indicates 4 hops, resulting in an SRH-6LoRH of 10 bytes.

## A.3. Example of SRH-6LoRH Life Cycle

This section illustrates the operation specified in Section 5.6 of forwarding a packet with a compressed SRH along an A->B->C->D source route path. The operation of popping addresses is exemplified at each hop.

```
Packet as received by node A
----------------------------
  Type 3 SRH-6LoRH Size = 0   AAAA AAAA AAAA AAAA
  Type 1 SRH-6LoRH Size = 0                  BBBB
  Type 2 SRH-6LoRH Size = 1             CCCC CCCC
                                        DDDD DDDD

 Step 1: Popping BBBB, the first entry of the next SRH-6LoRH
 Step 2: If larger value (2 vs. 1), the SRH-6LoRH is removed

  Type 3 SRH-6LoRH Size = 0   AAAA AAAA AAAA AAAA
  Type 2 SRH-6LoRH Size = 1             CCCC CCCC
                                        DDDD DDDD

 Step 3: Recursion ended; coalescing BBBB with the first entry
  Type 3 SRH-6LoRH Size = 0   AAAA AAAA AAAA BBBB

 Step 4: Routing based on next segment endpoint to B
```

Figure 22: Processing at Node A

```
   Packet as received by node B
   ----------------------------
     Type 3 SRH-6LoRH Size = 0   AAAA AAAA AAAA BBBB
     Type 2 SRH-6LoRH Size = 1             CCCC CCCC
                                           DDDD DDDD

    Step 1: Popping CCCC CCCC, the first entry of the next SRH-6LoRH
    Step 2: Removing the first entry and decrementing the Size (by 1)

     Type 3 SRH-6LoRH Size = 0   AAAA AAAA AAAA BBBB
     Type 2 SRH-6LoRH Size = 0             DDDD DDDD

    Step 3: Recursion ended; coalescing CCCC CCCC with the first entry
     Type 3 SRH-6LoRH Size = 0   AAAA AAAA CCCC CCCC

    Step 4: Routing based on next segment endpoint to C
```

Figure 23: Processing at Node B

```
   Packet as received by node C
   ----------------------------

     Type 3 SRH-6LoRH Size = 0   AAAA AAAA CCCC CCCC
     Type 2 SRH-6LoRH Size = 0             DDDD DDDD

    Step 1: Popping DDDD DDDD, the first entry of the next SRH-6LoRH
    Step 2: The SRH-6LoRH is removed

     Type 3 SRH-6LoRH Size = 0   AAAA AAAA CCCC CCCC

    Step 3: Recursion ended; coalescing DDDD DDDDD with the first entry
     Type 3 SRH-6LoRH Size = 0   AAAA AAAA DDDD DDDD

    Step 4: Routing based on next segment endpoint to D
```

Figure 24: Processing at Node C

```
   Packet as received by node D
   ----------------------------
     Type 3 SRH-6LoRH Size = 0   AAAA AAAA DDDD DDDD

    Step 1: The SRH-6LoRH is removed
    Step 2: No more header; routing based on inner IP header
```

Figure 25: Processing at Node D

## Acknowledgements

The authors wish to thank Tom Phinney, Thomas Watteyne, Tengfei Chang, Martin Turon, James Woodyatt, Samita Chakrabarti, Jonathan Hui, Gabriel Montenegro, and Ralph Droms for constructive reviews to the design in the 6lo working group. The overall discussion involved participants to the 6MAN, 6TiSCH, and ROLL WGs; thank you all. Special thanks to Michael Richardson and Ines Robles (the Chairs of the ROLL WG), Brian Haberman (the Internet Area AD), and Alvaro Retana and Adrian Farrel (Routing Area ADs) for driving this complex effort across working groups and areas.

## Authors' Addresses

Pascal Thubert (editor)
Cisco Systems
Building D - Regus
45 Allee des Ormes
BP1200
MOUGINS - Sophia Antipolis 06254
France

Phone: +33 4 97 23 26 34
Email: pthubert@cisco.com

Carsten Bormann
Universitaet Bremen TZI
Postfach 330440
Bremen D-28359
Germany

Phone: +49-421-218-63921
Email: cabo@tzi.org

Laurent Toutain
IMT Atlantique
2 rue de la Chataigneraie
CS 17607
Cesson-Sevigne Cedex 35576
France

Email: Laurent.Toutain@IMT-Atlantique.fr

Robert Cragie
ARM Ltd.
110 Fulbourn Road
Cambridge CB1 9NJ
United Kingdom

Email: robert.cragie@arm.com