Network Working Group
Internet-Draft
Intended status: Experimental
Expires: May 26, 2017

M. Boucadair
C. Jacquenet
Orange

November 22, 2016

A Compact LISP Encapsulation Scheme to Transport IPv4 Packets over an IPv6 Network
draft-boucadair-lisp-v6-compact-header-03

Abstract

The encapsulation scheme used by the Locator/ID Separation Protocol (LISP) may sometimes raise MTU issues at the cost of possibly degrading the overall performance of the LISP network, especially in IPv6 migration contexts. This document proposes a new, more compact, encapsulation scheme that aims to accommodate such issues and facilitate LISP deployment for IPv6 migration purposes, in particular. This compact header may be considered to provide IPv4 over IPv6 connectivity for mobile LISP-capable devices.

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].

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 May 26, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
2. A Compact LISP Header 6
2.1. C Flag 6
2.2. On the Use of IPv4-Embedded IPv6 RLOCs 7
2.3. Truncated TCP Header 8
2.4. Compact LISP Header Format 9
3. LISP Encapsulation with the Compact Header Form 11
3.1. UDP Packets 11
3.2. TCP Packets 12
3.3. Fragments 12
4. LISP Decapsulation with the Compact LISP Header 13
4.1. Build an IPv4/UDP Header 13
4.2. Build an IPv4/TCP Header 13
5. A More Compact LISP Encapsulation Flavor 14
5.1. LISP Encapsulation with the More Compact Header Form . . 16
5.1.1. UDP Packets 16
5.1.2. TCP Packets 17
5.1.3. Fragments 17
5.2. LISP Decapsulation with the More Compact LISP Header . . 17
5.2.1. Build an IPv4/UDP Header 18
5.2.2. Build an IPv4/TCP Header 18
6. Discussion 19
7. Security Considerations 19
8. IANA Considerations 19
9. Acknowledgments 20
10. Normative references 20
Authors' Addresses 20

1. Introduction

The base specification of the Locator/ID Separation Protocol (LISP, [RFC6830]) defines an encapsulation scheme for transporting packets between xTR routers. When applied at the scale of the Internet, this encapsulation scheme may raise MTU issues because of the LISP overhead. This overhead may be aggravated when IPv6 transfer capabilities are used to interconnect LISP sites.

As a reminder, Figure 1 shows the format of an encapsulated TCP ([RFC0793]) packet over IPv6, while Figure 2 covers UDP ([RFC0768]).

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  / |Version| Traffic Class |           Flow Label                  |
 /  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   |         Payload Length        | Next Header=17|   Hop Limit   |
v   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
O   +                                                               +
u   |                                                               |
t   +                     Source Routing Locator                    +
e   |                                                               |
r   +                                                               +
    |                                                               |
H   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
d   |                                                               |
r   +                                                               +
    |                                                               |
^   +                  Destination Routing Locator                  +
|   |                                                               |
 \  +                                                               +
  \ |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  / |       Source Port = xxxx      |       Dest Port = 4341        |
UDP +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  \ |           UDP Length          |        UDP Checksum           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
L   |N|L|E|V|I|flags|            Nonce/Map-Version                  |
I \ +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
S / |                 Instance ID/Locator-Status-Bits               |
P   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  / |Version|  IHL  |Type of Service|          Total Length         |
 /  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   |         Identification        |Flags|      Fragment Offset    |
|   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
IH  |  Time to Live |    Protocol   |         Header Checksum       |
```

```
   |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   |                           Source EID                          |
    \  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     \ |                         Destination EID                       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |          Source Port          |       Destination Port        |
    /  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   |                        Sequence Number                        |
   |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   |                    Acknowledgment Number                      |
   |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  TCP  |  Data |           |U|A|P|R|S|F|                               |
   |   | Offset| Reserved  |R|C|S|S|Y|I|            Window             |
   |   |       |           |G|K|H|T|N|N|                               |
   |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   |           Checksum            |         Urgent Pointer        |
   |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    \  |                      (Optional) Options                       |
     \ +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: LISP IPv4-in-IPv6 Header Format (TCP)

```
        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
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |Version| Traffic Class |           Flow Label                  |
    /  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   |         Payload Length        | Next Header=17|   Hop Limit   |
   v   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                               |
   O   +                                                               +
   u   |                                                               |
   t   +                     Source Routing Locator                    +
   e   |                                                               |
   r   +                                                               +
       |                                                               |
   H   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   d   |                                                               |
   r   +                                                               +
       |                                                               |
   ^   +                  Destination Routing Locator                  +
   |   |                                                               |
    \  +                                                               +
     \ |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |       Source Port = xxxx      |       Dest Port = 4341        |
   UDP +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     \ |           UDP Length          |        UDP Checksum           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   L   |N|L|E|V|I|flags|            Nonce/Map-Version                  |
   I \ +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   S / |                  Instance ID/Locator-Status-Bits               |
   P   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |Version|  IHL  |Type of Service|          Total Length         |
    /  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   |         Identification        |Flags|      Fragment Offset    |
   |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   IH  |  Time to Live |    Protocol   |         Header Checksum       |
   |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   |                            Source EID                         |
    \  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     \ |                          Destination EID                      |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |           Source Port         |       Destination Port        |
   UDP +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     \ |           UDP Length          |        UDP Checksum           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: LISP IPv4-in-IPv6 Header Format (UDP)

This document proposes a new LISP encapsulation scheme that aims to reduce the overhead induced by LISP encapsulation (i.e., the one defined in [RFC6830]) to transport IPv4 packets over an IPv6 infrastructure.

This proposal does not suggest to obsolete the current LISP base encapsulation mode as defined in [RFC6830]. Rather, this document proposes to associate a meaning with one of the reserved flag bits (see Section 5.3 of [RFC6830]) to explicitly indicate that, when the bit is set, compact LISP encapsulation is in use. This bit is called the C-bit ("Compact" flag bit). Defining this capability in the base LISP header will help experimenting this compact header and assess its efficiency.

This document does not introduce an overhead compared to the encapsulation scheme in [RFC6830] given that the solution relies on a compact encoding. Some examples to illustrate the compression ratio achieved with the proposed compact header are shown below.

	Origin Size	RFC6830 IPv4-in-IPv6	Compact Header 1	Compact Header 2
TCP ACK	40 bytes	96 bytes	68 bytes Gain: 29%	64 bytes Gain: 33%
RTP	60 bytes	116 bytes	80 bytes Gain: 31%	76 bytes Gain: 34%

This document assumes that RLOCs can be encoded as prefixes. One of the bits of "Unused Flags" in a Map-Register and Map-Reply can be used to explicitly indicate the enclosed locator is an IPv6 prefix. The length of the prefix can be 32, 40, 48, 56, 64, or 96 [RFC6052]. The RLOC address will be built using the algorithm in [RFC6052].

2. A Compact LISP Header

2.1. C Flag

In order to allow for the co-existence of both legacy LISP header and this compact new header, this document associates a meaning with one of the reserved flag bits in [RFC6830]. Concretely, Figure 3 shows the required change to the LISP header to support the new compact LISP header.

```
OLD:
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   L   |N|L|E|V|I|flags|            Nonce/Map-Version                  |
   I \ +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   S / |                 Instance ID/Locator-Status-Bits               |
   P   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

NEW:
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   L   |N|L|E|V|I|flg|C|            Nonce/Map-Version                  |
   I \ +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   S / |                 Instance ID/Locator-Status-Bits               |
   P   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: C-bit in the LISP Header

The "flg" bits are reserved bits for future assignment as additional flag bits. These additional flag bits MUST each be set to zero and MUST be ignored upon receipt.

The description of the remaining fields is the same as in [RFC6830]. Note, the definition of the C-bit does not interfere with the functionality provided by other flag bits.

The use of the C-bit as defined in this document is encouraged in IPv6 migration contexts that rely upon IPv4-embedded IPv6 addresses, as defined in [RFC6052]. Concretely, IPv4-embedded IPv6 addresses are used to convey Source/Destination IPv4 EIDs in Source/Destination Routing Locators.

2.2. On the Use of IPv4-Embedded IPv6 RLOCs

Figure 4 summarizes how the IPv4-embedded IPv6 RLOCs are synthesized from IPv4 EIDs. As discussed in [RFC6052], the "u" byte is set to zero.

```
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|     RLOC Prefix (64 bits)     | u |   EID(32)     | suffix    |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Where "suffix" is :

* the concatenation of "Protocol" field of the Internet header as conveyed in the original packet and "source port" of the transport header of the original packet (Source IPv4-embedded IPv6 address).

* the concatenation of a null octet and "source port" of transport header of the original packet (Source IPv4-embedded IPv6 address).

Figure 4: IPv4-embedded RLOCs

2.3. Truncated TCP Header

In addition to the use of IPv4-embedded IPv6 addresses, this document proposes the use of a truncated TCP header as shown in Figure 5 to reduce the size of the LISP header.

TCP Header:

```
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |          Source Port          |       Destination Port        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                        Sequence Number                        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                    Acknowledgment Number                      |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  Data |           |U|A|P|R|S|F|                               |
 | Offset| Reserved  |R|C|S|S|Y|I|            Window             |
 |       |           |G|K|H|T|N|N|                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |           Checksum            |         Urgent Pointer        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                     (Optional) Options                        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Truncated TCP Header:

```
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                        Sequence Number                        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                    Acknowledgment Number                      |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  Data |           |U|A|P|R|S|F|                               |
 | Offset| Reserved  |R|C|S|S|Y|I|            Window             |
 |       |           |G|K|H|T|N|N|                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                     (Optional) Options                        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: Truncated TCP Header

2.4. Compact LISP Header Format

The compact LISP header for a TCP packet is shown in Figure 6, while the compact LISP header for UDP is depicted in Figure 7.

```
      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
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   / |Version| Traffic Class |           Flow Label                  |
  /  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |   |         Payload Length        | Next Header=17|   Hop Limit   |
 |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ =
 |   |                                                               | |
 I   +                Source Routing Locator (64 bits)               + |
 P   |                                                               | S
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ R
 H   |     u byte    |         Source EID ...                        | C
 E   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
 A   |     ....      |    Protocol   |  (Inner) Source Port          | |
 D   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ =
 E   |                                                               | |
 R   +            Destination Routing Locator (64 bits)              + |
 |   |                                                               | D
 |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ S
 |   |     u byte    |          Destination EID ...                  | T
  \  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
   \ |     ....      |    00000000   |(Inner)Destination Port        | |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ =
   / |       Source Port = xxxx      |       Dest Port = 4341        |
 UDP +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   \ |           UDP Length          |        UDP Checksum           |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 L   |N|L|E|V|I|flg|C|            Nonce/Map-Version                  |
 I \ +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 S / |                 Instance ID/Locator-Status-Bits               |
 P   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                        Sequence Number                        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                    Acknowledgment Number                      |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |  Data |           |U|A|P|R|S|F|                               |
     | Offset| Reserved  |R|C|S|S|Y|I|            Window             |
     |       |           |G|K|H|T|N|N|                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                      (Optional) Options                       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Compact LISP Header Format (TCP Case)

```
      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
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   / |Version| Traffic Class |           Flow Label                  |
  /  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |   |         Payload Length        | Next Header=17|   Hop Limit   |
 |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ =
 |   |                                                               | |
 I   +                Source Routing Locator (64 bits)               + |
 P   |                                                               | S
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ R
 H   |     u byte    |         Source EID ...                        | C
 E   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
 A   |     ....      |    Protocol   |  (Inner) Source Port          | |
 D   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ =
 E   |                                                               | |
 R   +            Destination Routing Locator (64 bits)              + |
 |   |                                                               | D
 |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ S
 |   |     u byte    |          Destination EID ...                  | T
  \  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
   \ |     ....      |    00000000   |(Inner)Destination Port        | |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ =
   / |       Source Port = xxxx      |       Dest Port = 4341        |
 UDP +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   \ |           UDP Length          |        UDP Checksum           |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 L   |N|L|E|V|I|flg|C|            Nonce/Map-Version                  |
 I \ +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 S / |                 Instance ID/Locator-Status-Bits               |
 P   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: Compact LISP Header Format (UDP Case)

3. LISP Encapsulation with the Compact Header Form

Upon receipt of an IPv4 packet that needs to be forwarded over a LISP-enabled IPv6 infrastructure, the ITR proceeds as described in Section 3.1 for UDP packets and in Section 3.2 for TCP packets. Section 3.3 specifies how fragments are handled.

3.1. UDP Packets

- o Retrieve the destination/source RLOC IPv6 prefix.
- o Concatenate the source RLOC IPv6 prefix, the u byte, the source IPv4 address, the "Protocol" as indicated in the IP header, and the source port number to form the source IPv6 address as

specified in Section 2.2 for non-fragmented packets and fragments that convey a transport header.

- Concatenate the destination RLOC IPv6 prefix, the u byte, the destination IPv4 address, a null octet, and the destination port number to form the destination IPv6 address as specified in Section 2.2 for non-fragmented packets and fragments that convey a transport header.
- Remove both the IP and UDP headers of the original packet.
- Prepend the LISP header with the C-flag set (Section 2.1).
- Prepend the UDP header.
- Prepend the IPv6 header.

3.2. TCP Packets

- Retrieve the destination/source RLOC IPv6 prefix.
- Concatenate the source RLOC IPv6 prefix, the u byte, the source IPv4 address, the "Protocol" as indicated in the IP header, and the source port number to form the source IPv6 address as specified in Section 2.2 for non-fragmented packets and fragments that convey a transport header.
- Concatenate the destination RLOC IPv6 prefix, the u byte, the destination IPv4 address, a null octet, and the destination port number to form the destination IPv6 address as specified in Section 2.2 for non-fragmented packets and fragments that convey a transport header.
- Remove the IP header, the first 4 bytes of the TCP header, and the 4 bytes right after the "window" field from the original TCP header (Section 2.3).
- Prepend the LISP header with the C-flag set (Section 2.1).
- Prepend the UDP header.
- Prepend the IPv6 header.

3.3. Fragments

- Retrieve the destination/source RLOC IPv6 prefix.
- Concatenate the source RLOC IPv6 prefix, the u byte, the source IPv4 address, and 3 bytes the "Protocol" as indicated in the IP header, and 2 bytes paddings of "1".
- Concatenate the destination RLOC IPv6 prefix, the u byte, the destination IPv4 address, and 3 bytes paddings of "1".
- Remove both the IP header of the original packet.
- Prepend the LISP header with the C-flag set (Section 2.1).
- Prepend the UDP header.
- Prepend the IPv6 header.

4. LISP Decapsulation with the Compact LISP Header

Upon receipt of a LISP packet with the C-bit set, the ETR proceeds as follows to extract the inner IP packets (Section 4.1 for UDP and Section 4.2 for TCP).

The processing of the other flag bits is not detailed in this specification. Other than encoding RLOCs as prefixes, the behavior defined in [RFC6830] is not impacted by this specification.

Obviously if the C-bit is unset, xTR routers follow the behavior defined in [RFC6830].

The UDP checksum setting and validation of LISP-encapsulated packets MUST follow the guidelines documented in Section 5.3 of [RFC6830].

4.1. Build an IPv4/UDP Header

- o Check whether the destination IPv6 address matches an RLOC prefix owned by the xTR.
- o Extract the Source EID that is encoded in positions 72 to 103 of the Source IPv6 address.
- o Extract the "Protocol" field that is encoded in positions 104 to 111 of the Source IPv6 address. This value is used to set the corresponding field in the IPv4 header of the de-capsulated packet.
- o Extract the Source Port that is encoded in positions 112 to 127 of the Source IPv6 address, for non-fragmented packets and fragments that convey a transport header.
- o Extract the Destination EID that is encoded in positions 72 to 103 of the Destination IPv6 address.
- o Extract the Destination Port that is encoded in positions 112 to 127 of the Destination IPv6 address, for non-fragmented packets and fragments that convey a transport header.
- o Remove the IPv6 header, the UDP header, and the LISP header.
- o Use the extracted Source Port and Destination Port to build the UDP header. Prepend the new UDP header.
- o Use the extracted Source IP address, Destination IP address, and Protocol to build the IPv4 header.
- o Prepend the new IPv4 header.

4.2. Build an IPv4/TCP Header

- o Check whether the destination IPv6 address matches an RLOC prefix owned by the xTR.
- o Extract the Source EID that is encoded in positions 72 to 103 of the Source IPv6 address.

- o Extract the "Protocol" field that is encoded in positions 104 to 111 of the Source IPv6 address. This value is used to set the corresponding field in the IPv4 header of the de-capsulated packet.
- o Extract the Source Port that is encoded in positions 112 to 127 of the Source IPv6 address, for non-fragmented packets and fragments that convey a transport header.
- o Extract the Destination EID that is encoded in positions 72 to 103 of the Destination IPv6 address.
- o Extract the Destination Port that is encoded in positions 112 to 127 of the Destination IPv6 address, for non-fragmented packets and fragments that convey a transport header.
- o Remove the IPv6 header, UDP header, and LISP header.
- o For non-fragmented packets and fragments that convey a transport header, prepend 4 bytes with the source/destination port number and insert 4 bytes right after the "window" field to build a proper TCP header. The extracted Source Port and Destination Port are used in this step.
- o Prepend an IPv4 header. Use the extracted Source IP address, Destination IP address, and Protocol to build the IPv4 header.

5. A More Compact LISP Encapsulation Flavor

A more compact LISP encapsulation scheme can be considered if the following conditions are met:

- o Compatibility with "u" byte is not required.
- o The origin "Source Port" number is copied into the UDP header of the encapsulated packet, and vice versa.
- o The LISP shim is split into two parts: 4 bytes that are placed right after the UDP header while "Instance ID/Locator-Status-Bits" are encoded in the last 32 bits of the source IPv4-embedded IPv6 RLOC.

This alternate proposal leads to a 4-byte overhead when transporting IPv4-over-IPv6 LISP packets for both TCP (Figure 8) and UDP (Figure 9).

```
      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
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   / |Version| Traffic Class |           Flow Label                  |
  /  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |   |         Payload Length        | Next Header=17|   Hop Limit   |
 |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ =
 |   |                                                               | |
 I   +                Source Routing Locator (64 bits)               + |
 P   |                                                               | S
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ R
 H   |                          Source EID                           | C
 E   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
 A   |                 Instance ID/Locator-Status-Bits               | |
 D   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ =
 E   |                                                               | |
 R   +            Destination Routing Locator (64 bits)              + |
 |   |                                                               | D
 |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ S
 |   |                     Destination EID                           | T
  \  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
   \ |    00000000   |(Inner)Protocol|(Inner)Destination Port        | |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ =
   / | Source Port = Inner Src Port  |       Dest Port = 4341        |
 UDP +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   \ |           UDP Length          |        UDP Checksum           |
 L \ +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 I | |N|L|E|V|I|flg|C|            Nonce/Map-Version                  |
 S / +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 P   |                        Sequence Number                        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                    Acknowledgment Number                      |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |  Data |           |U|A|P|R|S|F|                               |
     | Offset| Reserved  |R|C|S|S|Y|I|            Window             |
     |       |           |G|K|H|T|N|N|                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                      (Optional) Options                       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: More Compacted LISP Header Format (TCP Case)

```
      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
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   / |Version| Traffic Class |           Flow Label                  |
  /  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |   |         Payload Length        | Next Header=17|   Hop Limit   |
 |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ =
 |   |                                                               | |
 I   +                Source Routing Locator (64 bits)               + |
 P   |                                                               | S
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ R
 H   |                          Source EID                           | C
 E   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
 A   |                 Instance ID/Locator-Status-Bits               | |
 D   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ =
 E   |                                                               | |
 R   +            Destination Routing Locator (64 bits)              + |
 |   |                                                               | D
 |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ S
 |   |                     Destination EID                           | T
  \  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
   \ |    00000000   |(Inner)Protocol|(Inner)Destination Port        | |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ =
   / | Source Port = Inner Src Port  |       Dest Port = 4341        |
 UDP +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   \ |           UDP Length          |        UDP Checksum           |
 L \ +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 I | |N|L|E|V|I|flg|C|            Nonce/Map-Version                  |
 S / +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 P
```

Figure 9: More Compacted LISP Header Format (UDP Case)

5.1. LISP Encapsulation with the More Compact Header Form

Upon receipt of an IPv4 packet that needs to be forwarded over a LISP-enabled infrastructure, the ITR proceeds as follows:

5.1.1. UDP Packets

- o Retrieve the destination/source RLOC IPv6 prefix.
- o Concatenate the source RLOC IPv6 prefix, the source IPv4 address, and the "Instance ID/Locator-Status-Bits" to form the source IPv6 address as shown in Figure 9.
- o Concatenate the destination RLOC IPv6 prefix, the destination IPv4 address, a null octet, the "Protocol" as indicated in the IP header, and the destination port number to form the destination

IPv6 address as shown in Figure 9, for non-fragmented packets and fragments that convey a transport header.

- Remove the IPv4 header.
- Set the destination port number of the UDP header to 4341.
- Insert the LISP header right after the UDP header; the C-flag must be set (Section 2.1).
- Prepend the IPv6 header.

5.1.2. TCP Packets

- Retrieve the destination/source RLOC IPv6 prefix.
- Concatenate the source RLOC IPv6 prefix, the source IPv4 address, and the "Instance ID/Locator-Status-Bits" to form the source IPv6 address as shown in Figure 8.
- Concatenate the destination RLOC IPv6 prefix, the destination IPv4 address, a null octet, the "Protocol" as indicated in the IP header, and the destination port number to form the destination IPv6 address as shown in Figure 8, for non-fragmented packets and fragments that convey a transport header.
- Remove the IPv4 header.
- Remove the first 4 bytes and the 4 bytes right after the "window" field of the TCP header (Section 2.3).
- Prepend the LISP header; the C-flag must be set (Section 2.1)..
- Prepend the UDP header. Set to the source port number to the same port indicated in the original TCP header. Set the destination port number of the UDP header to 4341.
- Prepend an IPv6 header.

5.1.3. Fragments

- Retrieve the destination/source RLOC IPv6 prefix.
- Concatenate the source RLOC IPv6 prefix, the source IPv4 address, and the "Instance ID/Locator-Status-Bits" to form the source IPv6 address as shown in Figure 9.
- Concatenate the destination RLOC IPv6 prefix, the destination IPv4 address, a non-null octet, the "Protocol" as indicated in the IP header, and a null octet padding to form the destination IPv6 address.
- Remove the IPv4 header.
- Insert the LISP header.
- Insert the UDP header with a destination port number set to 4341.
- Prepend the IPv6 header.

5.2. LISP Decapsulation with the More Compact LISP Header

Upon receipt of a LISP packet with the C-bit set, the ETR proceeds as follows to extract the inner IP packets: (Figure 9 for UDP and Figure 8 for TCP). Like in Section 2, the UDP checksum setting and

validation of LISP-encapsulated packets MUST follow the guidelines documented in Section 5.3 of [RFC6830].

5.2.1. Build an IPv4/UDP Header

- o Check whether the destination IPv6 address matches an RLOC prefix owned by the xTR.
- o Extract the Source EID that is encoded in positions 64 to 95 of the Source IPv6 address.
- o Extract the "Instance ID/Locator-Status-Bits" field that is encoded in positions 96 to 127 of the Source IPv6 address.
- o Extract the Destination EID that is encoded in positions 64 to 95 of the Destination IPv6 address.
- o Extract the "Protocol" that is encoded in positions 104 to 111 of the Destination IPv6 address.
- o Extract the Destination Port that is encoded in positions 112 to 127 of the Destination IPv6 address if the octet in positions 96 to 103 is not null.
- o Remove the IPv6 header, the UDP header, and the LISP header.
- o For non-fragmented packets and fragments that convey a transport header, use the extracted Source Port and Destination Port to build the UDP header. Prepend the new UDP header.
- o Use the extracted Source IPv4 address, Destination IPv4 address, and Protocol to build the IPv4 header. Prepend the new IPv4 header.

5.2.2. Build an IPv4/TCP Header

- o Check whether the destination IPv6 address matches an RLOC prefix owned by the xTR.
- o Extract the Source EID that is encoded in positions 64 to 95 of the Source IPv6 address.
- o Extract the "Instance ID/Locator-Status-Bits" field that is encoded in positions 96 to 127 of the Source IPv6 address.
- o Extract the Destination EID that is encoded in positions 64 to 95 of the Destination IPv6 address.
- o Extract the "Protocol" that is encoded in positions 104 to 111 of the Destination IPv6 address.
- o Extract the Destination Port that is encoded in positions 112 to 127 of the Destination IPv6 address if the octet in positions 96 to 103 is not null.
- o Check whether the destination IPv6 address matches an RLOC prefix owned by the xTR.
- o Remove the IPv6 header, UDP header, and LISP header.
- o For non-fragmented packets and fragments that convey a transport header, prepend 4 bytes with the source/destination port number and insert 4 bytes right after the "window" field to build a

proper TCP header. The extracted Source Port and Destination Port are used during this step.

- o Prepend an IPv4 header. Use the extracted Source IP address, Destination IP address, and Protocol to build the IPv4 header.

6. Discussion

The proposed compact headers are experimental. What primarily motivates this specification is the need to assess its technical feasibility thanks to an existing LISP-enabled platform. Experiments will help evaluate the gain brought by using such compact headers compared to base LISP encapsulation scheme in typical IPv6 migration scenarios

The proposed compact encapsulation schemes guarantee a functional parity with the base LISP specification, given that the signalling carried in a LISP packet remains usable.

This specification does not include any capability checks to ensure that remote xTRs support the proposed header encoding. Particularly, deployability considerations in multi-domain LISP environments are not detailed in this document.

This specification assumes that a configuration parameter should be supported by LISP implementations to tweak the encapsulation scheme to be used.

The handling of fragmented packets by an ETR follows the same steps as in Section 2 except that, for the fragments that do not carry the source/destination port numbers, a non-null octet of the "suffix" defined Figure 4 is used to signal that the LISP-encapsulated packet is a fragment that does not convey transport-related information.

7. Security Considerations

The security considerations discussed in Section 12 of[RFC6830] are valid for this document.

Security considerations related to building an IPv4-embedded IPv6 address are discussed in [RFC6052].

8. IANA Considerations

This document does not make any request to IANA.

9. Acknowledgments

This work is partly funded by ANR LISP-Lab project #ANR-13-INFR-009-X.

Many thanks to S. Secci, L. Iannone, and J. Saldana for the review and comments.

The gain ratio table is a courtesy of J. Saldana.

10. Normative references

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <http://www.rfc-editor.org/info/rfc768>.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <http://www.rfc-editor.org/info/rfc793>.

[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>.

[RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, DOI 10.17487/RFC6052, October 2010, <http://www.rfc-editor.org/info/rfc6052>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <http://www.rfc-editor.org/info/rfc6830>.

Authors' Addresses

Mohamed Boucadair
Orange
Rennes 35000
France

EMail: mohamed.boucadair@orange.com

Christian Jacquenet
Orange
Rennes 35000
France

EMail: christian.jacquenet@orange.com