Network Working Group
Internet-Draft
Intended status: Experimental
Expires: May 6, 2021

V. Ermagan
Google
D. Farinacci
lispers.net
D. Lewis
F. Maino
M. Portoles
Cisco Systems, Inc.
J. Skriver
Arista
C. White
Logicalelegance, Inc.
A. Lopez
A. Cabellos
UPC/BarcelonaTech
November 2, 2020

NAT traversal for LISP
draft-ermagan-lisp-nat-traversal-18

Abstract

This document describes a mechanism for IPv4 NAT traversal for LISP tunnel routers (xTR) and LISP Mobile Nodes (LISP-MN) behind a NAT device. A LISP device both detects the NAT and initializes its state. Forwarding to the LISP device through a NAT is enabled by the LISP Re-encapsulating Tunnel Router (RTR) network element, which acts as an anchor point in the data plane, forwarding traffic from unmodified LISP devices through the NAT.

Copyright Notice

Table of Contents

1. Introduction

The Locator/ID Separation Protocol [I-D.ietf-lisp-rfc6830bis] [I-D.ietf-lisp-rfc6833bis]defines a set of functions for encapsulating routers to exchange information used to map from Endpoint Identifiers (EIDs) to routable Routing Locators (RLOCs). The assumption that the LISP Tunnel Routers are reachable at their RLOC breaks when a LISP device is behind a NAT. LISP relies on the xTR being able to receive traffic at its RLOC on destination port 4341. However nodes behind a NAT are only reachable through the NAT's public address and in most cases only after the appropriate mapping state is set up in the NAT. Depending on the type of the NAT device, this mapping state may be address and port dependent. In other words, the mapping state in the NAT device may be associated with the 5 tuple that forms a specific flow, preventing incoming traffic from any LISP router other than the one associated with the 5 tuple. A NAT traversal mechanism is needed to make the LISP device behind a NAT reachable.

This document briefly discusses available NAT traversal options, and then it introduces in detail a NAT traversal mechanism for LISP. Two new LISP control messages - LISP Info-Request and LISP Info-Reply - are introduced in order to detect whether a LISP device is behind a NAT, and discover the global IP address and global ephemeral port used by the NAT to forward LISP packets sent by the LISP device. A new LISP component, the LISP Re-encapsulating Tunnel Router (RTR), acts as a re-encapsulating LISP tunnel router [I-D.ietf-lisp-rfc6830bis] to pass traffic through the NAT, to and from the LISP device. A modification to how the LISP Map-Register messages are sent allows LISP device to initialize NAT state to use the RTR services. This mechanism addresses the scenario where the LISP device is behind the NAT, but the associated Map-Server [I-D.ietf-lisp-rfc6833bis] is on the public side of the NAT.

2. Definition of Terms

LISP Info-Request: A LISP control message sent by a LISP device to its Map-Server.

LISP Info-Reply: A LISP control message sent by a Map Server to a LISP device in response to an Info-Request control message.

LISP Re-encapsulating Tunnel Router (RTR): An RTR is a re-encapsulating LISP Router (see [I-D.ietf-lisp-rfc6830bis]). One function that an RTR provides is enabling a LISP device to traverse NATs.

LISP Data-Map-Notify: A LISP Map-Notify message encapsulated in a LISP data header.

LISP xTR-ID A 128-bit field that, together with a site-ID, can be appended at the end of a Map-Register or Map-Notify message. An xTR-ID is used as a unique identifier of the xTR that is sending the Map-Register and is especially useful for identifying multiple xTRs serving the same site/EID-prefix. A value of all zeros indicate the xTR-ID is unspecified.

LISP site-ID A 64-bit field that, together with a xTR-ID, can be appended at the end of a Map-Register or Map-Notify message. A site-ID is used as a unique identifier of a group of xTRs belonging to the same site. A value of 0 indicate the site-ID is unspecified.

NAT: "Network Address Translation is a method by which IP addresses are mapped from one address realm to another, providing transparent routing to end hosts". "Traditional NAT would allow hosts within a private network to transparently access hosts in the external network, in most cases. In a traditional NAT, sessions are uni-directional, outbound from the private network." --RFC 2663 [NAT]. Basic NAT and NAPT are two varieties of traditional NAT.

Basic NAT: "With Basic NAT, a block of external addresses are set aside for translating addresses of hosts in a private domain as they originate sessions to the external domain. For packets outbound from the private network, the source IP address and related fields such as IP, TCP, UDP and ICMP header checksums are translated. For inbound packets, the destination IP address and the checksums as listed above are translated." --RFC 2663[NAT].

NAPT: "NAPT extends the notion of translation one step further by also translating transport identifier (e.g., TCP and UDP port numbers, ICMP query identifiers). This allows the transport identifiers of a number of private hosts to be multiplexed into the transport identifiers of a single external address. NAPT allows a set of hosts to share a single external address. Note that NAPT can be combined with Basic NAT so that a pool of external addresses are used in conjunction with port translation." --RFC 2663[NAT]. Transport identifiers of the destination hosts are not modified by the NAPT.

In this document the general term NAT is used to refer to both Basic NAT and NAPT.

While this document specifies LISP NAT Traversal for LISP tunnel routers, a LISP-MN can also use the same procedure for NAT traversal. The modifications attributed to a LISP-Device, xTR, ETR, and ITR must be supported by a LISP-MN where applicable, in order to achieve NAT traversal for such a LISP node. A NAT traversal mechanism for LISP-MN is also proposed in [NAT-MN].

For definitions of other terms, notably Map-Request, Map-Reply, Ingress Tunnel Router (ITR), and Egress Tunnel Router (ETR), please consult the LISP specification [I-D.ietf-lisp-rfc6833bis].

3. Basic Overview

There are a variety of NAT devices and a variety of network topologies utilizing NAT devices in deployments. Most NAT devices deployed today are designed primarily around the client/server paradigm, where client machines inside a private network initiate connections to public servers with public IP addresses. As such, any protocol requiring a device or host in a private network behind a NAT to receive packets or accept sessions from destinations without first initiating a session or sending packets towards those destinations, will be challenged by deployed NAT devices.

NAT devices are loosely classified based on how restrictive they are. These classifications are essentially identifying the type of mapping state that the NAT device is requiring to allow incoming traffic. For instance, the mapping state may be end-point independent: once device A inside the private network sends traffic to a destination outside, a mapping state in the NAT is created that only includes information about device A, namely its IP address and perhaps its port number. Once this mapping is established in the NAT device, any external device with any IP address could send packets to device A. More restrictive NAT devices could include the 5 tuple information of the flow as part of the mapping state, in other words, the mapping state in the NAT is dependent upon Source IP and Port, as well as destination IP and port (symmetric NAT or Endpoint-dependent NAT). Such a NAT only allows traffic from the specified destination IP and port to reach the specified source device on the specified source port. Traffic with a different 5 tuple signature will not be allowed to pass. In general, in the case of less restrictive NATs it may be possible to eventually establish direct peer-to-peer connections, by means of various hole punching techniques and initial rendezvous servers. However, in the case of symmetric NATs or NATs with endpoint-address-and-port-dependent mappings, direct connection may prove impossible. In such cases a relay device is required that is in the public Network and can relay packets between the two endpoints.

Various methods have been designed to address NAT traversal challenges, mostly in the context of peer-to-peer applications and protocols. Among these, the Interactive Connectivity Establishment (ICE) [ICE] seems the most comprehensive, which defines a protocol that leverages other protocols such as Session Traversal Utilities for NAT(STUN) [STUN] and Traversal Using Relays around NAT (TURN) [TURN], as well as a rendezvous server to identify and exchange a list of potential transport (IP and Port) addresses between the two endpoints. All possible pairs of transport addresses are exhaustively tested to find the best possible option for communication, preferring direct connection to connections using a relay. In the case of most restrictive NATs, ICE leads to use of TURN servers as relay for the traffic. TURN requires a list of allowed peer IP addresses defined as permissions, before allowing a peer to use the relay server to reach a TURN client.

Common NAT traversal techniques such as ICE generally assume bi-directional traffic with the same 5 tuple. LISP, however, requires traffic to use destination UDP port 4341, without specifying the source port. As a result, LISP traffic is generally uni-directional. This means that, in the case of symmetric or endpoint-address-and-port-dependent mapping NATs, even when an outgoing mapping is established, still incoming traffic may not match the established mapping and will not be allowed to pass. As a result, while ICE may be used to traverse less restrictive NATs, use of standard TURN servers as relays to traverse symmetric NATs for LISP protocol is not possible. The rest of this document specifies a NAT traversal technique for the LISP protocol that enables LISP protocol to traverse multiple types of NATs including symmetric NATs.

3.1. LISP NAT Traversal Overview

There are two attributes of a LISP device behind a typical NAT that requires special consideration in LISP protocol behavior in order to make the device reachable. First, the RLOC assigned to the device is typically not globally unique nor globally routable. The NAT likely has a restrictive translation table and forwarding policy, requiring outbound packets to create state before the NAT accepts inbound packets. Second, LISP protocol requires an xTR to receive traffic on a specific UDP port 4341, so the random UDP port allocated by the NAT on its public side to associate with a xTR behind the NAT can not be used by other xTRs to send LISP traffic to. This section provides an overview of the LISP NAT traversal mechanism which deals with these conditions. The following sections specify the mechanism in more detail.

When a LISP device receives a new RLOC and wants to register it with the mapping system, it needs to first discover whether it is behind a

NAT. To do this, an ETR queries its Map-Server to discover the ETR's translated global RLOC and port via the two new LISP messages: Info-Request and Info-Reply. Once an ETR detects that it is behind a NAT, it uses a LISP Re-encapsulating Tunnel Router (RTR) entity as an anchor point for sending and receiving data plane traffic through the NAT device. The ETR registers the RTR RLOC(s) to its Map-Server using the RTR as a proxy for the Map-Register message. The ETR encapsulates the Map-Register message in a LISP ECM header destined to the RTR's RLOC. The RTR strips the LISP ECM header and sends it to the Map-Server. This initializes state in the NAT device so the ETR can receive traffic on port 4341 from the RTR. The ETR also registers the RTR RLOC as the RLOC where the ETR EID prefix is reachable. As a result, all packets destined to the ETR's EID will go to its RTR. The RTR will then re-encapsulate and forward the ETR's traffic via the existing NAT state to the ETR.

Outbound LISP data traffic from the xTR should also be encapsulated to the RTR, where the RTR de-capsulates the LISP packets, and then re-encapsulates them or forwards them natively depending on their destination.

In the next sections these procedures are discussed in more detail.

This document does not support different xTR-ID registering the same EID prefix and using the same set of RTRs. Future versions of this spec will explore this use-case.

4. LISP RTR Message Details

The main modifications in the LISP protocol to enable LISP NAT traversal via an RTR include: (1) two new messages used for NAT discovery (Info-Request and Info-Reply), and (2) encapsulation of two LISP control messages (Map-Register and Map-Notify) between the xTR and the RTR. Map-Register is encapsulated in an ECM header while Map-Notify is encapsulated in a LISP data header (Data-Map-Notify). This section describes the message formats and details of the Info-Request, Info-Reply, and Data-Map-Notify messages, as well as encapsulation details and minor changes to Map-Register and Map-Notify messages.

4.1. Info-Request Message

An ETR sends an Info-Request message to its Map-Server in order to

1. detect whether there is a NAT on the path to its Map-Server

2. obtain a list of RTR RLOCs that can be used for LISP data plane NAT traversal.

An Info-Request message is a LISP control message, its source port is chosen by the xTR and its destination port is set to 4342.

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |Type=7 |R|            Reserved                                 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                         Nonce . . .                           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                      . . . Nonce                              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |              Key ID           |  Authentication Data Length   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    ~                     Authentication Data                       ~
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                              TTL                              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Reserved    | EID mask-len  |        EID-prefix-AFI         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                          EID-prefix                           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             AFI = 0           |   <Nothing Follows AFI=0>     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

LISP Info-Request Message Format

Type: 7 (Info-Request)

R: R bit indicates this is a reply to an Info-Request (Info-Reply). R bit is set to 0 in an Info-Request. When R bit is set to 0, the AFI field (following the EID-prefix field) must be set to 0. When R bit is set to 1, the packet contents follow the format for an Info-Reply, as described below.

Reserved: Must be set to 0 on transmit and must be ignored on receipt.

TTL: The time in minutes the recipient of the Info-Reply will store the RTR Information.

Nonce: An 8-byte random value created by the sender of the Info-Request. This nonce will be returned in the Info-Reply. The nonce SHOULD be generated by a properly seeded pseudo-random (or strong random) source. Future version of this document will discuss anti-replay mitigation mechanisms

Descriptions for other fields can be found in the Map-Register section of [I-D.ietf-lisp-rfc6833bis]. Field descriptions for the LCAF AFI = 0 can be found in the LISP LCAF RFC [LCAF] .

4.2. LISP Info-Reply

When a Map-Server receives an Info-Request message, it responds with an Info-Reply message. The Info-Reply message source port is 4342, and destination port is taken from the source port of the triggering Info-Request. Map-Server fills the NAT LCAF (LCAF Type = 7) fields according to their description. The Map-Server uses AFI=0 for the Private ETR RLOC Address field in the NAT LCAF.

```
       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
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |Type=7 |R|               Reserved                              |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         Nonce . . .                           |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                      . . . Nonce                              |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |            Key ID             |  Authentication Data Length   |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      ~                     Authentication Data                       ~
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                              TTL                              |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |   Reserved    | EID mask-len  |        EID-prefix-AFI         |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                          EID-prefix                           |
   +->+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  |           AFI = 16387         |    Rsvd1      |     Flags     |
   |  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  |    Type = 7   |     Rsvd2     |             4 + n             |
   |  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   N  |        MS UDP Port Number     |      ETR UDP Port Number      |
   A  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   T  |              AFI = x          | Global ETR RLOC Address  ...  |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   L  |              AFI = x          |       MS RLOC Address  ...    |
   C  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   A  |              AFI = x          | Private ETR RLOC Address ...  |
   F  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  |              AFI = x          |      RTR RLOC Address 1 ...   |
   |  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  |              AFI = x          |       RTR RLOC Address n ...  |
   +->+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

LISP Info-Reply Message Format

Type: 7 , R = 1, (Info-Reply)

The format is similar to the Info-Request message. See Info-Request section for field descriptions. Field descriptions for the NAT LCAF section can be found in the LISP LCAF RFC [LCAF] .

4.3. LISP Map-Register Message

The third bit after the Type field in the Map-Register message is allocated as "I" bit. I bit indicates that a 128 bit xTR-ID and a 64 bit site-ID field are present at the end of the Map-Register message. If an xTR is configured with an xTR-ID or site-ID, it MUST set the I bit to 1 and include its xTR-ID and site-ID in the Map-Register messages it generates. If either the xTR-ID or site-ID is not configured an all zeros value is encoded for whichever ID that is not configured.

xTR-ID is a 128 bit field at the end of the Map-Register message, starting after the final Record in the message. The xTR-ID is used to identify the intended recipient xTR for a Map-Notify message, especially in the case where a site has more than one xTR. A value of all zeros indicate that an xTR-ID is not specified, though encoded in the message. This is useful in the case where a site-ID is specified, but no xTR-ID is configured. When a Map-Server receives a Map-Register with an xTR-ID specified (I bit set and xTR-ID has a non-zero value), it MUST copy the XTR-ID from the Map-Register to the associated Map-Notify message. When a Map-Server is sending an unsolicited Map-Notify to an xTR to notify the xTR of a change in locators, the Map-Server must include the xTR-ID for the intended recipient xTR, if it has one stored locally.

site-ID is a 64 bit field at the end of the Map-Register message, following the xTR-ID. site-ID is used by the Map-Server receiving the Map-Register message to identify which xTRs belong to the same site. A value of 0 indicate that a site-ID is not specified, though encoded in the message. When a Map-Server receives a Map-Register with a site-ID specified (I bit set and site-ID has non-zero value), it must copy the site-ID from the Map-Register to the associated Map-Notify message. When a Map-Server is sending an unsolicited Map-Notify to an xTR to notify the xTR of a change in locators, the Map-Server must include the site-ID for the intended recipient xTR, if it has one stored locally.

A LISP device that sends a Map-Register to an RTR must encapsulate the Map-Register message using an Encapsulated Control Message (ECM) [I-D.ietf-lisp-rfc6833bis]. The 6th bit in the ECM LISP header is allocated as the "R" bit. The R bit indicates that the encapsulated Map-Register is to be processed by an RTR. The 7th bit in the ECM header is allocated as the "N" bit. The N bit indicates that this Map-Register is being relayed by an RTR. When an RTR relays the ECM-ed Map-Register to a Map-Server, the N bit must be set to 1.

The outer header source RLOC of the ECM is set to the LISP device's local RLOC, and the outer header source port is set to 4341. The outer header destination RLOC and port are set to RTR RLOC and 4342 respectively. The inner header source RLOC is set to LISP device's local RLOC, and the inner source port is picked at random. The inner header destination RLOC is set to the xTR's Map-Server RLOC, and inner header destination port is set to 4342.

4.4. LISP Map-Notify

The first bit after the Type field in a Map-Notify message is allocated as the "I" bit. I bit indicates that a 128 bit xTR-ID and 64 bit site-ID field is present at the end of the Map-Notify message, following the final Record in the Map-Notify (See Section 4.3 for details on xTR-ID and site-ID). A Map-Server MUST set the I bit in a Map-Notify and include the xTR-ID and/or site-ID of the intended recipient xTR if the associated Map-Register has an xTR-ID and/or site-ID specified, or when the Map-Server has previously cached an xTR-ID and/or site-ID for the destination xTR.

A LISP device that sends a Map-Notify to an RTR must encapsulate the Map-Notify message using an ECM. The 6th bit in the ECM LISP header, allocated as the "R" bit, must be set when the encapsulated Map-Notify is to be processed by an RTR. If the S bit is also set in the Map-Notify ECM header, it indicates that additional MS-RTR authentication data is included after the LISP header in the ECM. If the I bit is also set in the Map-Notify, the xTR-ID and site-ID fields are included in the Map-Notify. If a Map-Server receiving an ECM-ed Map-Register has a shared key associated with the sending RTR, it must generate a Map-Notify message with the S bit in the ECM header set to 1, and with the additional MS-RTR authentication related fields described below.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    AD Type    |                  Reserved                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        MS-RTR Key ID          |  MS-RTR Auth. Data Length     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                MS-RTR Authentication Data                     ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Changes to LISP Map-Notify Message

AD Type: 2 (RTR Authentication Data)

MS-RTR Key ID: A configured ID to find the configured Message Authentication Code (MAC) algorithm and key value used for the authentication function. See [I-D.ietf-lisp-rfc6833bis] section 12.5 for code point assignments.

MS-RTR Authentication Data Length: The length in bytes of the MS-RTR Authentication Data field that follows this field. The length of the Authentication Data field is dependent on the Message Authentication Code (MAC) algorithm used. The length field allows a device that doesn't know the MAC algorithm to correctly parse the packet.

MS-RTR Authentication Data: The message digest used from the output of the Message Authentication Code (MAC) algorithm. The entire Map-Notify payload is authenticated. After the MAC is computed, it is placed in this field. Implementations of this specification MUST support HMAC-SHA-1-96 [RFC2404] and SHOULD support HMAC-SHA-256-128 [RFC6234].

For a full description of all fields in the Map-Notify message refer to Map-Notify section in [I-D.ietf-lisp-rfc6833bis].

The outer header source RLOC of the ECM is set to the xTR's Map-Server RLOC, and the outer header source port is set to 4342. The outer header destination RLOC and port are set to RTR's RLOC and 4342 respectively. The inner header source RLOC is set to the xTR's Map-Server RLOC, and the inner source port is set to 4342. The inner header destination RLOC is set to the LISP device's local RLOC copied from the Map-Register, and inner header destination port is set to 4342.

4.5. LISP Data-Map-Notify Message

When an RTR receives an ECM-ed Map-Notify message with R bit in the ECM header set to 1, it has to relay the Map-Notify payload to the registering LISP device. After removing the ECM header and processing the Map-Notify message as described in Section 5.3, the RTR encapsulates the Map-Notify in a LISP data header and sends it to the associated LISP device. This Map-Notify inside a LISP data header is referred to as a Data-Map-Notify message.

```
        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
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |                       IPv4 or IPv6 Header                     |
   OH  |                      (uses RLOC addresses)                    |
     \ |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |       Source Port = 4342      |       Dest Port = xxxx        |
   UDP +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     \ |           UDP Length          |        UDP Checksum           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   L   |                           LISP Header  ~                      |
   I \ +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   S / |                           ~ LISP Header                       |
   P   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |                       IPv4 or IPv6 Header                     |
   IH  |                  (uses RLOC or EID addresses)                 |
     \ |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |       Source Port = 4342      |       Dest Port = 4342        |
   UDP +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     \ |           UDP Length          |        UDP Checksum           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   LCM |                      LISP Map-Notify Message                  ~
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

LISP Data-Map-Notify Message

In a Data-Map-Notify, the outer header source RLOC is set to the RTR's RLOC that was used in the associated Map-Register. This is previously cached by the RTR. The outer header source port is set to 4342. The outer header destination RLOC and port are filled based on the translated global RLOC and port of the registering LISP device previously stored locally at the RTR. The inner header source address is Map-Server's RLOC, and inner header source port is 4342. The inner header destination address is the LISP device's local RLOC. The inner header destination port is 4342.

Since a Data-Map-Notify is a control message encapsulated in a LISP data header, a special Instance ID is used as a signal for the xTR to trigger processing of the control packet inside the data header. The Instance ID value 0xFFFFFF is reserved for this purpose. The Instance ID field in a Data-Map-Notify must be set to 0xFFFFFF.

5. Protocol Operations

There are two main steps in the NAT traversal procedure. First, the ETR's translated global RLOC must be discovered. Second, the NAT translation table must be primed to accept incoming connections. At the same time, the RTR must be informed of the ETR's translated global RLOC including the translated ephemeral port number(s) at which the RTR can reach the LISP device.

5.1. xTR Processing

Upon receiving a new local RLOC, an ETR first has to detect whether the new RLOC is behind a NAT device. For this purpose the ETR sends an Info-Request message to its Map-Server in order to discover the ETR's translated global RLOC as it is visible to the Map-Server. The ETR uses its new local RLOC as the source RLOC of the message. The Map-Server, after authenticating the message, responds with an Info-Reply message. The Map-Server includes the source RLOC and port from the Info-Request message in the Global ETR RLOC Address and ETR UDP Port Number fields of the Info-Reply. The Map Server also includes the destination RLOC and port number of the Info-Request message in the MS RLOC Address and MS UDP Port Number fields of the Info-Reply. In addition, the Map-Server provides a list of RTR RLOCs that the ETR may use in case it needs NAT traversal services. The source port of the Info-Reply is set to 4342 and the destination port is copied from the source port of the triggering Info-Request message.

Upon receiving the Info-Reply message, the ETR compares the source RLOC and source port used for the Info-Request message with the Global ETR RLOC Address and ETR UDP Port Number fields of the Info-Reply message. If the two are not identical, the ETR concludes that its new local RLOC is behind a NAT and that it requires an RTR for NAT traversal services in order to be reachable at that RLOC. An ETR behind other statefull devices (e.g. statefull firewalls) may also use an RTR and the procedure specified here for traversing the statefull device. Detecting existence of such devices are beyond scope of this document.

It is worth noting that a STUN server can also be used to do NAT detection and to discover the NAT-translated public IP address and port number for the ETR behind NAT. If a STUN server is used, list of RTR devices that can be used by the xTR for NAT traversal must be provisioned to the xTR via other means which are outside the scope of this document.

If there is no NAT on the path identified by an info-Request and an Info-Reply, the ETR registers the associated RLOC with its Map-Server as described in [I-D.ietf-lisp-rfc6833bis].

A device that is not behind a NAT can still choose to use RTR as an anchor point. In some deployments, devices can roam from NATed connections to non-NATed connections. In such scenarios, the deployer may choose to always use an RTR, thus avoiding frequent notifications of RLOC changes. The main disadvantage of this policy is that RTRs may increase path length.

5.1.1. ETR Registration

Once an ETR has detected that it is behind a NAT, based on local policy the ETR selects one (or more) RTR(s) from the RTR RLOCs provided in the Info-Reply and initializes state in the NAT device in order to receive LISP data traffic on UDP port 4341 from the selected RTR. To do so, the ETR sends a Map-Register encapsulated in an ECM header to the selected RTR(s). The Map-Register message is created as specified in [I-D.ietf-lisp-rfc6833bis]. More specifically, the source RLOC of the Map-Register is set to ETR's local RLOC, while the destination RLOC is set to the ETR's Map-Server RLOC, and destination port is set to 4342. The ETR sets the P bit (proxy Map-Reply) and the M bit (want-Map-Notify) in Map-Register to 1, and it includes the selected RTR RLOC(s) as the locators in the Map-Register message. The ETR can also include its local RLOCs as locators in the Map-Register, including weight and priorities, while setting the R bit to 0 for each local RLOC. This can be used by the RTR for load balancing when forwarding data to a multi-homed xTR behind a NAT. The R bit is set to 1 for all RTR locators included in the Map-Register. The ETR must also set the I bit in the Map-Register message to 1 and include its xTR-ID and site-ID in the corresponding field. In the ECM header of this Map-Register the source RLOC is set to ETR's local RLOC and the source port is set to 4341, while the destination RLOC is the RTR's RLOC and the destination port is set to LISP control port 4342. The R bit in the ECM header is also set to 1, to indicate that this EDCM-ed Map-Register is to be processed by an RTR.

This ECM-ed Map-Register is then sent to the RTR. The RTR removes the ECM header, encapsulates the new Map-Register in a new ECM header with R bit set to 0, and sends it to the associated Map-Server. The RTR then encapsulates the corresponding Map-Notify message in a LISP data header (Data-Map-Notify) and sends it back to the xTR.

Upon receiving a Data-Map-Notify from the RTR, the ETR must strip the outer LISP data header, and process the inner Map-Notify message as described in [I-D.ietf-lisp-rfc6833bis]. Since outer header destination port in Data-Map-Notify is set to LISP data port 4341, the Instance ID 0xFFFFFF in the LISP header of the Data-Map-Notify is used by the ETR to detect and process the Data-Map-Notify as a control message encapsulated in a LISP data header. While processing the Data-Map-Notify, the xTR also stores the RTR RLOC(s) as its data plane proxy for the interface/RLOC behind the NAT.

If the xTR is not multi-homed, or if all its interfaces are behind the NAT and will use the same RTR, then the xTR should map the EID prefix 0/0 to this RTR RLOC(s) in its map-cache. This results in the xTR encapsulating all LISP data plane traffic to this RTR, reducing the state created in the NAT. Note that not installing the default map-cache entry will lead to normal Map-Request and Map-Reply messages for EID mapping lookups which is only supported if the xTR has interfaces not behind NAT. If outgoing traffic is sent directly to destinations without passing through the RTR, this will result in additional state to be created in the NAT device.

At this point the registration and state initialization is complete and the xTR can use the RTR services. The state created in the NAT device based on the ECM-ed Map-Register and corresponding Data-Map-Notify is used by the xTR behind the NAT to send and receive LISP control packets to/from the RTR, as well as for receiving LISP data packets form the RTR.

If ETR receives a Data-Map-Notify with a xTR-ID specified, but the xTR-ID is not equal to its local xTR-ID, it must log this as an error. The ETR should discard such Data-Map-Notify message.

The ETR must periodically send ECM-ed Map-Register messages to its RTR in order to both refresh its registration to the RTR and the Map-Server, and as a keep alive in order to preserve the state in the NAT device. RFC 2663 [NAT] points out that the period for sending the keep alives can be set to default value of two minutes, however since shorter timeouts may exist in some NAT deployments, the interval for sending periodic ECM-ed Map-Registers must be configurable.

5.1.2. Map-Request and Map-Reply Handling

To avoid the creation of extra status in the NAT devices due to the Map-Replies send to requesting ITRs, the ETR set the proxy-bit of the Map-Register to one in order the Map-Server proxy-reply the ETR as described in [I-D.ietf-lisp-rfc6833bis]

When an ITR behind a NAT is encapsulating outbound LISP traffic, it can use its RTR RLOC as the locator for all destination EIDs that it wishes to send data to. As such, the ITR does not need to send Map-Requests for finding EID-to-RLOC mappings. However, if the ITR is multihomed and has at least one interface not behind NAT, it can choose to send Map-Requests. For this, the ITR specifies in the ITR-RLOC field of the Map-Request the list of RLOCs that are not behind NAT to receive the Map-Reply messages. It should be noted that sending packets directly to destination RLOCs through the interface behind NAT will result in creating additional state in the NAT device. Also, it should be noted that outgoing packets use a direct path while the incoming packets are forwarded through an RTR.

For RLOC-probing, the periodic ECM-ed Map-Register and Data-Map-Notify messages between xTR and RTR can also serve the purpose of RLOC probes. However, if RLOC-probing is used, no changes are required to the RLOC-probing specification in [I-D.ietf-lisp-rfc6833bis], except that the LISP device behind a NAT only needs to probe the RTR's RLOC.

5.1.3. xTR Sending and Receiving Data

When a Map-Request for a LISP device behind a NAT is received by its Map-Server, the Map-Server responds with a Map-Reply including RTR's RLOC as the locator for the requested EID. As a result, all LISP data traffic destined for the ETR's EID behind the NAT is encapsulated to its RTR. The RTR re-encapsulates the LISP data packets to the ETR's translated global RLOC and port number so the data can pass through the NAT device and reach the ETR. As a result the ETR receives LISP data traffic with outer header destination port set to 4341 as specified in [I-D.ietf-lisp-rfc6830bis].

For sending outbound LISP data, an ITR behind a NAT SHOULD use the RTR RLOC as the locator for all EIDs that it wishes to send data to via the interface behind the NAT. The ITR then encapsulates the LISP traffic in a LISP data header with outer header destination set to RTR RLOC and outer header destination port set to 4341. This may create a secondary state in the NAT device. ITR SHOULD set the outer header source port in all egress LISP data packets to a random but static port number in order to avoid creating excessive state in the NAT device.

If the ITR and ETR of a site are not collocated, the RTR RLOC must be configured in the ITR via an out-of-band mechanism. Other procedures specified here would still apply.

5.2. Map-Server Processing

Upon receiving an Info-Request message a Map-Server first verifies the authenticity of the message. Next the Map-Server creates an Info-Reply message and copies the source RLOC and port number of the Info-Request message to the Global ETR RLOC Address and ETR UDP Port Number fields of the Info-Reply message. The Map-Server also includes a list of RTR RLOCs that the ETR may use for NAT traversal services. The Map-Server sends the Info-Reply message to the ETR, by setting the destination RLOC and port of the Info-Reply to the source

RLOC and port of the triggering Info-Request. The Map-Server sets the source port of the Info-Reply to 4342.

Upon receiving an ECM-ed Map-Register message with the N bit in the ECM header set to 1, the Map-Server removes the ECM header and if the M bit in the Map-Register is set, the Map-Server processes the Map-Register message and generates the resulting Map-Notify as described in [I-D.ietf-lisp-rfc6833bis]. The Map-Server encapsulates the Map-Notify in an ECM header and sets the R bit in the ECM header to 1. This indicates that the ECM-ed Map-Notify is to be processed by an RTR. If the Map-Server has a shared secret configured with the RTR sending the Map-Register, the Map-Server also sets the S bit in the ECM header of the Map-Notify and includes the MS-RTR authentication data after the ECM LISP header. See Security Considerations Section for more details. If the I bit is set in the Map-Register message, the Map-Server also locally stores the xTR-ID and site-ID from the Map-Register, and sets the I bit in the corresponding Map-Notify message and includes the same xTR-ID and site-ID in the Map-Notify. The ECM-ed Map-Notify is then sent to the RTR sending the corresponding Map-Register.

5.3. RTR Processing

Upon receiving an ECM-encapsulated Map-Register with the R bit set in the ECM header, the RTR creates a map-cache entry for the EID-prefix that was specified in the Map-Register message. The RTR stores the outer header source RLOC and outer header source port, the outer header destination RLOC (RTR's own RLOC), the inner header source RLOC (xTR's local RLOC), the xTR-ID, the weight and priority associated with the xTR's local RLOC that was used to send this Map-Register if present, and the nonce field of the Map-Register in this local map-cache entry. The RTR uses the inner header source address to identify which xTR local RLOC (R bit =0) was used by the xTR to send this Map-Register. The outer header source RLOC and outer header source port is the ETR's translated global RLOC and port number visible to the RTR. Once the registration process is complete, this map-cache entry can be used to send LISP data traffic to the ETR. The outer header destination RLOC is the RTR's RLOC used by the ETR. The RTR can later use these fields as the source RLOC for sending data-encapsulated control messages (Data-Map-Notify) back to the ETR. The nonce field is used for security purposes and is matched with the nonce field in the corresponding Map-Notify message. This map-cache entry is stored as an "unverified" mapping, until the corresponding Map-Notify message is received.

In the cases where the xTR has multiple RLOCs behind the NAT, and requires the RTR to load balance the traffic across those interfaces, the xTR must include the local RLOCs associated with each interface behind the NAT with the R bit in the locator record set to 0 in the ECM-ed Map-Register sent to the RTR. The RTR uses the weight and priority policies of the RLOCs with R=0 in the Map-Register to load balance the traffic from the RTR to the xTR behind the NAT. The RTR compares the RLOCs with the R bit set to 0 in the Map-Register to the inner header source address of the Map-Register to find the matching RLOC that the xTR used to send the Map-Register from. The RTR associates the weight and priority policies of this local RLOC with the NAT-translated RLOC and xTR-ID for this map-cache entry. For all other local RLOCs included in the Map-Register, that the Map-Register is not originating from, the RTR only updates previously cached weight and priority policies if it already has those local RLOCs previously stored for that EID prefix and xTR-ID. In other words, the RTR only adds new local RLOCs and their weight and priority policies to its cache if the Map-Register is actually originating from that RLOC. The TTL for every map-cache is also only updated when a Map-Register is originating from the same RLOC. However, the weight and priorities of all previously cached local RLOCs will be updated by every Map-Register, whether it is originating from that RLOC or not. The xTR-ID is used to define the Merge domain for these RLOCs. In other words, a Map-Register originating from a unique xTR-ID will always overwrite previously stored policies for that xTR-ID. However it does not modify in any way the policies indicated by any other xTR-ID serving the same EID prefix. As a result, in the case of a renumbering or xTR reboot, the xTR uses its unique xTR-ID to send a new Map-Register, overwriting the previously stored policies for that xTR. Using this method the xTR can immediately remove any RLOCs from the RTR cache that are no longer active. In order to implement this, the RTR must compare the list of local RLOCs in the Map-Register (R=0) with the ones it has previously cached associated with the same xTR-ID. If there is any RLOC previously cached that does not appear in the newly received Map-Register, the RTR must remove that RLOC together with the associated translated RLOC and associated policies, because removal of a local (behind-the-NAT) RLOC also invalidates the NAT-ed address associated with it. .

After filling the local map-cache entry, the RTR strips the outer header and extracts the Map-Register message, encapsulated in a new ECM header with the R bit set to 0, and N bit set to 1, and sends the ECM-ed Map-Register to destination Map-Server.

Map-Server responds with a ECM-ed Map-Notify message to the RTR.

Upon receiving an ECM-ed Map-Notify message with R bit set to 1 in the ECM header, if the S bit in ECM header is set to 1, RTR uses the MS-RTR Key ID to verify the MS-RTR Authentication Data included after the ECM header. If the MS-RTR authentication fails, the RTR must drop the packet. Once the authenticity of the message is verified, RTR can confirm that the Map-Register message for the ETR with the matching xTR-ID was accepted by the Map-Server. At this point the RTR can change the state of the associated map-cache entry to verified for the duration of the Map-Register TTL.

The RTR then uses the information in the associated map-cache entry to create a Data-Map-Notify message according to the following procedure: The RTR encapsulates the Map-Notify in a LISP data header, where the outer header destination RLOC and port number are set to the ETR's translated global RLOC and port number. If more than one ETR translated RLOC and port exists in the map-cache entry for the same EID prefix specified in the Map-Notify, the RTR can use the xTR-ID from the Map-Notify to identify which ETR is the correct destination for the Data-Map-Notify. The RTR sets the outer header source RLOC to RTR's RLOC from the map-cache entry and the outer header source port is set to 4342. The RTR also sets the Instance ID field in the LISP header of the Data-Map-Notify to 0xFFFFFF. The RTR then sends the Data-Map-Notify to the ETR.

If the S bit is set to 0 in the ECM header of the Map-Notify, and the RTR has a shared key configured locally with the sending Map-Server, the RTR must drop the packet. If the S bit is set to 0, and the RTR does not have a shared key configured with the associated Map-Server, according to local policy, the RTR may drop the packet. If the Map-Notify with S bit set to 0 is processed, the RTR must match the nonce field from this Map-Notify with the nonce stored in the local map-cache entry with the matching xTR-ID. If the nonces do not match, the RTR must drop the packet.

An RTR receiving an unsolicited Map-Notify for a registered EID should check if any of the RTR's RLOCs is present in the received record. If it does, the RTR rewrites the inner header destination RLOC of the Map-Notify message to ETR's local RLOC obtained from the associated map-cache entry of the EID. Then the RTR encapsulates and forwards the Map-Notify in a LISP data packet as explained above. If the record of the received Map-Notify doesn't contain any locator of the RTR it should drop it and request the new mapping to the mapping system. If the map reply record doesn't contain any RTR's locator, the map-cache entry for the EID is replaced otherwise, the Map-Reply is silently dropped.

5.3.1. RTR Data Forwarding

For all LISP data packets encapsulated to RTR's RLOC and outer header destination port 4341, the RTR first verifies whether the source or destination EID is a previously registered EID. If so, the RTR must process the packet according to the following. If the destination or source EID is not a registered EID, the RTR can drop or process the packets based on local policy.

In the case where the destination EID is a previously registered EID, the RTR must strip the LISP data header and re-encapsulate the packet in a new LISP data header. The outer header RLOCs and UDP ports are then filled based on the matching map-cache entry for the associated destination EID prefix. The RTR uses the RTR RLOC from the map-cache entry as the outer header source RLOC. The outer header source port is set to 4342. The RTR sets the outer header destination RLOC and outer header destination port based on the ETR translated global RLOC and port stored in the map-cache entry. Then the RTR forwards the LISP data packet.

In the case where the source EID is a previously registered EID, the RTR process the packet as if it is a Proxy ETR (PETR). The RTR must strip the LISP data header, and process the packet based on its inner header destination address. The packet may be forwarded natively, it may be LISP encapsulated to the destination ETR, or it may trigger the RTR to send a LISP Map-Request.

5.4. Multi-homed xTRs

In the case where an xTR has multiple interfaces and RLOCs, info-Requests can be sent per each interface and NAT discovery is done per each interface. NAT traversal is accomplished by following state and processes described above per each interface/RLOC. In other words, if multiple interfaces of an xTR are behind a NAT, the ECM-ed Map-Register messages should be sent via each xTR interface behind NAT if the xTR desires to receive traffic via that interface. This is required to establish the state in the NAT device for that interface. The M bit (want Map-Notify) must be set in ECM-ed Map-Register messages sent from at least one of xTR interfaces behind the NAT. If additional interfaces behind the NAT are using the same RTR for NAT traversal, no Map-Notify processing is required for such interfaces and M bit in Map-Register can be set to 0 for these to reduce processing on the RTR and the Map-Server.

The RLOCs included in Map-Register messages when the xTR has multiple interfaces SHOULD be the union of the locators (behind NAT or not) resulting from the process defined above per each RLOC of the xTR, according to the specifics of that interface (whether it is behind the NAT or not).

In cases where some xTR interfaces are behind NAT while others are not, ECM-ed Map-Register messages should be sent via interfaces behind the NAT through the selected RTRs. xTR can receive traffic via both types of interfaces by including the associated RLOCs (as well as the RTR RLOCs) in its ECM-ed Map-Register messages. The xTR can choose to store RTR RLOCs in a default map-cache entry to forward all the traffic through the RTR. If the xTR decides to populate its map-cache, the xTR may configure the RTR as a proxy of the interface behind NAT instead of sending the traffic directly to avoid generate new state in the NAT device.

5.5. Updating contents of EID-to-RLOC Mappings

When an interface of a LISP device is configured with a new RLOC, it needs to discover whether it is behind NAT by sending an Info-Request message. If the modified interface was previously behind NAT and the new RLOC is also behind NAT using the same RTRs association, the cache of the RTR is updated through the ECM-ed Map-Register send by the ETR. Otherwise, the change of the EID-to-RLOC mapping needs to be notified to remote ITRs/PiTRs.

LISP defines several mechanisms to update mappings in the Data-Plane [I-D.ietf-lisp-rfc6830bis] and in the control plane [I-D.ietf-lisp-rfc6833bis]. Also an additional Control-Plane mechanism based on the Publish/subscribe paradigm is specified in [I-D.ietf-lisp-pubsub]. It is recommended to use [I-D.ietf-lisp-pubsub] to notify mapping changes as it is a faster mechanism and it avoids the temporal increase of the path length associated with an ETR desassociating of an RTR.

In the case of Solicit-Map-Request mechanism, when an RTR receives a Soliciting Map-Request for a cached EID-Prefix, the RTR should request for the new mapping to the mapping system. If the Map-Reply record contains any RTR's locator, the RTR silently discards the Map-Reply; otherwise, the status of the map-cache entry used to punch the NAT of the EID-Prefix would be lost. If the Map-Reply record doesn't contain any RTR's locator, the RTR must update the cache entry.

A Map-Server receiving a new registration for an EID should send unsolicited Map-Notify to the departure ETRs as specified in section 4.2.3 of [I-D.ietf-lisp-eid-mobility]. An RTR receiving an unsolicited Map-Notify for a registered EID should process it as described in Section 5.3.

An RTR that replace a registered cache entry through a Map-Request / Map-Reply must forward traffic to the EID as described in [I-D.ietf-lisp-rfc6833bis] until the map-cache entry expires. Optionally, the RTR can send Data Driven SMRs to ITRs sending traffic to the removed EID through the RTR as described in section 4.2.4 of [I-D.ietf-lisp-eid-mobility].

5.6. Example

What follows is an example of an ETR initiating a registration of a new RLOC to its Map-Server, when there is a NAT device on the path between the ETR and the Map-Server.

In this example, the ETR (site1-ETR) is configured with the local RLOC of 192.168.1.2. The NAT's global (external) addresses are from 2.0.0.1/24 prefix. The Map-Server is at 3.0.0.1. And one potential RTR has an IP address of 1.0.0.1. The site1-ETR has an EID Prefix of 128.1.0.0/16.

An example of the registration process follows:

1. The Site1-ETR receives the private IP address, 192.168.1.2 as its RLOC via DHCP.

2. The Site1-ETR sends an Info-Request message with the destination RLOC of the Map-Server, 3.0.0.1, and source RLOC of 192.168.1.2. This packet has the destination port set to 4342 and the source port is set to (for example) 5001.

3. The NAT device translates the source IP from 192.168.1.2 to 2.0.0.1, and source port to (for example) 20001 global ephemeral source port.

4. The Map-Server receives and responds to this Info-Request with an Info-Reply message. This Info-Reply has the destination address set to ETR's translated address of 2.0.0.1 and the source address is the Map-Server's RLOC, namely 3.0.0.1. The destination port is 20001 and the source port is 4342. Map-Server includes a copy of the source address and port of the Info-Request message (2.0.0.1:20001), and a list of RTR RLOCs including RTR RLOC 1.0.0.1 in the Info-Reply contents.

5. The NAT translates the Info-Reply packet's destination IP from 2.0.0.1 to 192.168.1.2, and translates the destination port from 20001 to 5001, and forwards the Info-Reply to site1-ETR at 192.168.1.2.

6. The Site1-ETR detects that it is behind a NAT by comparing its local RLOC (192.168.1.2) with the Global ETR RLOC Address in the Info-Reply (2.0.0.2) . Then site1-ETR picks the RTR 1.0.0.1 from the list of RTR RLOCs in the Info-Reply. ETR stores the RTR RLOC in a default map-cache entry to periodically send ECM-ed Map-Registers to.

7. The ETR sends an ECM encapsulated Map-Register to RTR at 1.0.0.1. The outer header source RLOC of this Map-Register is set to 192.168.1.2 and the outer header source port is set to 4341. The outer header destination RLOC and port are set to RTR RLOC at 1.0.0.1 and 4342 respectively. The R bit in ECM header is set to 1. The inner header destination RLOC is set to ETR's Map-Server 3.0.0.1, and the inner header destination port is set to 4342. The inner header source RLOC is set to ETR's local RLOC 192.168.1.2 and the source port is set to (for example) 5002. In the Map-Register message the RTR RLOC 1.0.0.1 appears as the locator set for the ETR's EID prefix (128.1.0.0/16). In this example ETR also sets the Proxy bit in the Map-Register to 1, and sets I bit to 1, and includes its xTR-ID and site-ID in the Map-Register.

8. The NAT translates the source RLOC in the ECM header of the Map-Register, by changing it from 192.168.1.2 to 2.0.0.2, and translates the source port in the ECM header from 4341 to (for example) 20002, and forwards the Map-Register to RTR.

9. The RTR receives the Map-Register and creates a map-cache entry with the ETR's xTR-ID, EID prefix, and the source RLOC and port of the ECM header of the Map-Register as the locator (128.1.0.0/16 is mapped to 2.0.0.2:20002). RTR also caches the inner header source RLOC of the Map-Register namely 192.168.1.2, and the outer header destination RLOC of the ECM header in the Map-Register (this would be RTR's RLOC 1.0.0.1) to use for sending back a Data-Map-Notify. RTR then removes the outer header, adds a new ECM header with R=0, and N=1, and forwards the Map-Register to the destination Map-Server.

10. The Map-Server receives the ECM-ed Map-Register with N bit set to 1, removes the ECM header, and processes it according to [I-D.ietf-lisp-rfc6833bis]. Since Map-Server has a shared secret with the sending RTR, after registering the ETR, Map-Server responds with a ECM-ed Map-Notify with the R bit and S bit both set to 1 in the ECM header and including the MS-RTR authentication data. Since the I bit is set in the Map-Register, the Map-Server also sets the I bit in the Map-Notify and copies the xTR-ID and site-ID from the Map-Register to the Map-Notify. The source address of this Map-Notify is set to 3.0.0.1. The destination is copied from the local source address of the Map Register (192.168.1.2), and both source and destination ports are set to 4342.

11. The RTR receives the ECM-ed Map-Notify and verifies the MS-RTR authentication data. The RTR data-encapsulates the Map-Notify and sends the resulting Data-Map-Notify to site1-ETR with a

matching xTR-ID. The outer header source RLOC and port of the Data-Map-Notify are set to 1.0.0.1:4342. The outer header destination RLOC and port are retrieved from previously cached map-cache entry in step 9 namely 2.0.0.2:20002. RTR sets the Instance ID in the LISP header to 0xFFFFFF. At this point RTR marks ETR's EID prefix as "Registered" status and forwards the Data-Map-Notify to ETR.

12. The NAT device translates the destination RLOC and port of the Data-Map-Notify to 192.168.1.2:4341 and forwards the packet to ETR.

13. The Site1-ETR receives the packet with a destination port 4341, and processes the packet as a control packet after observing the Instance ID value 0xFFFFFF in the LISP header. At this point ETR's registration to the RTR is complete.

Assume a requesting ITR in a second LISP (site2-ITR) site has an RLOC of 74.0.0.1. The following is an example process of an EID behind site2-ITR sending a data packet to an EID behind the site1-ETR:

1. The ITR sends a Map-Request which arrives via the LISP mapping system to the ETR's Map Server.

2. The Map-Server sends a Map-Reply on behalf of the ETR, using the RTR's RLOC (1.0.0.1) in the Map-Reply's Locator Set.

3. The ITR encapsulates a LISP data packet with ITR's local RLOC (74.0.0.1) as the source RLOC and the RTR as the destination RLOC (1.0.0.1) in the outer header.

4. The RTR decapsulates the packet, evaluates the inner header against its map-cache and then re-encapsulates the packet. The new outer header's source RLOC is the RTR's RLOC 1.0.0.1 and the new outer header's destination RLOC is the Global NAT address 2.0.0.2. The destination port of the packet is set to 20002 (discovered above during the registration phase) and the source port is 4342.

5. The NAT translates the LISP data packet's destination IP from to 2.0.0.2 to 192.168.1.2, and translates the destination port from 20002 to 4341, and forwards the LISP data packet to the ETR at 192.168.1.2.

6. For the reverse path the ITR uses its local map-cache entry with the RTR RLOC as the default locator and encapsulates the LISP data packets using RTR RLOC, and 4341 as destination RLOC and port. The ITR must pick a random source port to use for all

outbound LISP data traffic in order to avoid creating excessive state in the NAT.

6. Security Considerations

By having the RTR relay the ECM-ed Map-Register message from an ETR to its Map-Server, the RTR can restrict access to the RTR services, only to those ETRs that are registered with a given Map-Server. To do so, the RTR and the Map-Server may be configured with a shared key that is used to authenticate the origin and to protect the integrity of the Map-Notify messages sent by the Map Server to the RTR. This prevents an on-path attacker from impersonating the Map-Server to the RTR, and allows the RTR to cryptographically verify that the ETR is properly registered with the Map-Server.

Having the RTR re-encapsulate traffic only when the source or the destination are registered EIDs, protects against the adverse use of an RTR for EID spoofing.

Upon receiving a Data-Map-Notify, an xTR can authenticate the origin of the Map-Notify message using the key that the ETR shares with the Map-Server. This enables the ETR to verify that the ECM-ed Map-Register was indeed forwarded by the RTR to the Map-Server, and was accepted by the Map-Server.

6.1. Acknowledgments

The authors would like to thank Noel Chiappa, Alberto Rodriguez Natal, Lorand Jakab, Albert Cabellos, Dominik Klein, Matthias Hartmann, and Michael Menth for their previous work, feedback and helpful suggestions.

7. IANA Considerations

This document does not request any IANA actions.

8. Normative References

[I-D.ietf-lisp-eid-mobility]
Portoles-Comeras, M., Ashtaputre, V., Moreno, V., Maino, F., and D. Farinacci, "LISP L2/L3 EID Mobility Using a Unified Control Plane", draft-ietf-lisp-eid-mobility-06 (work in progress), May 2020.

[I-D.ietf-lisp-pubsub]
Rodriguez-Natal, A., Ermagan, V., Cabellos-Aparicio, A., Barkai, S., and M. Boucadair, "Publish/Subscribe Functionality for LISP", draft-ietf-lisp-pubsub-06 (work in progress), July 2020.

[I-D.ietf-lisp-rfc6830bis]
Farinacci, D., Fuller, V., Meyer, D., Lewis, D., and A. Cabellos-Aparicio, "The Locator/ID Separation Protocol (LISP)", draft-ietf-lisp-rfc6830bis-35 (work in progress), September 2020.

[I-D.ietf-lisp-rfc6833bis]
Farinacci, D., Maino, F., Fuller, V., and A. Cabellos-Aparicio, "Locator/ID Separation Protocol (LISP) Control-Plane", draft-ietf-lisp-rfc6833bis-29 (work in progress), September 2020.

[ICE] Rosenberg, J., "Interactive Connectivity Establishment (ICE)", RFC rfc5245, October 2008.

[LCAF] Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format (LCAF)", RFC 8060, December 2015.

[NAT] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

[NAT-MN] Klein, D., Hartmann, M., and M. Menth, "NAT traversal for LISP mobile node, In Proceedings of the Re-Architecting the Internet Workshop (ReARCH ’10).", 2010.

[STUN] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC rfc5389, October 2008.

[TURN] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN)", RFC rfc5766, April 2010.

Authors’ Addresses

Vina Ermagan
Google

Email: ermagan@gmail.com

Dino Farinacci
lispers.net

Email: farinacci@gmail.com

Darrel Lewis
Cisco Systems, Inc.

Email: darlewis@cisco.com

Fabio Maino
Cisco Systems, Inc.

Email: fmaino@cisco.com

Marc Portoles Comeras
Cisco Systems, Inc.

Email: mportole@cisco.com

Jesper Skriver
Arista

Email: jesper@skriver.dk

Chris White
Logicalelegance, Inc.

Email: chris@logicalelegance.com

Albert Lopez
UPC/BarcelonaTech

Email: alopez@ac.upc.edu

Albert Cabellos
UPC/BarcelonaTech

Email: acabello@ac.upc.edu

Network Working Group D. Farinacci
Internet-Draft lispers.net
Intended status: Experimental November 14, 2020
Expires: May 18, 2021

LISP Distinguished Name Encoding
draft-farinacci-lisp-name-encoding-11

Abstract

This draft defines how to use the AFI=17 Distinguished Names in LISP.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The LISP architecture and protocols [RFC6830] introduces two new numbering spaces, Endpoint Identifiers (EIDs) and Routing Locators (RLOCs) which are intended to replace most use of IP addresses on the Internet. To provide flexibility for current and future applications, these values can be encoded in LISP control messages using a general syntax that includes Address Family Identifier (AFI) [RFC1700].

The length of the value field is implicit in the type of address that follows. For AFI 17, a Distinguished Name can be encoded. A name can be a variable length field so the length cannot be determined solely from the AFI value 17. This draft defines a termination character, an 8-bit value of 0 to be used as a string terminator so the length can be determined.

LISP Distinguished Names are useful when encoded either in EID-records or RLOC-records in LISP control messages. As EIDs, they can be registered in the mapping system to find resources, services, or simply used as a self-documenting feature that accompany other address specific EIDs. As RLOCs, Distinguished Names, along with RLOC specific addresses and parameters, can be used as labels to identify equipment type, location, or any self-documenting string a registering device desires to convey.

2. Definition of Terms

Address Family Identifier (AFI): a term used to describe an address encoding in a packet. An address family currently defined for IPv4 or IPv6 addresses. See [AFI] and [RFC1700] for details on other types of information that can be AFI encoded.

3. Distinguished Name Format

An AFI=17 Distinguished Name is encoded as:

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |            AFI = 17           |       ASCII String ...        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                ...  ASCII String          |        0          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The string of characters are encoded in the ASCII character-set definition [RFC0020].

When Distinguished Names are encoded for EIDs, the EID-prefix length of the EIDs as they appear in EID-records for all LISP control messages is the length of the string in bits (include the null 0 byte). Where Distinguished Names are encoded anywhere else (i.e. nested in LCAF encodings), then any length field is the length of the ASCII string including the null 0 byte in units of bytes.

When Map-Requests are sent for an EID encoded in Distinguished Name format, an exact match request is performed. So the Map-Server (when configured for proxy-Map-Replying) or the ETR will return a Map-Reply with the same EID-prefix length.

4. Example Use-Cases

This section identifies three specific use-cases for the Distinguished Name format. Two are used for an EID encoding and one for a RLOC-record encoding. When storing public keys in the mapping system, as in [I-D.ietf-lisp-ecdsa-auth], a well known format for a public-key hash can be encoded as a Distinguished Name. When street location to GPS coordinate mappings exist in the mapping system, as in [I-D.farinacci-lisp-geo], the street location can be a free form ascii representation (with whitespace characters) encoded as a Distinguished Name. An RLOC that describes an xTR behind a NAT device can be identified by its router name, as in [I-D.farinacci-lisp-simple-nat], uses a Distinguished Name encoding. As well as identifying the router name (neither an EID or an RLOC) in NAT Info-Request messages uses Distinguished Name encodings.

5. Name Collision Considerations

When a Distinguished Name encoding is used to format an EID, the uniqueness and allocation concerns are no different than registering IPv4 or IPv6 EIDs to the mapping system. See [I-D.ietf-lisp-rfc6833bis] for more details. Also, the use-case documents specified in Section 4 provide allocation recommendations for their specific uses.

6. Security Considerations

There are no security considerations.

7. IANA Considerations

The code-point values in this specification are already allocated in [AFI].

8. References

8.1. Normative References

[AFI] "Address Family Identifier (AFIs)", ADDRESS FAMILY NUMBERS http://www.iana.org/numbers.html, Febuary 2007.

[I-D.ietf-lisp-rfc6833bis] Farinacci, D., Maino, F., Fuller, V., and A. Cabellos-Aparicio, "Locator/ID Separation Protocol (LISP) Control-Plane", draft-ietf-lisp-rfc6833bis-29 (work in progress), September 2020.

[RFC0020] Cerf, V., "ASCII format for network interchange", STD 80, RFC 20, DOI 10.17487/RFC0020, October 1969, <https://www.rfc-editor.org/info/rfc20>.

[RFC1700] Reynolds, J. and J. Postel, "Assigned Numbers", RFC 1700, DOI 10.17487/RFC1700, October 1994, <https://www.rfc-editor.org/info/rfc1700>.

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

8.2. Informative References

[I-D.farinacci-lisp-geo]
Farinacci, D., "LISP Geo-Coordinate Use-Cases", draft-farinacci-lisp-geo-10 (work in progress), October 2020.

[I-D.farinacci-lisp-simple-nat]
Farinacci, D., "A Simple LISP NAT-Traversal Implementation", draft-farinacci-lisp-simple-nat-01 (work in progress), November 2020.

[I-D.ietf-lisp-ecdsa-auth]
Farinacci, D. and E. Nordmark, "LISP Control-Plane ECDSA Authentication and Authorization", draft-ietf-lisp-ecdsa-auth-04 (work in progress), September 2020.

Appendix A. Acknowledgments

The author would like to thank the LISP WG for their review and acceptance of this draft.

Appendix B. Document Change Log

B.1. Changes to draft-farinacci-lisp-name-encoding-11

- Submitted November 2020.

- Made changes to reflect working group comments.

- Update references and document expiry timer.

B.2. Changes to draft-farinacci-lisp-name-encoding-10

- Submitted August 2020.

- Update references and document expiry timer.

B.3. Changes to draft-farinacci-lisp-name-encoding-09

- Submitted March 2020.

- Update references and document expiry timer.

B.4. Changes to draft-farinacci-lisp-name-encoding-08

- Submitted September 2019.

- Update references and document expiry timer.

B.5. Changes to draft-farinacci-lisp-name-encoding-07

- Submitted March 2019.

- Update referenes and document expiry timer.

B.6. Changes to draft-farinacci-lisp-name-encoding-06

- Submitted September 2018.

- Update document expiry timer.

B.7. Changes to draft-farinacci-lisp-name-encoding-05

- Submitted March 2018.

- Update document expiry timer.

B.8. Changes to draft-farinacci-lisp-name-encoding-04

- Submitted September 2017.

- Update document expiry timer.

B.9. Changes to draft-farinacci-lisp-name-encoding-03

- Submitted March 2017.

- Update document expiry timer.

B.10. Changes to draft-farinacci-lisp-name-encoding-02

- o Submitted October 2016.
- o Add a comment that the distinguished-name encoding is restricted to ASCII character encodings only.

B.11. Changes to draft-farinacci-lisp-name-encoding-01

- o Submitted October 2016.
- o Update document timer.

B.12. Changes to draft-farinacci-lisp-name-encoding-00

- o Initial draft submitted April 2016.

Author's Address

Dino Farinacci
lispers.net
San Jose, CA
USA

Email: farinacci@gmail.com

LISP Working Group
Internet-Draft
Intended status: Informational
Expires: April 17, 2020

S. Barkai
B. Fernandez-Ruiz
S. ZionB
R. Tamir
Nexar Inc.
A. Rodriguez-Natal
F. Maino
Cisco Systems
A. Cabellos-Aparicio
J. Paillissé Vilanova
Technical University of Catalonia
D. Farinacci
lispers.net
October 17,2020

Network-Hexagons: H3-LISP GeoState & Mobility Network
draft-ietf-lisp-nexagon-06

Abstract

This document specifies use of H3 and LISP to publish subscribe and reflect the real-time state and status of public spaces and public roads:
- Tile by tile, indexed annotation of streets & curbs in near real time
- Sharing hazards, blockages, parking, weather, maintenance, inventory..
- Between MobilityClients who produce and consume geo-state information
- Using geo-spatial IP channels of current state of the physical world

Table of Contents

1. Introduction

(1) The Locator/ID Separation Protocol (LISP) [RFC6830] splits current IP addresses in two different namespaces, Endpoint Identifiers (EIDs) and Routing Locators (RLOCs). LISP uses a map-and-encap approach that relies on (1) a Mapping System (distributed database) that stores and disseminates EID-RLOC mappings and on (2) LISP tunnel routers (xTRs) that encapsulate and decapsulate data packets based on the content of those mappings.

(2) H3 is a geospatial indexing system using a hexagonal grid that can be (approximately) subdivided into finer and finer hexagonal grids, combining the benefits of a hexagonal grid with hierarchical subdivisions. H3 supports sixteen resolutions. Each finer resolution has cells with one seventh the area of the coarser resolution. Hexagons cannot be perfectly subdivided into seven hexagons, so the finer cells are only approximately contained within a parent cell. Each cell is identified by a 64bit HID.

(3) The Berkeley Deep Drive (BDD) Industry Consortium investigates state-of-the-art technologies in computer vision and machine learning for automotive applications, and, for taxonomy of published automotive scene classification.

These standards are combined to create an in-network state which reflects the condition of each hexagonal tile (~1sqm) in every road. The lisp network maps & encapsulates traffic between MobilityClient endpoint identifiers (EID), and addressable tile-objects (HID=>EID). objects are aggregated by H3Service EIDs.

The H3-LISP mobility network bridges timing and location gaps between the production and consumption of information by MobilityClients:

- o vision, sensory, LIADR, AI applications -- information producers
- o driving-apps, map-apps, command & control -- information consumers

This is achieved by putting the physical world on a shared addressable state-grid typically at the edge. Tile by tile geo-state sharing using a brokered-network solves key issues in vehicle to vehicle information sharing. Challenges such as vision sensors when there are multiple perspectives, privacy and cyber when clients are directly communicating when they do not have to, and global geo pub-sub scenarios.

Given a situation observable by some end-points, it is unclear if the relevant end-points which need to know will receive consistent, conflicting, multiple, or no indications whatsoever. For example, when a vehicle experiences a sudden highway slow-down,"sees" many brake lights or "feels" accelerometer, there is no clear way for it to share this annotation with vehicles 20-30sec away to prevent a potential pile-up. Or, when a vehicle crosses an intersection, observing opposite-lane obstruction, construction, double-park, commercial loading, garbage truck, or stopped school-bus, there is no clear way for it to alert vehicles approaching that situation as it drives away.

Geo-state indirection also helps communicate advanced machine vision and/or radar annotations. These are constantly evolving technologies, and relaying road enumerations they produce, using peer-to-peer protocols, poses a significant interoperability challenge. It is hard to test each new annotation of any sensor or OEM vendor with any other driving application. Brokered IP multicast channels are themed, subscribing means interoperating.

These peer-to-peer limitations are inherent yet unnecessary, as in most road situations vehicles are not really proper peers. They just happen to be in the same place at the same time. H3-LISP mobility network solves the limitations of direct vehicle-to-vehicle communication by anchor brokers per geo-tile: timing, security, privacy, interoperability. Anchor brokering is achieved by MobilityClients communicating through in-network addressable geo-states. Addressable tiles are aggregated and maintained by LISP H3ServiceEIDs.

MobilityClients can provide drivers with heads-up alerts on hazards and obstacles beyond the line of sight of the driver and the in-car sensors: over traffic, around blocks, far-side junction, beyond road turns or surface curvatures. This highlights the importance of networks in providing road safety and the role networks play in future AV operation support systems (AV-OSS).

Beyond sharing use cases like finding freed-up curb-parking and help avoid construction zones, a mission critical use case for global geo-pub-sub has to do with supporting autonomous vehicle (AV) fleets.

Its now a consensus that as specific AV fleets start to role out, or regular cars with AV abilities become more perversive, that they need remote operations and remote take-over support. This means that for every M such cars there needs to be N human remote drivers ready to take over. These AV-OSS are put together by consortiums of multiple companies and are measured by their N/M.

Nexagon geo-channels role in this context is two fold:

1. flag tiles where the AV got confused because of current condition which is requiring remote intervention, so to steer other AVs away from this tile. This so not to pull-in more and more humans to intervene as more cars arrive.
2. fleets will heavily rely on platoons, convoys assembled on the road on the fly, driven at the head by a remote operator or human driver. AVs need to lock-on. Geo channels are used pub-sub these platoons as they drive by.

To summarize the H3-LISP solution outline:

(1) MicroPartition: 64bit indexed geo-spatial H3.r15 road-tiles
(2) EnumState: 64bit state values compile tile condition representation
(3) Aggregation: H3.r9 H3ServiceEID group individual H3.r15 road-tiles
(4) Channels: H3ServiceEIDs function as multicast state update channels
(5) Scale: H3ServiceEIDs distributed for in-network for latency-throughput
(6) Mapped Overlay: tunneled-network routes the mobility-network traffic
(7) Signal-free: tunneled overlay is used to map-register for mcast channels
(8) Aggregation: tunnels used between MobilityClients/H3ServiceEIDs <> edge
(9) Access: ClientXTRs/ServerXTRs tunnel traffic to-from the LISP EdgeRTRs
(10) Control: EdgeRTRs register-resolve H3ServiceEIDs and mcast subscription

```
|-0-|-1-|-2-|-3-|-4-|-5-|-6-|-7-|-8-|-9-|-A-|-B-|-C-|-D-|-E-|-F-|
|                         H3 Hexagon ID Key                      |
|-0-|-1-|-2-|-3-|-4-|-5-|-6-|-7-|-8-|-9-|-A-|-B-|-C-|-D-|-E-|-F-|
|                       H3 Hexagon State-Value                   |
```

```
  |---------------------------------------------------------------|


                          ___                                  ___
   H3ServiceEIDs   ___  /     \           H3ServiceEIDs ___  /     \
            ___  /     | H3.r9 |                 ___  /     | H3.r9 |
          /     | H3.r9 \ ___ /                /     | H3.r9 \ ___ /
         | H3.r9 \ ___ /  sXTR                | H3.r9 \ ___ /  sXTR
          \ ___ /  sXTR    |                   \ ___ /  sXTR      |
            sXTR    |      |                     sXTR     |       |
             |      |      |                      |       |       |
             |      |      |                      |       |       |
             + -  - + - - EdgeRTR           EdgeRTR - + - + - -   +
                             ||  (   (   ((  ||
                          (                        )
                        (      Network Hexagons      )
                      (            H3-LISP              )
                        (      Mobility Network       )
                          ((                        )
                            ||  ((   (())   ()  ||
                            ||                  ||
                = = = = = = =                     = = = = = = =
               ||                                             ||
            EdgeRTR                                        EdgeRTR
           ..    ..                                      ..      ..
          ..       ..                                  ..          ..
    ((((|))))    ((((|))))                         ((((|))))    ((((|))))
       /|\    RAN   /|\                               /|\    RAN   /|\
      ..                                                            ..
      ..                                                            ..
      ..          Road tiled by 1 sqm H3.r15 ID-Ed Geo-States       ..
      ..                                                            ..
      ..                        ___    ___    ___                   ..
      ..       .............  /     \/     \/     \ << cXTR::MobilityClientB
      ..       - - - - - - -  H3.r15  H3.r15 H3.r15 - - - - - - -
     MobilityClientA::cXTR >> \ ___ /\ ___ /\ ___ /..........
```

- MobilityClientA has seen MobilityClientB near future, and, vice versa
- Clients share information using addressable shared-state routed by LISP Edge
- ClientXTR (cXTR): tunnel encapsulation through access network to LISP Edge
- ServerXTR (sXTR): tunnel encapsulation through cloud network to LISP Edge
- The H3-LISP Mobility overlay starts in the cXTR and terminates in the sXTR
- The updates are routed to the appropriate tile geo-state by the LISP network
- EdgeRTRs perform multicast replication to edges and then native or to cXTRs
- Clients receive tile-by-tile geo-state updates via the multicast channels

Each H3.r9 hexagon is an EID Service with corresponding H3 hexagon ID. Bound to that service is a LISP xTR, called a ServerXTR, specified to deliver encapsulated packets to and from the H3ServiceEID and LISP Edge. EdgeRTRs are used to re-tunnel packets from MobilityClients to H3ServiceEIDs. Each H3ServiceEID is also a multicast source for updating MobilityClients on the state of the H3.r15 tiles aggregated by the H3ServiceEID.

2. 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 [RFC2119].

3. Definition of Terms

H3ServiceEID: Is an addressable aggregation of H3.r15 state-tiles. It is a designated source for physical world reported annotations, and an (s,g) source of multicast public-safety update channels. H3ServiceEID is itself an H3 hexagon, large enough to provide geo-spatial conditions context, but not too large as to over-burden (battery powered, cellular connected) subscribers with too much information. For Mobility Network it is H3.r9. It has a light-weight LISP protocol stack to tunnel packets aka ServerXTR. The EID is an IPv6 EID that contains the H3 64-bit address numbering scheme. See IANA consideration for details.

ServerXTR: Is a light-weight LISP protocol stack implementation that co-exists with H3ServiceEID process. When the server roams, the xTR roams with it. The ServerXTR encapsulates and decapsulates packets to/from EdgeRTRs.

MobilityClient: Is a roaming application that may be resident as part of an automobile, as part of a navigation application, part of municipal, state, of federal government command and control application, or part of live street view consumer type of application. It has a light-weight LISP protocol stack to tunnel packets aka ClientXTR.

MobilityClient EID: Is the IPv6 EID used by the Mobility Client applications to source packets. The destination of such packets are only H3ServiceEIDs. The EID format is opaque and is assigned as part of the MobilityClient network-as-a-service (NaaS) authorization.

ClientXTR: Is the light-weight LISP protocol stack implementation that is co-located with the Mobility Client application. It encapsulates packets sourced by applications to EdgeRTRs and decapsulates packets from EdgeRTRs.

EdgeRTR: Is the core scale and structure of the LISP mobility network. EdgeRTRs proxy H3ServiceEIDs and MobilityClient H3ServiceEID channel registration. EdgeRTRs aggregate MobilityClients and H3Services using tunnels to facilitate hosting-providers and mobile-hosting flexibility - for accessing the nexagon mobility network.
EdgeRTRs decapsulate packets from ClientXTRs, ServerXTRs and re-encapsulates packets to the clients and servers tunnels. EdgeRTRs glean H3ServiceEIDs and glean MobilityClient EIDs when it decapsulates packets. EdgeRTRs store H3ServiceEIDs and their own RLOC of where the H3ServiceEID is currently reachable from in the map-cache. These mappings are registered to the LISP mapping system so other EdgeRTRs know where to encapsulate for such EIDs. EdgeRTRs do not register MobilityClients' EIDs at the mapping service as these are temporary-renewed while using the mobility network. Enterprises may provide their own client facing EdgeRTRs to mask their clients geo-whereabouts while using the mobility network.

4. Deployment Assumptions

The specification described in this document makes the following deployment assumptions:

(1) Unique 64-bit HID is associated with each H3 geo-spatial tile
(2) MobilityClients and H3ServiceEIDs share this well known index
(3) 64-bit BDD state value is associated with each H3-indexed tile
(4) Tile state is compiled 16 fields of 4-bits, or max 16 enums

```
|-0-|-1-|-2-|-3-|-4-|-5-|-6-|-7-|-8-|-9-|-A-|-B-|-C-|-D-|-E-|-F-|
 0123012301230123012301230123012301230123012301230123012301230123
```

Subscription of MobilityClients to the mobility network is constantly renewed while on the move and is not intended as a means of basic connectivity. This is why MobilityClients use DNS/AAA to obtain temporary EIDs and EdgeRTRs and why they use (LISP) data-plane tunnels to communicate using their temporary EIDs with the dynamically assigned EdgeRTRs.

MobilityClient are otherwise unaware of the LISP network mechanism or mapping system and simply regard the data-plane tunnels as an application-specific virtual private network (VPN) that supports IPv6 EID addressable geo-state to publish (Ucast), Subscribe (Mcast) H3Services.

In order to get access to the MobilityVPN, MobilityClients first authenticate with the MobilityVPN AAA Server. DIAMETER based AAA is typically done at the provider edge (PE) by edge gateways. However, the typical case involves several types of customer equipment connected by wireline, or by wireless to a specific service provider. The Mobility VPN, on the other hand, potentially overlays a number of wireless networks and cloud-edge providers. It also involves dozens of Car-OEM, Driving-Applications, Smart-infrastructure vendors. This is why we require clients to first go through AAA in order to get both a MobilityClientEID and EdgeRTR gateway RLOC opened.

ClientXTR performs the following steps in order to use the mobility network:
1) obtain the address of the mobility network AAA server using DNS
2) obtain MobilityClientEID and EdgeRTR(s) from AAA server using DIAMETER
3) renew authorization from AAA while using the mobility network T1 minutes

```
MobilityClient   Domain Name Server    DIAMETER AAA        Mobility EdgeRTR
        |                    |                   |                   |
        | nslookup nexagon   |                   |                   |
        |------------------->|                   |                   |
        |<-------------------|                   |                   |
        |  Mobility AAA IP   |                   |                   |
        |                    |                   |                   |
        |  AAR(AVP:IMSI/User/Password/Toyota)    |                   |
        |--------------------------------------->|                   |
        |                    |                   | ACR(AVP ClientEID)|
        |                    |                   |------------------>|
        |                    |                   |<------------------|
        |                    |                   | ACA(AVP ClientEID)|
        |    AAA (Client::EID,EdgeRTR::RLOC)     |                   |
        |<---------------------------------------|                   |
        |                    |                   |                   |
        .                                                            .
        .                                                            .
        .                                                            .
        |   Publish IPv6 H3ServiceEID, Subscribe MLDv2 H3ServiceEID  |
                                  .
        |----------------------------------------------------------->|
        .                                                            .
        .                                                            .
        |<-----------------------------------------------------------|
        |       Signal freeing multicast Updates from H3ServiceEIDs  |
        .                                                            .
        .                                                            .
        .                                                            .
        |                    |                   |                   |
        |               AAR(Interim)             |                   |
        |--------------------------------------->|   ACR (Interim)   |
        |                    |                   |------------------>|
```

```
        |                    |                   |<------------------|
        |                    |                   |   ACA (Interim)   |
        |<-------------------------------------- |                   |
        |               AAA (Interim)            |                   |
```

Using this network login and re-login method we ensure that:
- the MobilityClientEIDs serve as credentials with the specific EdgeRTRs
- EdgeRTRs are provisioned to whitelist MobilityClient EIDs assigned to them
- EdgeRTRs are not tightly coupled to H3.r9 areas for privacy/load-balance
- Mobility Clients do not need to update EdgeRTRs while roaming in an area

The same EdgeRTR may serve several H3.r9 areas for smooth ride continuity, and, several EdgeRTRs may load balance an H3.r9 area with high density of originating MobilityClient rides. When a MobilityClient ClientXTR is homed to EdgeRTR, it is able to communicate with H3ServiceEIDs.

5. Mobility Clients Network Services

The mobility network functions as a standard LISP VPN overlay.
The overlay delivers unicast and multicast packets across:
- multiple access-networks and radio-access specifications
- multiple edie providers, public, private, and hybrid clouds

We use data-plane XTRs in the stack of each mobility client and server. ClientXTRs and ServerXTRs are homed to one or more EdgeRTRs at the LISP edge. This structure allows for MobilityClients to "show up" at any time, behind any network provider in a given mobility network administrative domain, and for any H3ServiceEID to be instantiated, moved, or failed-over to any rack in any cloud-provider. The LISP overlay enables these roaming mobility network elements to communicate uninterrupted. This quality is insured by the LISP RFCs. The determination of identities for MobilityClients to always refer to the correct H3ServiceEID is insured by H3 geo-spatial HIDs.

There are two options for how we associate ClientXTRs with LISP EdgeRTRs:

I. Semi-random load-balancing by DNS/AAA

In this option we assume that in a given metro edge, a pool of EdgeRTRs can distribute the Mobility Clients load randomly between them and that EdgeRTRs are topologically more or less equivalent. Each RTR uses LISP to tunnel traffic to and from other EdgeRTRs for MobilityClient with H3Service exchanges. MobilityClients can (multi) home to EdgeRTRs while moving.

II. Topological by anycast

In this option we align an EdgeRTR with topological aggregation like in Evolved Packet or 5GCore aggregation. Mobility Clients are roaming in an area home to that RTR and so is the H3 Server. There is only one hop across the edge overlay between clients and servers and mcast replication is more focused, but clients need to keep re-homing as they move.

To summarize the H3LISP mobility network layout:

(1) Mobility-Clients traffic is tunneled via data-plane ClientXTRs
    ClientXTRs are (multi) homed to EdgeRTR(s)
(2) H3ServiceEID traffic is tunneled via data-plane ServerXTR
    ServerXTRs are (multi) homed to EdgeRTR(s)
(3) EdgeRTRs use mapping service to resolve Ucast HIDs to RTR RLOCs
    EdgeRTRs also register to (Source, Group) H3ServiceEID multicasts

```
MobilityClients <> ClientXTR <Access Provider > EdgeRTR  v
                                                         v
v      << Map-Assisted Mobility-Network Overlay <<       v
v
>> EdgeRTR <Cloud Provider> ServerXTR <> H3ServiceEID
```

6. Mobility Unicast and Multicast

Regardless of the way a given ClientXTR was associated with an EdgeRTR, an authenticated MobilityClient EID can send: [64bitH3.15ID :: 64bitState] annotations to the H3.r9 H3ServiceEID. The H3.r9 EID can be calculated by clients algorithmically from the H3.15 localized annotation snapped-to-tile.

The ClientXTR encapsulates MobilityClient EID and H3ServiceEID in a packet sourced from the ClientXTR with the destination of the EdgeRTR RLOC LISP port. EdgeRTRs then re-encapsulate annotation packets either to remote EdgeRTR (option 1) or to homed H3ServiceEID ServerXTR (option 2).
The remote EdgeRTR aggregating H3ServiceEIDs re-encapsulates MobilityClient EID to the ServerXTR and from there to the H3ServiceEID.

```
    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  |   Hop Limit   |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |
|                                                               |  |
+                                                               +  |
|                                                               |  |
+                     Source MobilityClientEID                  +  |
|                                                               | IPv6
+                                                               +  |
|                                                               |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |
|                                                               |  |
+                                                               +  |
|                                                               |  |
+                        Dest H3ServiceEID                      +  |
|                                                               |  |
+                                                               +  |
|                                                               | /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Source Port = xxxx      |       Dest Port = xxxx        | \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ UDP
|           UDP Length          |        UDP Checksum           | /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ \
|  Type          |gzip |       Reserved          | Pair Count = X| Nexgon Header
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ /
|                                                               |
+                        64 Bit H3-R15 ID                       +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                        64 Bit State                           +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
|                                                               |
+                       64 Bit H3-R15 ID                        +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                       64 Bit State                            +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

To Summarize Unicast:

(1) MobilityClients can send annotations which are localized to an H3.r15 tile
    These annotations are sent to an H3.r9 mobility H3ServiceEIDs
(2) MobilityClient EID and H3ServiceEID HID are encapsulated:
      XTR <> RTR <> RTR <> XTR
    * RTRs can map-resolve re-tunnel HIDs
(3) RTRs re-encapsulate original source-dest to ServerXTRs
    ServerXTRs decapsulate packets to H3ServiceEID

Each H3.r9 Server is also an IP Multicast channel Source used to update subscribers on the aggregate state of the H3.r15 tiles in the H3.r9 Server. This forms a multipoint to multipoint state channel per H3 geo-location, where the H3 hairpin aggregation point has programable propagation functionality.

We use rfc8378 signal-free multicast to implement mcast channels in the overlay. The mobility network has many channels, with only a few thousands of subscribers per channel. MobilityClients driving through or subscribing to an H3.r9 area can explicitly issue an rfc4604 MLDv2 in order to subscribe, or, may be subscribed implicitly by the EdgeRTR gleaning to ucast HID destination.

The advantage of an explicit client MLDv2 registration as a trigger to rfc8378 is that the clients manage their own mobility mcast handover according to their location-direction vectors, and that it allows for otherwise silent, or, non annotating clients. The advantage of EdgeRTR implicit registration is less signaling required.

MLDv2 signaling messages are encapsulated between the ClientXTR and the LISP EdgeRTR, therefore there is no requirement for the underlying network to support native multicast. If native access multicast is supported (for example native 5G multicast), then MobilityClient registration to H3ServiceEID safety channels may be integrated with it, in which case the mobile packet-core (EPC) element supporting it (eNB) will use this standard to register with the appropriate H3.r9 channels in its area.

Multicast update packets are of the following structure:

```
  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  |   Hop Limit   |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |
|                                                               |  |
+                                                               +  |
|                                                               |  |
```

```
+                        Source H3-R9 EID Address                +  |
|                                                                | IPv6
+                                                                +  |
|                                                                |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |
|                                                                |  |
+                                                                +  |
|                                                                |  |
+                          Group Address                         +  |
|                                                                |  |
+                                                                +  |
|                                                                | /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Source Port = xxxx        |       Dest Port = xxxx       | \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ UDP
|           UDP Length            |        UDP Checksum          | /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ \
|                                                                | Nexagons Header
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ /
~                            Nexagons Payload                    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+


Outer headers = 40 (IPv6) + 8 (UDP) + 8 (LISP) = 56
Inner headers = 40 (IPv6) + 8 (UDP) + 4 (Nexagon Header) = 52

1500 (MTU) - 56 - 52 = 1392 bytes of effective payload

Type 1:key-value, key-value.. 1392 / (8 + 8) = 87 pairs
Type 2:value, key,key,key.. (1392 - 8) / 8 = 173 H3-R15 IDs


 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ \
|  Type =   1   |gzip |         Reserved         | Pair Count = X| Nexagon Header
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ /
|                                                                |
+                        64 Bit H3-R15 ID                        +
|                                                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                                |
+                        64 Bit State                            +
|                                                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                                |
+                        64 Bit H3-R15 ID                        +
|                                                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                                |
+                        64 Bit State                            +
|                                                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+



 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ \
|  Type =   2   |gzip |      Reserved            |H3R15 Count = X| Nexagon Header
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ /
```

```
|                                                               |
+                       64 Bit State                            +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                       64 Bit H3-R15 ID                        +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                       64 Bit H3-R15 ID                        +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                       64 Bit H3-R15 ID                        +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

' The remote EdgeRTRs homing MobilityClients in turn replicate the packet to the MobilityClients registered with them.

We expect an average of 600 H3.r15 tiles of the full 7^6 (~100K) possible in H3.r9 to be part of any road. The H3.r9 server can transmit the status of all 600 or just those with meaningful states based on updated SLA and policy.

To Summarize:

(1) H3LISP Clients tune to H3.r9 mobility updates using rfc8378
    H3LISP Client issue MLDv2 registration to H3.r9 HIDs
    ClientXTRs encapsulate MLDv2 to EdgeRTRs who register (s,g)

(2) ServerXTRs encapsulate updates to EdgeRTRs who map-resolve (s,g) RLOCs
    EdgeRTRs replicate mobility update and tunnel to registered EdgeRTRs
    Remote EdgeRTRs replicate updates to registered ClientXTRs

7. Security Considerations

The nexagon layer3 v2n network is inherently more secure and private then peer to peer alternatives because of the indirection. No car or infrastructure element communicates directly with MobilityClients. All information is conveyed using shared addressable geo-state. MobilityClients receive information only from network channels published by a trusted broker. MobilityClients have no indication as to the origin of the information. This is an important step towards better privacy, security, extendability, and interoperability compared with legacy layer2 protocols.

In order to be able to use the nexagon mobility network for a given period, the mobility clients go through a DNS/AAA stage by which they obtain their clientEID identifiers-credentials and the RLOCs of EdgeRTRs they may use as gateways to the network. This MobilityClient <> EdgeRTR interface is the most sensitive in this network to privacy and security considerations.

The traffic on the MobilityClient<>EdgeRTR interface is tunneled and its UDP content may be encrypted; still, the EdgeRTR will know based on the LISP headers alone the MobilityClient RLOC and H3-R9 (~0.1sqkm) geo-spatial area to which a given client publishes or subscribes to.

For this reason we envision the ability of enterprise or groups of users to "bring their own" EdgeRTRs. BYO-RTR masks individual clients' IP-RLOC to H3-R9 association and is pre-provisioned to be able to use the mapping system and be on a white-list of EdgeRTRs aggregating H3ServiceEIDs.

Beyond this sensitive hop, the mapping system does not hold MobilityClientEIDs, and remote EdgeRTRs are only aware of MobilityClient ephemeral EIDs, not their actual IP RLOC or any other mobile-device identifiers. EdgeRTRs register in the mapping (s,g) H3-R9 multicast groups, but which clients reside beyond which EdgeRTR is not in the mapping system, only the AAA server is aware of that. The H3ServiceEIDs themselves decrypt and parse actual H3-R15 annotations; they also consider during this the MobilityClientEID credentials to avoid "fake-news", but again these are only temporary EIDs allocated to clients in order to be able to use the mobility network and not for their actual IP.

In summary of main risk mitigations for the lisp-nexagon interface we can say:

(1) tapping: all communications are through dynamic tunnels therefore may be encrypted using IP-Sec or other supported point to point underlay standards. These are not static tunnels but lisp re-tunneling routers (RTRs) perform all nexagon Overlay aggregation.

(2) spoofing: it is very hard to guess a MobilityClientEID valid for a short period of time. Clients and H3Services EIDs are whitelisted in EdgeRTRs, Clients using the AAA procedure, H3Services via dev-ops.

(3) impersonating: efforts to use MobilityClients and H3Services RLOCs should be caught by the underlying service provider edge and access networks. EID impersonating is caught by EdgeRTR EID RLOC whitelist mismatch.

(4) credibility: the interface crowd-sources geo-state and does not assume to trust single detections. Credit history track to MobilityClientEIDs by as part of normal H3Services fact checking, aggregate scores affect AAA credentials.

(5) privacy: Only EdgeRTRs are aware of both clients' RLOC and geo-location, only AAA is aware of client IDs credentials and credit but not geo-location. aggregate credit score span all H3Services administratively without source.

8. Acknowledgments

This work is partly funded by the ANR LISP-Lab project #ANR-13-INFR-009 (https://lisplab.lip6.fr).

9. IANA Considerations

I. Formal H3 to IPv6 EID mapping

II. State enum fields of H3 tiles:

Field 0x: Traffic Direction {
0x - null
1x - Lane North
2x - Lane North + 30
3x - Lane North + 60
4x - Lane North + 90
5x - Lane North + 120
6x - Lane North + 150
7x - Lane North + 180
8x - Lane North + 210
9x - Lane North + 240

```
Ax - Lane North + 270
Bx - Lane North + 300
Cx - Lane North + 330
Dx - junction
Ex - shoulder
Fx - sidewalk
}

field 1x: Persistent or Structural {
 0x - null
 1x - pothole light
 2x - pothole severe
 3x - speed-bump low
 4x - speed-bump high
 5x - icy
 6x - flooded
 7x - snow-cover
 8x - snow-deep
 9x - construction cone
 Ax - gravel
 Bx - choppy
 Cx - blind-curve
 Dx - steep-slope
 Ex - low-bridge
}

field 2x: Transient Condition {
 0x - null
 1x - pedestrian
 2x - bike scooter
 3x - stopped car / truck
 4x - moving car / truck
 5x - first responder vehicle
 6x - sudden slowdown
 7x - oversized over-height vehicle
 8x - red-light-breach
 9x - light collision (fender bender)
 Ax - hard collision / casualty
 Bx - collision course car/structure
 Cx - recent collision residues
 Dx - hard brake
 Ex - sharp cornering
 Fx - freeing-parking
}

field 3x: Traffic-light Cycle {
 0x - null
 1x - 1 seconds to green
 2x - 2 seconds to green
 3x - 3 seconds to green
 4x - 4 seconds to green
 5x - 5 seconds to green
 6x - 6 seconds to green
 7x - 7 seconds to green
 8x - 8 seconds to green
 9x - 9 seconds to green
 Ax - 10 seconds or less
 Bx - 20 seconds or less
 Cx - 30 seconds or less
 Dx - 60 seconds or less
 Ex - green now
 Fx - red now
```

```
}

field 4x: Impacted Tile from Neighboring {
 0x – null
 1x – epicenter
 2x – light yellow
 3x – yellow
 4x – light orange
 5x – orange
 6x – light red
 7x – red
 8x – light blue
 9x – blue
 Ax – green
 Bx – light green
}

field 5x: Transient, Cycle, Impacted, Valid for Next{
 0x – null
 1x – 1sec
 2x – 5sec
 3x – 10sec
 4x – 20sec
 5x – 40sec
 6x – 60sec
 7x – 2min
 8x – 3min
 9x – 4min
 Ax – 5min
 Bx – 10min
 Cx – 15min
 Dx – 30min
 Ex – 60min
 Fx – 24hours
}

field 6x: LaneRightsSigns {
 0x – null
 1x – yield
 2x – speedLimit
 3x – straightOnly
 4x – noStraight
 5x – rightOnly
 6x – noRight
 7x – rightStraight
 8x – leftOnly
 9x – leftStraight
 Ax – noLeft
 Bx – noUTurn
 Cx – noLeftU
 Dx – bikeLane
 Ex – HOVLane
 Fx – Stop
}

field 7x: MovementSigns {
0x – null
1x – keepRight
2x – keepLeft
3x – stayInLane
4x – doNotEnter
5x – noTrucks
```

```
6x - noBikes
7x - noPeds
8x - oneWay
9x - parking
Ax - noParking
Bx - noStandaing
Cx - noPassing
Dx - loadingZone
Ex - railCross
Fx - schoolZone
}

field 8x: CurvesIntersectSigns {
0x - null
1x - turnsLeft
2x - turnsRight
3x - curvesLeft
4x - curvesRight
5x - reversesLeft
6x - reversesRight
7x - windingRoad
8x - hairPin
9x - pretzelTurn
Ax - crossRoads
Bx - crossT
Cx - crossY
Dx - circle
Ex - laneEnds
Fx - roadNarrows
}

field 9x: Current Tile Speed {
0x - null
1x - < 5kmh
2x - < 10kmh
3x - < 15kmh
4x - < 20kmh
5x - < 30kmh
6x - < 40kmh
7x - < 50kmh
8x - < 60kmh
9x - < 80kmh
Ax - < 100kmh
Bx - < 120kmh
Cx - < 140kmh
Dx - < 160kmh
Ex - > 160kmh
Fx - queuedTraffic
}

field Ax: Vehicle / Pedestrian Traffic {
0x - null
1x - probability of ped/vehicle on tile close to 100%
2x - 95%
3x - 90%
4x - 85%
5x - 80%
6x - 70%
7x - 60%
8x - 50%
9x - 40%
Ax - 30%
```

Bx - 20%
Cx - 15%
Dx - 10%
Ex - 5%
Fx - probability of ped/vehicle on tile close to 0%, empty
}

filed Bx - reserved
field Cx - reserved
field Dx - reserved
field Ex - reserved
field Fx - reserved

10. Normative References

[I-D.ietf-lisp-rfc6833bis]
Fuller, V., Farinacci, D., and A. Cabellos-Aparicio, "Locator/ID Separation Protocol (LISP) Control-Plane", draft-ietf-lisp-rfc6833bis-07 (work in progress), December 2017.

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

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

[RFC8378] Farinacci, D., Moreno, V., "Signal-Free Locator/ID Separation Protocol (LISP) Multicast", RFC8378, DOI 10.17487/RFC8378, May 2018, <https://www.rfc-editor.org/info/rfc8378>.

Authors' Addresses

Sharon Barkai
Nexar
CA
USA

Email: sbarkai@gmail.com

Bruno Fernandez-Ruiz
Nexar
London
UK

Email: b@getnexar.com

S ZionB
Nexar

Israel

Email: sharon@fermicloud.io

Rotem Tamir
Nexar
Israel

rotemtamir@getnexar.com

Alberto Rodriguez-Natal
Cisco Systems
170 Tasman Drive
San Jose, CA
USA

Email: natal@cisco.com

Fabio Maino
Cisco Systems
170 Tasman Drive
San Jose, CA
USA

Email: fmaino@cisco.com

Albert Cabellos-Aparicio
Technical University of Catalonia
Barcelona
Spain

Email: acabello@ac.upc.edu

Jordi Paillissé-Vilanova
Technical University of Catalonia
Barcelona
Spain

Email: jordip@ac.upc.edu

Dino Farinacci
lispers.net
San Jose, CA
USA

Email: farinacci@gmail.com

LISP Working Group A. Rodriguez-Natal
Internet-Draft Cisco
Intended status: Experimental V. Ermagan
Expires: January 11, 2021 Google
A. Cabellos
UPC/BarcelonaTech
S. Barkai
Nexar
M. Boucadair
Orange
July 10, 2020

Publish/Subscribe Functionality for LISP
draft-ietf-lisp-pubsub-06

Abstract

This document specifies an extension to the use of Map-Request to enable Publish/Subscribe (PubSub) operation for LISP.

Table of Contents

1. Introduction

The Locator/ID Separation Protocol (LISP) [I-D.ietf-lisp-rfc6833bis] splits current IP addresses in two different namespaces, Endpoint Identifiers (EIDs) and Routing Locators (RLOCs). LISP uses a map-and-encap approach that relies on (1) a Mapping System (basically a distributed database) that stores and disseminates EID-RLOC mappings and on (2) LISP tunnel routers (xTRs) that encapsulate and decapsulate data packets based on the content of those mappings.

Ingress Tunnel Routers (ITRs) / Re-encapsulating Tunnel Routers (RTRs) / Proxy Ingress Tunnel Routers (PITRs) pull EID-to-RLOC mapping information from the Mapping System by means of an explicit request message. Section 7.1 of [I-D.ietf-lisp-rfc6833bis] indicates how Egress Tunnel Routers (ETRs) can tell ITRs/RTRs/PITRs about mapping changes. This document presents a Publish/Subscribe (PubSub) extension in which the Mapping System can notify ITRs/RTRs/PITRs about mapping changes. When this mechanism is used, mapping changes can be notified faster and can be managed in the Mapping System versus the LISP sites.

In general, when an ITR/RTR/PITR wants to be notified for mapping changes for a given EID-prefix, the following steps occur:

(1) The ITR/RTR/PITR sends a Map-Request for that EID-prefix.

(2) The ITR/RTR/PITR sets the Notification-Requested bit (N-bit) on the Map-Request and includes its xTR-ID and Site-ID.

(3) The Map-Request is forwarded to one of the Map-Servers that the EID-prefix is registered to.

(4) The Map-Server creates subscription state for the ITR/RTR/PITR on the EID-prefix.

(5) The Map-Server sends a Map-Notify to the ITR/RTR/PITR to acknowledge the successful subscription.

(6) When there is an RLOC-set change for the EID-prefix, the Map-Server sends a Map-Notify message to each ITR/RTR/PITR in the subscription list.

(7) Each ITR/RTR/PITR sends a Map-Notify-Ack to acknowledge the received Map-Notify.

This operation is repeated for all EID-prefixes for which ITR/RTR/PITR want to be notified. The ITR/RTR/PITR can set the N-bit for several EID-prefixes within a single Map-Request.

2. Requirements Language

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Deployment Assumptions

The specification described in this document makes the following deployment assumptions:

(1) A unique 128-bit xTR-ID (plus a 64-bit Site-ID) identifier is assigned to each xTR.

(2) Map-Servers are configured in proxy-reply mode, i.e., they are solicited to generate and send Map-Reply messages for the mappings they are serving.

The distribution of xTR-IDs (and Site-IDs) are out of the scope of this document.

4. Map-Request PubSub Additions

Figure 1 shows the format of the updated Map-Request to support the PubSub functionality.

```
        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
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |Type=1 |A|M|P|S|p|s|R|I|  Rsvd    |L|D|   IRC   | Record Count  |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         Nonce . . .                           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         . . . Nonce                           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |         Source-EID-AFI        |   Source EID Address  ...     |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |         ITR-RLOC-AFI 1        |    ITR-RLOC Address 1  ...    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                              ...                              |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |         ITR-RLOC-AFI n        |    ITR-RLOC Address n  ...    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |N|   Reserved    | EID mask-len  |        EID-Prefix-AFI         |
   Rec +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     \ |                       EID-Prefix  ...                         |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                   Map-Reply Record  ...                       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                               |
       +                                                               +
       |                                                               |
       +                            xTR-ID                             +
       |                                                               |
       +                                                               +
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                               |
       +                           Site-ID                             +
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Map-Request with I-bit, N-bit, xTR-ID, and Site-ID

The following is added to the Map-Request message defined in Section 5.2 of [I-D.ietf-lisp-rfc6833bis]:

xTR-ID bit (I-bit): The I-bit of a Map-Request message is set to 1 to indicate that a 128 bit xTR-ID and a 64 bit Site-ID fields are present at the end of the Map-Request message. If an xTR is configured with an xTR-ID or Site-ID, it MUST set the I-bit to 1 and include its xTR-ID and Site-ID in the Map-Request messages it generates. If either the xTR-ID or Site-ID is not configured, an unspecified value is encoded for whichever ID that is not configured.

Notification-Requested bit (N-bit): The N-bit of an EID-record is set to 1 to specify that the xTR wants to be notified of updates for that mapping record.

xTR-ID field: xTR-ID is a 128 bit field at the end of the Map-Request message, starting after the final Record in the message (or the Map-Reply Record, if present). The xTR-ID is used to uniquely identify the sender of a Map-Request message. The xTR-ID is defined in Section 5.6 of [I-D.ietf-lisp-rfc6833bis]

Site-ID field: Site-ID is a 64 bit field at the end of the Map-Request message, following the xTR-ID. Site-ID is used by the Map-Server receiving the Map-Request message to identify which xTRs belong to the same site. The Site-ID is defined in Section 5.6 of [I-D.ietf-lisp-rfc6833bis]

5. Mapping Request Subscribe Procedures

The xTR subscribes for RLOC-set changes for a given EID-prefix by sending a Map-Request to the Mapping System with the N-bit set on the EID-Record. The xTR builds a Map-Request according to Section 5.3 of [I-D.ietf-lisp-rfc6833bis] but also does the following:

(1) The xTR MUST set the I-bit to 1 and append its xTR-ID and Site-ID to the Map-Request. The xTR-ID uniquely identifies the xTR.

(2) The xTR MUST set the N-bit to 1 for each EID-Record to which the xTR wants to subscribe.

The Map-Request is forwarded to the appropriate Map-Server through the Mapping System. This document does not assume that a Map-Server is pre-assigned to handle the subscription state for a given xTR. The Map-Server that receives the Map-Request will be the Map-Server responsible to notify that specific xTR about future mapping changes for the subscribed mapping records.

Upon receipt of the Map-Request, the Map-Server processes it as described in Section 8.3 of [I-D.ietf-lisp-rfc6833bis]. Furthermore, upon processing, for each EID-Record that has the N-bit set to 1, the Map-Server proceeds adding the xTR-ID contained in the Map-Request to the list of xTR that have requested to be subscribed to that mapping record.

If the xTR-ID is added to the list, the Map-Server MUST send a Map-Notify message back to the xTR to acknowledge the successful subscription. The Map-Server MUST follow the specification in Section 5.7 of [I-D.ietf-lisp-rfc6833bis] to build the Map-Notify with the following considerations:

(1) The Map-Server MUST use the nonce from the Map-Request as the nonce for the Map-Notify.

(2) The Map-Server MUST use its security association with the xTR (see Section 3) to compute the authentication data of the Map-Notify.

(3) The Map-Server MUST send the Map-Notify to one of the ITR-RLOCs received in the Map-Request.

When the xTR receives a Map-Notify with a nonce that matches one in the list of outstanding Map-Request messages sent with an N-bit set, it knows that the Map-Notify is to acknowledge a successful subscription. The xTR processes this Map-Notify as described in Section 5.7 of [I-D.ietf-lisp-rfc6833bis] with the following considerations. The xTR MUST use its security association with the Map-Server (see Section 7.1) to validate the authentication data on the Map-Notify. The xTR MUST use the Map-Notify to populate its map-cache with the returned EID-prefix and RLOC-set.

The subscription of an xTR-ID to the list of subscribers for the EID-Record may fail for a number of reasons. For example, because of local configuration policies (such as accept and drop lists of subscribers), or because the Map-Server has exhausted the resources to dedicate to the subscription of that EID-Record (e.g., the number of subscribers excess the capacity of the Map-Server).

If the subscription fails, the Map-Server MUST send a Map-Reply to the originator of the Map-Request, as described in Section 8.3 of [I-D.ietf-lisp-rfc6833bis]. The xTR processes the Map-Reply as specified in Section 8.1 of [I-D.ietf-lisp-rfc6833bis].

If an xTR-ID is successfully added to the list of subscribers for an EID-Record, the Map-Server MUST extract the nonce and ITR-RLOCs present in the Map-Request, and store the association between the EID-Record, xTR-ID, ITR-RLOCs and nonce. Any already present state regarding ITR-RLOCs and/or nonce for the same xTR-ID MUST be overwritten.

If the Map-Request only has one ITR-RLOC with AFI = 0 (i.e., Unknown Address), the Map-Server MUST remove the subscription state for that xTR-ID. In this case, the Map-Server MUST send the Map-Notify to the source RLOC of the Map-Request. When the TTL for the EID-record expires, the EID-prefix is removed from the Map-Server's subscription cache. On EID-Record removal, the Map-Server notifies the subscribers via a Map-Notify with TTL equal 0.

6. Mapping Notification Publish Procedures

The publish procedure is implemented via Map-Notify messages that the Map-Server sends to xTRs. The xTRs acknowledge the reception of Map-Notifies via sending Map-Notify-Ack messages back to the Map-Server. The complete mechanism works as follows.

When a mapping stored in a Map-Server is updated (e.g., via a Map-Register from an ETR), the Map-Server MUST notify the subscribers of that mapping via sending Map-Notify messages with the most updated mapping information. The Map-Notify message sent to each of the subscribers as a result of an update event MUST follow the exact encoding and logic defined in Section 5.7 of [I-D.ietf-lisp-rfc6833bis] for Map-Notify, except for the following:

(1) The Map-Notify MUST be sent to one of the ITR-RLOCs associated with the xTR-ID of the subscriber.

(2) The Map-Server increments the nonce every time it sends a Map-Notify as publication to an xTR-ID for a particular EID-Record. The starting nonce is set as follows, if the subscription state at the Map-Server was created by a received Map-Request with the N-bit set, the starting nonce in the Map-Notify sent as publication MUST be the one used in the Map-Request that created the subscription state. If the subscription state was created by explicit configuration at the Map-Server, the starting nonce in the Map-Notify sent as publication MUST be randomly generated by the Map-Server.

(3) The Map-Server MUST use its security association with the xTR to compute the authentication data of the Map-Notify.

When the xTR receives a Map-Notify with an EID not local to the xTR, the xTR knows that the Map-Notify has been received to update an entry on its map-cache. Processing of unsolicited Map-Notify messages MUST be explicitly enabled via configuration at the xTR. The xTR keeps track of the last nonce seen in a Map-Notify received as a publication from the Map-Server for the EID-Record. If a Map-Notify received as a publication has a nonce value that is not greater than the saved nonce, the xTR drops the Map-Notify message and logs the fact a replay attack could have occurred. The same considerations discussed in Section 5.6 of [I-D.ietf-lisp-rfc6833bis] regarding storing Map-Register nonces apply here for Map-Notify nonces.

The xTR processes the received Map-Notify as specified in Section 5.7 of [I-D.ietf-lisp-rfc6833bis], with the following considerations. The xTR MUST use its security association with the Map-Server (see Section 7.1) to validate the authentication data on the Map-Notify. The xTR MUST use the mapping information carried in the Map-Notify to update its internal map-cache. The xTR MUST acknowledge the Map-Notify by sending back a Map-Notify-Ack (specified in Section 5.7 of [I-D.ietf-lisp-rfc6833bis]), with the nonce from the Map-Notify, to the Map-Server. If after a configurable timeout, the Map-Server has not received back the Map-Notify-Ack, it can try to send the Map-Notify to a different ITR-RLOC for that xTR-ID.

7. Security Considerations

Generic security considerations related to LISP control messages are discussed in Section 9 of [I-D.ietf-lisp-rfc6833bis].

In the particular case of PubSub, cache poisoning via malicious Map-Notify messages is avoided by the use of nonce and the security association between the ITRs and the Map-Servers.

7.1. Security Association between ITR and MS

Since Map-Notifies from the Map-Server to the ITR need to be authenticated, there is a need for a soft-state or hard-state security association (e.g. a PubSubKey) between the ITRs and the Map-Servers. For some controlled deployments, it might be possible to have a shared PubSubKey (or set of keys) between the ITRs and the Map-Servers. However, if pre-shared keys are not used in the deployment, LISP-SEC [I-D.ietf-lisp-sec] can be used as follows to create a security association between the ITR and the MS.

First, when the ITR is sending a Map-Request with the N-bit set following Section 5, the ITR also performs the steps described in Section 5.4 of [I-D.ietf-lisp-sec]. The ITR can then generate a PubSubKey by deriving a key from the OTK as follows: PubSubKey = KDF(OTK), where KDF is the Key Derivation Function indicated by the OTK Wrapping ID. If OTK Wrapping ID equals NULL-KEY-WRAP-128 then the PubSubKey is the OTK. Note that as opposed to the pre-shared PubSubKey, this generated PubSubKey is different per EID-Record the ITR subscribes to (since the ITR will use a different OTK per Map-Request).

When the Map-Server receives the Map-Request it follows Section 5. If according to Section 5 the Map-Server is to reply with a Map-Reply (e.g. due to PubSub not supported or subscription not accepted), then it follows normal LISP-SEC procedure described in Section 5.7 of [I-D.ietf-lisp-sec]. No PubSubKey or security association is created in this case.

Otherwise, if, by following Section 5, the Map-Server is to reply with a Map-Notify (e.g. due to subscription accepted) to a received Map-Request, the following extra steps take place (note that if the MS replies with a Map-Notify, none of the regular LISP-SEC steps regarding Map-Reply described in Section 5.7 of [I-D.ietf-lisp-sec] takes place).

- The MS extracts the OTK and OTK Wrapping ID from the LISP-SEC ECM Authentication Data.

- The MS generates a PubSubKey by deriving a key from the OTK as described before for the ITR. This is the same PubSubKey derived at the ITR which is used to establish a security association between the ITR and the MS.

- The PubSubKey can now be used to sign and authenticate any Map-Notify between the MS and the ITR for the subscribed EID-Record. This includes the Map-Notify sent as a confirmation to the subscription. When the ITR wants to update the security association for that MS and EID-Record, it follows again the procedure described in this section.

7.2. DDoS Attack Mitigation

Misbehaving nodes may send massive subscription requests which may lead to exhaust the resources of Map-Servers. Furthermore, frequently changing the state of a subscription may also be considered as an attack vector. To mitigate such issues, xTRs SHOULD rate-limit Map-Requests and Map-Servers SHOULD rate-limit Map-Notifies. Rate-limiting Map-Requests is discussed in Section 5.3 of [I-D.ietf-lisp-rfc6833bis] and the same guidelines apply here. To rate-limit Map-Notifies, a Map-Server MUST NOT send more than one Map-Notify per second to a particular xTR-ID. This parameter MUST be configurable. Note that when the Map-Notify rate-limit threshold is met for a particular xTR-ID, the Map-Server will silently discard additional subscription requests from that xTR-ID. Similarly, for pending mapping updates that need to be notified to that xTR-ID, the Map-Server will combine them into a single Map-Notify (with multiple EID-records) which it will send when the rate-limit mechanism allows it to transmit again Map-Notifies to that xTR-ID.

8. Contributors

Dino Farinacci
lispers.net
San Jose, CA
USA

Email: farinacci@gmail.com

Johnson Leong

Email: johnsonleong@gmail.com

Fabio Maino
Cisco
170 Tasman Drive
San Jose, CA
USA

Email: fmaino@cisco.com

Christian Jacquenet
Orange
Rennes 35000
France

Email: christian.jacquenet@orange.com

Stefano Secci
Cnam
France

Email: stefano.secci@cnam.fr

9. Acknowledgments

This work is partly funded by the ANR LISP-Lab project #ANR-13-INFR-009 (https://www.lisp-lab.org).

10. IANA Considerations

This document is requesting bit allocations in the Map-Request message from the "LISP Control Plane Header Bits" registry introduced in Section 12.6 of [I-D.ietf-lisp-rfc6833bis]. In particular, this document requests allocating the following two bits from the sub-registry "Map-Request Header Bits". The position of these two bits in the Map-Request message can be found in Figure 1.

Spec Name	IANA Name	Bit Position	Description
I	map-request-I	11	xTR-ID Bit
N	map-request-N	... + 0	Notification-Requested Bit

Table 1: Additions to the LISP Map-Request Header Bits Sub-Registry

11. Normative References

[I-D.ietf-lisp-rfc6833bis]
Farinacci, D., Maino, F., Fuller, V., and A. Cabellos-Aparicio, "Locator/ID Separation Protocol (LISP) Control-Plane", draft-ietf-lisp-rfc6833bis-27 (work in progress), January 2020.

[I-D.ietf-lisp-sec]
Maino, F., Ermagan, V., Cabellos-Aparicio, A., and D. Saucez, "LISP-Security (LISP-SEC)", draft-ietf-lisp-sec-21 (work in progress), July 2020.

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

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

Authors' Addresses

Alberto Rodriguez-Natal
Cisco
170 Tasman Drive
San Jose, CA
USA

Email: natal@cisco.com

Vina Ermagan
Google
USA

Email: ermagan@gmail.com

Albert Cabellos
UPC/BarcelonaTech
Barcelona
Spain

Email: acabello@ac.upc.edu

Sharon Barkai
Nexar

Email: sharon.barkai@getnexar.com

Mohamed Boucadair
Orange
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Network Working Group   D. Farinacci
Internet-Draft   lispers.net
Obsoletes: 6830 (if approved)   V. Fuller
Intended status: Standards Track   vaf.net Internet Consulting
Expires: May 22, 2021   D. Meyer
                                                                1-4-5.net
                                                                 D. Lewis
                                                            Cisco Systems
                                                      A. Cabellos (Ed.)
                                                      UPC/BarcelonaTech
                                                      November 18, 2020

The Locator/ID Separation Protocol (LISP)
draft-ietf-lisp-rfc6830bis-36

Abstract

This document describes the Data-Plane protocol for the Locator/ID Separation Protocol (LISP). LISP defines two namespaces, End-point Identifiers (EIDs) that identify end-hosts and Routing Locators (RLOCs) that identify network attachment points. With this, LISP effectively separates control from data, and allows routers to create overlay networks. LISP-capable routers exchange encapsulated packets according to EID-to-RLOC mappings stored in a local Map-Cache.

LISP requires no change to either host protocol stacks or to underlay routers and offers Traffic Engineering, multihoming and mobility, among other features.

This document obsoletes RFC 6830.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document describes the Locator/Identifier Separation Protocol (LISP). LISP is an encapsulation protocol built around the fundamental idea of separating the topological location of a network attachment point from the node's identity [CHIAPPA]. As a result LISP creates two namespaces: Endpoint Identifiers (EIDs), that are used to identify end-hosts (e.g., nodes or Virtual Machines) and routable Routing Locators (RLOCs), used to identify network attachment points. LISP then defines functions for mapping between the two namespaces and for encapsulating traffic originated by devices using non-routable EIDs for transport across a network infrastructure that routes and forwards using RLOCs. LISP encapsulation uses a dynamic form of tunneling where no static provisioning is required or necessary.

LISP is an overlay protocol that separates control from Data-Plane, this document specifies the Data-Plane as well as how LISP-capable routers (Tunnel Routers) exchange packets by encapsulating them to the appropriate location. Tunnel routers are equipped with a cache, called Map-Cache, that contains EID-to-RLOC mappings. The Map-Cache is populated using the LISP Control-Plane protocol [I-D.ietf-lisp-rfc6833bis].

LISP does not require changes to either the host protocol stack or to underlay routers. By separating the EID from the RLOC space, LISP offers native Traffic Engineering, multihoming and mobility, among other features.

Creation of LISP was initially motivated by discussions during the IAB-sponsored Routing and Addressing Workshop held in Amsterdam in October 2006 (see [RFC4984]).

This document specifies the LISP Data-Plane encapsulation and other LISP forwarding node functionality while [I-D.ietf-lisp-rfc6833bis] specifies the LISP control plane. LISP deployment guidelines can be found in [RFC7215] and [RFC6835] describes considerations for network operational management. Finally, [I-D.ietf-lisp-introduction] describes the LISP architecture.

This document obsoletes RFC 6830.

1.1. Scope of Applicability

LISP was originally developed to address the Internet-wide route scaling problem [RFC4984]. While there are a number of approaches of interest for that problem, as LISP as been developed and refined, a large number of other LISP uses have been found and are being used. As such, the design and development of LISP has changed so as to focus on these use cases. The common property of these uses is a large set of cooperating entities seeking to communicate over the public Internet or other large underlay IP infrastructures, while keeping the addressing and topology of the cooperating entities separate from the underlay and Internet topology, routing, and addressing.

2. Requirements Notation

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Definition of Terms

Address Family Identifier (AFI): AFI is a term used to describe an address encoding in a packet. An address family that pertains to addresses found in Data-Plane headers. See [AFN] and [RFC3232] for details. An AFI value of 0 used in this specification indicates an unspecified encoded address where the length of the address is 0 octets following the 16-bit AFI value of 0.

Anycast Address: Anycast Address refers to the same IPv4 or IPv6 address configured and used on multiple systems at the same time. An EID or RLOC can be an anycast address in each of their own address spaces.

Client-side: Client-side is a term used in this document to indicate a connection initiation attempt by an end-system represented by an EID.

Egress Tunnel Router (ETR): An ETR is a router that accepts an IP packet where the destination address in the "outer" IP header is one of its own RLOCs. The router strips the "outer" header and forwards the packet based on the next IP header found. In general, an ETR receives LISP-encapsulated IP packets from the Internet on one side and sends decapsulated IP packets to site end-systems on the other side. ETR functionality does not have to be limited to a router device. A server host can be the endpoint of a LISP tunnel as well.

EID-to-RLOC Database: The EID-to-RLOC Database is a distributed database that contains all known EID-Prefix-to-RLOC mappings. Each potential ETR typically contains a small piece of the database: the EID-to-RLOC mappings for the EID-Prefixes "behind" the router. These map to one of the router's own IP addresses that are routable on the underlay. Note that there MAY be transient conditions when the EID-Prefix for the LISP site and Locator-Set for each EID-Prefix may not be the same on all ETRs. This has no negative implications, since a partial set of Locators can be used.

EID-to-RLOC Map-Cache: The EID-to-RLOC Map-Cache is generally short-lived, on-demand table in an ITR that stores, tracks, and is responsible for timing out and otherwise validating EID-to-RLOC mappings. This cache is distinct from the full "database" of EID-to-RLOC mappings; it is dynamic, local to the ITR(s), and relatively small, while the database is distributed, relatively static, and much more widely scoped to LISP nodes.

EID-Prefix: An EID-Prefix is a power-of-two block of EIDs that are allocated to a site by an address allocation authority. EID-Prefixes are associated with a set of RLOC addresses. EID-Prefix allocations can be broken up into smaller blocks when an RLOC set is to be associated with the larger EID-Prefix block.

End-System: An end-system is an IPv4 or IPv6 device that originates packets with a single IPv4 or IPv6 header. The end-system supplies an EID value for the destination address field of the IP header when communicating outside of its routing domain. An end-system can be a host computer, a switch or router device, or any network appliance.

Endpoint ID (EID): An EID is a 32-bit (for IPv4) or 128-bit (for IPv6) value that identifies a host. EIDs are generally only found in the source and destination address fields of the first (most inner) LISP header of a packet. The host obtains a destination EID the same way it obtains a destination address today, for example, through a Domain Name System (DNS) [RFC1034] lookup or Session Initiation Protocol (SIP) [RFC3261] exchange. The source EID is obtained via existing mechanisms used to set a host's "local" IP address. An EID used on the public Internet MUST have the same properties as any other IP address used in that manner; this means, among other things, that it MUST be unique. An EID is allocated to a host from an EID-Prefix block associated with the site where the host is located. An EID can be used by a host to refer to other hosts. Note that EID blocks MAY be assigned in a hierarchical manner, independent of the network topology, to facilitate scaling of the mapping database. In addition, an EID block assigned to a site MAY have site-local structure (subnetting) for routing within the site; this structure is not visible to the underlay routing system. In theory, the bit string that represents an EID for one device can represent an RLOC for a different device. When used in discussions with other Locator/ID separation proposals, a LISP EID will be called an "LEID". Throughout this document, any references to "EID" refer to an LEID.

Ingress Tunnel Router (ITR): An ITR is a router that resides in a LISP site. Packets sent by sources inside of the LISP site to destinations outside of the site are candidates for encapsulation by the ITR. The ITR treats the IP destination address as an EID and performs an EID-to-RLOC mapping lookup. The router then prepends an "outer" IP header with one of its routable RLOCs (in the RLOC space) in the source address field and the result of the mapping lookup in the destination address field. Note that this destination RLOC may be an intermediate, proxy device that has better knowledge of the EID-to-RLOC mapping closer to the destination EID. In general, an ITR receives IP packets from site end-systems on one side and sends LISP-encapsulated IP packets toward the Internet on the other side.

LISP Header: LISP header is a term used in this document to refer to the outer IPv4 or IPv6 header, a UDP header, and a LISP-specific 8-octet header that follow the UDP header and that an ITR prepends or an ETR strips.

LISP Router: A LISP router is a router that performs the functions of any or all of the following: ITR, ETR, RTR, Proxy-ITR (PITR), or Proxy-ETR (PETR).

LISP Site: LISP site is a set of routers in an edge network that are under a single technical administration. LISP routers that reside in the edge network are the demarcation points to separate the edge network from the core network.

Locator-Status-Bits (LSBs): Locator-Status-Bits are present in the LISP header. They are used by ITRs to inform ETRs about the up/down status of all ETRs at the local site. These bits are used as a hint to convey up/down router status and not path reachability status. The LSBs can be verified by use of one of the Locator reachability algorithms described in Section 10. An ETR MUST rate-limit the action it takes when it detects changes in the Locator-Status-Bits.

Proxy-ETR (PETR): A PETR is defined and described in [RFC6832]. A PETR acts like an ETR but does so on behalf of LISP sites that send packets to destinations at non-LISP sites.

Proxy-ITR (PITR): A PITR is defined and described in [RFC6832]. A PITR acts like an ITR but does so on behalf of non-LISP sites that send packets to destinations at LISP sites.

Recursive Tunneling: Recursive Tunneling occurs when a packet has more than one LISP IP header. Additional layers of tunneling MAY be employed to implement Traffic Engineering or other re-routing as needed. When this is done, an additional "outer" LISP header is added, and the original RLOCs are preserved in the "inner" header.

Re-Encapsulating Tunneling Router (RTR): An RTR acts like an ETR to remove a LISP header, then acts as an ITR to prepend a new LISP header. This is known as Re-encapsulating Tunneling. Doing this allows a packet to be re-routed by the RTR without adding the overhead of additional tunnel headers. When using multiple mapping database systems, care must be taken to not create re-encapsulation loops through misconfiguration.

Route-Returnability: Route-returnability is an assumption that the underlying routing system will deliver packets to the destination. When combined with a nonce that is provided by a sender and returned by a receiver, this limits off-path data insertion. A route-returnability check is verified when a message is sent with a nonce, another message is returned with the same nonce, and the destination of the original message appears as the source of the returned message.

Routing Locator (RLOC): An RLOC is an IPv4 [RFC0791] or IPv6 [RFC8200] address of an Egress Tunnel Router (ETR). An RLOC is the output of an EID-to-RLOC mapping lookup. An EID maps to zero or more RLOCs. Typically, RLOCs are numbered from blocks that are assigned to a site at each point to which it attaches to the underlay network; where the topology is defined by the connectivity of provider networks. Multiple RLOCs can be assigned to the same ETR device or to multiple ETR devices at a site.

Server-side: Server-side is a term used in this document to indicate that a connection initiation attempt is being accepted for a destination EID.

xTR: An xTR is a reference to an ITR or ETR when direction of data flow is not part of the context description. "xTR" refers to the router that is the tunnel endpoint and is used synonymously with the term "Tunnel Router". For example, "An xTR can be located at the Customer Edge (CE) router" indicates both ITR and ETR functionality at the CE router.

4. Basic Overview

One key concept of LISP is that end-systems operate the same way they do today. The IP addresses that hosts use for tracking sockets and connections, and for sending and receiving packets, do not change. In LISP terminology, these IP addresses are called Endpoint Identifiers (EIDs).

Routers continue to forward packets based on IP destination addresses. When a packet is LISP encapsulated, these addresses are referred to as Routing Locators (RLOCs). Most routers along a path between two hosts will not change; they continue to perform routing/forwarding lookups on the destination addresses. For routers between the source host and the ITR as well as routers from the ETR to the destination host, the destination address is an EID. For the routers between the ITR and the ETR, the destination address is an RLOC.

Another key LISP concept is the "Tunnel Router". A Tunnel Router prepends LISP headers on host-originated packets and strips them prior to final delivery to their destination. The IP addresses in this "outer header" are RLOCs. During end-to-end packet exchange between two Internet hosts, an ITR prepends a new LISP header to each packet, and an ETR strips the new header. The ITR performs EID-to-RLOC lookups to determine the routing path to the ETR, which has the RLOC as one of its IP addresses.

Some basic rules governing LISP are:

- o End-systems only send to addresses that are EIDs. EIDs are typically IP addresses assigned to hosts (other types of EID are supported by LISP, see [RFC8060] for further information). End-systems don't know that addresses are EIDs versus RLOCs but assume that packets get to their intended destinations. In a system where LISP is deployed, LISP routers intercept EID-addressed packets and assist in delivering them across the network core where EIDs cannot be routed. The procedure a host uses to send IP packets does not change.

- o LISP routers mostly deal with Routing Locator addresses. See details in Section 4.2 to clarify what is meant by "mostly".

- o RLOCs are always IP addresses assigned to routers, preferably topologically oriented addresses from provider CIDR (Classless Inter-Domain Routing) blocks.

- o When a router originates packets, it MAY use as a source address either an EID or RLOC. When acting as a host (e.g., when terminating a transport session such as Secure SHell (SSH), TELNET, or the Simple Network Management Protocol (SNMP)), it MAY use an EID that is explicitly assigned for that purpose. An EID that identifies the router as a host MUST NOT be used as an RLOC; an EID is only routable within the scope of a site. A typical BGP configuration might demonstrate this "hybrid" EID/RLOC usage where a router could use its "host-like" EID to terminate iBGP sessions to other routers in a site while at the same time using RLOCs to terminate eBGP sessions to routers outside the site.

- o Packets with EIDs in them are not expected to be delivered end-to-end in the absence of an EID-to-RLOC mapping operation. They are expected to be used locally for intra-site communication or to be encapsulated for inter-site communication.

- o EIDs MAY also be structured (subnetted) in a manner suitable for local routing within an Autonomous System (AS).

An additional LISP header MAY be prepended to packets by a TE-ITR when re-routing of the path for a packet is desired. A potential use-case for this would be an ISP router that needs to perform Traffic Engineering for packets flowing through its network. In such a situation, termed "Recursive Tunneling", an ISP transit acts as an additional ITR, and the destination RLOC it uses for the new prepended header would be either a TE-ETR within the ISP (along an intra-ISP traffic engineered path) or a TE-ETR within another ISP (an inter-ISP traffic engineered path, where an agreement to build such a path exists).

In order to avoid excessive packet overhead as well as possible encapsulation loops, this document RECOMMENDS that a maximum of two LISP headers can be prepended to a packet. For initial LISP deployments, it is assumed that two headers is sufficient, where the first prepended header is used at a site for Location/Identity separation and the second prepended header is used inside a service provider for Traffic Engineering purposes.

Tunnel Routers can be placed fairly flexibly in a multi-AS topology. For example, the ITR for a particular end-to-end packet exchange might be the first-hop or default router within a site for the source host. Similarly, the ETR might be the last-hop router directly connected to the destination host. Another example, perhaps for a VPN service outsourced to an ISP by a site, the ITR could be the site's border router at the service provider attachment point. Mixing and matching of site-operated, ISP-operated, and other Tunnel Routers is allowed for maximum flexibility.

4.1. Deployment on the Public Internet

Several of the mechanisms in this document are intended for deployment in controlled, trusted environments, and are insecure for use over the public Internet. In particular, on the public internet xTRs:

- o MUST set the N, L, E, and V bits in the LISP header (Section 5.1) to zero.

- o MUST NOT use Locator-Status-Bits and echo-nonce, as described in Section 10 for Routing Locator Reachability. Instead MUST rely solely on control-plane methods.

- o MUST NOT use Gleaning or Locator-Status-Bits and Map-Versioning, as described in Section 13 to update the EID-to-RLOC Mappings. Instead relying solely on control-plane methods.

4.2. Packet Flow Sequence

This section provides an example of the unicast packet flow, including also Control-Plane information as specified in [I-D.ietf-lisp-rfc6833bis]. The example also assumes the following conditions:

- o Source host "host1.abc.example.com" is sending a packet to "host2.xyz.example.com", exactly as it would if the site was not not using LISP.

- o Each site is multihomed, so each Tunnel Router has an address (RLOC) assigned from the service provider address block for each provider to which that particular Tunnel Router is attached.

- o The ITR(s) and ETR(s) are directly connected to the source and destination, respectively, but the source and destination can be located anywhere in the LISP site.

- o A Map-Request is sent for an external destination when the destination is not found in the forwarding table or matches a default route. Map-Requests are sent to the mapping database system by using the LISP Control-Plane protocol documented in [I-D.ietf-lisp-rfc6833bis].

- o Map-Replies are sent on the underlying routing system topology using the [I-D.ietf-lisp-rfc6833bis] Control-Plane protocol.

Client host1.abc.example.com wants to communicate with server host2.xyz.example.com:

1. host1.abc.example.com wants to open a TCP connection to host2.xyz.example.com. It does a DNS lookup on host2.xyz.example.com. An A/AAAA record is returned. This address is the destination EID. The locally assigned address of host1.abc.example.com is used as the source EID. An IPv4 or IPv6 packet is built and forwarded through the LISP site as a normal IP packet until it reaches a LISP ITR.

2. The LISP ITR must be able to map the destination EID to an RLOC of one of the ETRs at the destination site. A method to do this is to send a LISP Map-Request, as specified in [I-D.ietf-lisp-rfc6833bis].

3. The mapping system helps forwarding the Map-Request to the corresponding ETR. When the Map-Request arrives at one of the ETRs at the destination site, it will process the packet as a control message.

4. The ETR looks at the destination EID of the Map-Request and matches it against the prefixes in the ETR's configured EID-to-RLOC mapping database. This is the list of EID-Prefixes the ETR is supporting for the site it resides in. If there is no match, the Map-Request is dropped. Otherwise, a LISP Map-Reply is returned to the ITR.

5. The ITR receives the Map-Reply message, parses the message, and stores the mapping information from the packet. This information is stored in the ITR's EID-to-RLOC Map-Cache. Note that the Map-Cache is an on-demand cache. An ITR will manage its Map-Cache in such a way that optimizes for its resource constraints.

6. Subsequent packets from host1.abc.example.com to host2.xyz.example.com will have a LISP header prepended by the ITR using the appropriate RLOC as the LISP header destination address learned from the ETR. Note that the packet MAY be sent to a different ETR than the one that returned the Map-Reply due to the source site's hashing policy or the destination site's Locator-Set policy.

7. The ETR receives these packets directly (since the destination address is one of its assigned IP addresses), checks the validity of the addresses, strips the LISP header, and forwards packets to the attached destination host.

8. In order to defer the need for a mapping lookup in the reverse direction, an ETR can OPTIONALLY create a cache entry that maps the source EID (inner-header source IP address) to the source RLOC (outer-header source IP address) in a received LISP packet. Such a cache entry is termed a "glean mapping" and only contains a single RLOC for the EID in question. More complete information about additional RLOCs SHOULD be verified by sending a LISP Map-Request for that EID. Both the ITR and the ETR MAY also influence the decision the other makes in selecting an RLOC.

5. LISP Encapsulation Details

Since additional tunnel headers are prepended, the packet becomes larger and can exceed the MTU of any link traversed from the ITR to the ETR. It is RECOMMENDED in IPv4 that packets do not get fragmented as they are encapsulated by the ITR. Instead, the packet is dropped and an ICMP Unreachable/Fragmentation-Needed message is returned to the source.

In the case when fragmentation is needed, this specification RECOMMENDS that implementations provide support for one of the proposed fragmentation and reassembly schemes. Two existing schemes are detailed in Section 7.

Since IPv4 or IPv6 addresses can be either EIDs or RLOCs, the LISP architecture supports IPv4 EIDs with IPv6 RLOCs (where the inner header is in IPv4 packet format and the outer header is in IPv6 packet format) or IPv6 EIDs with IPv4 RLOCs (where the inner header is in IPv6 packet format and the outer header is in IPv4 packet format). The next sub-sections illustrate packet formats for the homogeneous case (IPv4-in-IPv4 and IPv6-in-IPv6), but all 4 combinations MUST be supported. Additional types of EIDs are defined in [RFC8060].

As LISP uses UDP encapsulation to carry traffic between xTRs across the Internet, implementors should be aware of the provisions of [RFC8085], especially those given in section 3.1.11 on congestion control for UDP tunneling.

Implementors are encouraged to consider UDP checksum usage guidelines in section 3.4 of [RFC8085] when it is desirable to protect UDP and LISP headers against corruption.

5.1. LISP IPv4-in-IPv4 Header Format

```
        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|  IHL  |    DSCP   |ECN|          Total Length         |
    /  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   |         Identification        |Flags|      Fragment Offset    |
   |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   OH  |  Time to Live | Protocol = 17 |         Header Checksum       |
   |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   |                    Source Routing Locator                     |
    \  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     \ |                 Destination Routing Locator                   |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |       Source Port = xxxx      |       Dest Port = 4341        |
   UDP +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     \ |           UDP Length          |        UDP Checksum           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   L   |N|L|E|V|I|R|K|K|            Nonce/Map-Version                  |
   I \ +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   S / |                 Instance ID/Locator-Status-Bits               |
   P   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |Version|  IHL  |    DSCP   |ECN|          Total Length         |
    /  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   |         Identification        |Flags|      Fragment Offset    |
   |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   IH  |  Time to Live |    Protocol   |         Header Checksum       |
   |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   |                           Source EID                          |
    \  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     \ |                         Destination EID                       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

       IHL = IP-Header-Length
```

5.2. LISP IPv6-in-IPv6 Header Format

```
        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|    DSCP   |ECN|           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|R|K|K|            Nonce/Map-Version                  |
   I \ +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   S / |                 Instance ID/Locator-Status-Bits               |
   P   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |Version|    DSCP   |ECN|           Flow Label                  |
    /  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   /   |         Payload Length        | Next Header   |   Hop Limit   |
   v   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                               |
   I   +                                                               +
   n   |                                                               |
   n   +                          Source EID                           +
   e   |                                                               |
   r   +                                                               +
       |                                                               |
   H   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   d   |                                                               |
   r   +                                                               +
       |                                                               |
   ^   +                        Destination EID                        +
   \   |                                                               |
    \  +                                                               +
     \ |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

5.3. Tunnel Header Field Descriptions

Inner Header   (IH): The inner header is the header on the datagram received from the originating host [RFC0791] [RFC8200] [RFC2474]. The source and destination IP addresses are EIDs.

Outer Header: (OH) The outer header is a new header prepended by an ITR. The address fields contain RLOCs obtained from the ingress router's EID-to-RLOC Cache. The IP protocol number is "UDP (17)" from [RFC0768]. The setting of the Don't Fragment (DF) bit 'Flags' field is according to rules listed in Sections 7.1 and 7.2.

UDP Header: The UDP header contains an ITR selected source port when encapsulating a packet. See Section 12 for details on the hash algorithm used to select a source port based on the 5-tuple of the inner header. The destination port MUST be set to the well-known IANA-assigned port value 4341.

UDP Checksum: The 'UDP Checksum' field SHOULD be transmitted as zero by an ITR for either IPv4 [RFC0768] and IPv6 encapsulation [RFC6935] [RFC6936]. When a packet with a zero UDP checksum is received by an ETR, the ETR MUST accept the packet for decapsulation. When an ITR transmits a non-zero value for the UDP checksum, it MUST send a correctly computed value in this field. When an ETR receives a packet with a non-zero UDP checksum, it MAY choose to verify the checksum value. If it chooses to perform such verification, and the verification fails, the packet MUST be silently dropped. If the ETR chooses not to perform the verification, or performs the verification successfully, the packet MUST be accepted for decapsulation. The handling of UDP zero checksums over IPv6 for all tunneling protocols, including LISP, is subject to the applicability statement in [RFC6936].

UDP Length: The 'UDP Length' field is set for an IPv4-encapsulated packet to be the sum of the inner-header IPv4 Total Length plus the UDP and LISP header lengths. For an IPv6-encapsulated packet, the 'UDP Length' field is the sum of the inner-header IPv6 Payload Length, the size of the IPv6 header (40 octets), and the size of the UDP and LISP headers.

N: The N-bit is the nonce-present bit. When this bit is set to 1, the low-order 24 bits of the first 32 bits of the LISP header contain a Nonce. See Section 10.1 for details. Both N- and V-bits MUST NOT be set in the same packet. If they are, a decapsulating ETR MUST treat the 'Nonce/Map-Version' field as having a Nonce value present.

L: The L-bit is the 'Locator-Status-Bits' field enabled bit. When this bit is set to 1, the Locator-Status-Bits in the second 32 bits of the LISP header are in use.

```
 x 1 x x 0 x x x
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|N|L|E|V|I|R|K|K|            Nonce/Map-Version                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Locator-Status-Bits                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

E: The E-bit is the echo-nonce-request bit. This bit MUST be ignored and has no meaning when the N-bit is set to 0. When the N-bit is set to 1 and this bit is set to 1, an ITR is requesting that the nonce value in the 'Nonce' field be echoed back in LISP-encapsulated packets when the ITR is also an ETR. See Section 10.1 for details.

V: The V-bit is the Map-Version present bit. When this bit is set to 1, the N-bit MUST be 0. Refer to Section 13.2 for more details. This bit indicates that the LISP header is encoded in this case as:

```
 0 x 0 1 x x x x
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|N|L|E|V|I|R|K|K|  Source Map-Version   |   Dest Map-Version    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Instance ID/Locator-Status-Bits               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

I: The I-bit is the Instance ID bit. See Section 8 for more details. When this bit is set to 1, the 'Locator-Status-Bits' field is reduced to 8 bits and the high-order 24 bits are used as an Instance ID. If the L-bit is set to 0, then the low-order 8 bits are transmitted as zero and ignored on receipt. The format of the LISP header would look like this:

```
 x x x x 1 x x x
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|N|L|E|V|I|R|K|K|            Nonce/Map-Version                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Instance ID                   |     LSBs      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

R: The R-bit is a Reserved and unassigned bit for future use. It MUST be set to 0 on transmit and MUST be ignored on receipt.

KK: The KK-bits are a 2-bit field used when encapsulated packets are encrypted. The field is set to 00 when the packet is not encrypted. See [RFC8061] for further information.

LISP Nonce: The LISP 'Nonce' field is a 24-bit value that is randomly generated by an ITR when the N-bit is set to 1. Nonce generation algorithms are an implementation matter but are required to generate different nonces when sending to different RLOCs. The nonce is also used when the E-bit is set to request the nonce value to be echoed by the other side when packets are returned. When the E-bit is clear but the N-bit is set, a remote ITR is either echoing a previously requested echo-nonce or providing a random nonce. See Section 10.1 for more details. Finally, when both the N and V-bit are not set (N=0, V=0), then both the Nonce and Map-Version fields are set to 0 and ignored on receipt.

LISP Locator-Status-Bits (LSBs): When the L-bit is also set, the 'Locator-Status-Bits' field in the LISP header is set by an ITR to indicate to an ETR the up/down status of the Locators in the source site. Each RLOC in a Map-Reply is assigned an ordinal value from 0 to n-1 (when there are n RLOCs in a mapping entry). The Locator-Status-Bits are numbered from 0 to n-1 from the least significant bit of the field. The field is 32 bits when the I-bit is set to 0 and is 8 bits when the I-bit is set to 1. When a Locator-Status-Bit is set to 1, the ITR is indicating to the ETR that the RLOC associated with the bit ordinal has up status. See Section 10 for details on how an ITR can determine the status of the ETRs at the same site. When a site has multiple EID-Prefixes that result in multiple mappings (where each could have a different Locator-Set), the Locator-Status-Bits setting in an encapsulated packet MUST reflect the mapping for the EID-Prefix that the inner-header source EID address matches (longest-match). If the LSB for an anycast Locator is set to 1, then there is at least one RLOC with that address, and the ETR is considered 'up'.

When doing ITR/PITR encapsulation:

- The outer-header 'Time to Live' field (or 'Hop Limit' field, in the case of IPv6) SHOULD be copied from the inner-header 'Time to Live' field.

- The outer-header IPv4 'Differentiated Services Code Point' (DSCP) field or the 'Traffic Class' field, in the case of IPv6, SHOULD be copied from the inner-header IPv4 DSCP field or 'Traffic Class' field in the case of IPv6, to the outer-header. Guidelines for this can be found at [RFC2983].

- The IPv4 'Explicit Congestion Notification' (ECN) field and bits 6 and 7 of the IPv6 'Traffic Class' field requires special treatment in order to avoid discarding indications of congestion as specified in [RFC6040].

When doing ETR/PETR decapsulation:

- o The inner-header IPv4 'Time to Live' field or 'Hop Limit' field in the case of IPv6, MUST be copied from the outer-header 'Time to Live'/'Hop Limit' field, when the 'Time to Live'/'Hop Limit' value of the outer header is less than the 'Time to Live'/'Hop Limit' value of the inner header. Failing to perform this check can cause the 'Time to Live'/'Hop Limit' of the inner header to increment across encapsulation/decapsulation cycles. This check is also performed when doing initial encapsulation, when a packet comes to an ITR or PITR destined for a LISP site.

- o The outer-header IPv4 'Differentiated Services Code Point' (DSCP) field or the 'Traffic Class' field in the case of IPv6, SHOULD be copied from the outer-header IPv4 DSCP field or 'Traffic Class' field in the case of IPv6, to the inner-header. Guidelines for this can be found at [RFC2983].

- o The IPv4 'Explicit Congestion Notification' (ECN) field and bits 6 and 7 of the IPv6 'Traffic Class' field, requires special treatment in order to avoid discarding indications of congestion as specified in [RFC6040]. Note that implementations exist that copy the 'ECN' field from the outer header to the inner header even though [RFC6040] does not recommend this behavior. It is RECOMMENDED that implementations change to support the behavior in [RFC6040].

Note that if an ETR/PETR is also an ITR/PITR and chooses to re-encapsulate after decapsulating, the net effect of this is that the new outer header will carry the same Time to Live as the old outer header minus 1.

Copying the Time to Live (TTL) serves two purposes: first, it preserves the distance the host intended the packet to travel; second, and more importantly, it provides for suppression of looping packets in the event there is a loop of concatenated tunnels due to misconfiguration.

Some xTRs and PxTRs performs re-encapsulation operations and need to treat the 'Explicit Congestion Notification' (ECN) in a special way. Because the re-encapsulation operation is a sequence of two operations, namely a decapsulation followed by an encapsulation, the ECN bits MUST be treated as described above for these two operations.

The LISP dataplane protocol is not backwards compatible with [RFC6830] and does not have explicit support for introducing future protocol changes (e.g. an explicit version field). However, the LISP control plane [I-D.ietf-lisp-rfc6833bis] allows an ETR to register dataplane capabilities by means of new LCAF types [RFC8060]. In this way an ITR can be made aware of the dataplane capabilities of an ETR, and encapsulate accordingly. The specification of the new LCAF types, new LCAF mechanisms, and their use, is out of the scope of this document.

6. LISP EID-to-RLOC Map-Cache

ITRs and PITRs maintain an on-demand cache, referred as LISP EID-to-RLOC Map-Cache, that contains mappings from EID-prefixes to locator sets. The cache is used to encapsulate packets from the EID space to the corresponding RLOC network attachment point.

When an ITR/PITR receives a packet from inside of the LISP site to destinations outside of the site a longest-prefix match lookup of the EID is done to the Map-Cache.

When the lookup succeeds, the Locator-Set retrieved from the Map-Cache is used to send the packet to the EID's topological location.

If the lookup fails, the ITR/PITR needs to retrieve the mapping using the LISP Control-Plane protocol [I-D.ietf-lisp-rfc6833bis]. While the mapping is being retrieved, the ITR/PITR can either drop or buffer the packets. This document does not have specific recommendations about the action to be taken. It is up to the deployer to consider whether or not it is desirable to buffer packets and deploy a LISP implementation that offers the desired behaviour. Once the mapping is resolved it is then stored in the local Map-Cache to forward subsequent packets addressed to the same EID-prefix.

The Map-Cache is a local cache of mappings, entries are expired based on the associated Time to live. In addition, entries can be updated with more current information, see Section 13 for further information on this. Finally, the Map-Cache also contains reachability information about EIDs and RLOCs, and uses LISP reachability information mechanisms to determine the reachability of RLOCs, see Section 10 for the specific mechanisms.

7. Dealing with Large Encapsulated Packets

This section proposes two mechanisms to deal with packets that exceed the path MTU between the ITR and ETR.

It is left to the implementor to decide if the stateless or stateful mechanism SHOULD be implemented. Both or neither can be used, since it is a local decision in the ITR regarding how to deal with MTU issues, and sites can interoperate with differing mechanisms.

Both stateless and stateful mechanisms also apply to Re-encapsulating and Recursive Tunneling, so any actions below referring to an ITR also apply to a TE-ITR.

7.1. A Stateless Solution to MTU Handling

An ITR stateless solution to handle MTU issues is described as follows:

1. Define H to be the size, in octets, of the outer header an ITR prepends to a packet. This includes the UDP and LISP header lengths.

2. Define L to be the size, in octets, of the maximum-sized packet an ITR can send to an ETR without the need for the ITR or any intermediate routers to fragment the packet. The network administrator of the LISP deployment has to determine what is the suitable value of L so to make sure that no MTU issues arise.

3. Define an architectural constant S for the maximum size of a packet, in octets, an ITR MUST receive from the source so the effective MTU can be met. That is, L = S + H.

When an ITR receives a packet from a site-facing interface and adds H octets worth of encapsulation to yield a packet size greater than L octets (meaning the received packet size was greater than S octets from the source), it resolves the MTU issue by first splitting the original packet into 2 equal-sized fragments. A LISP header is then prepended to each fragment. The size of the encapsulated fragments is then (S/2 + H), which is less than the ITR's estimate of the path MTU between the ITR and its correspondent ETR.

When an ETR receives encapsulated fragments, it treats them as two individually encapsulated packets. It strips the LISP headers and then forwards each fragment to the destination host of the destination site. The two fragments are reassembled at the destination host into the single IP datagram that was originated by the source host. Note that reassembly can happen at the ETR if the encapsulated packet was fragmented at or after the ITR.

This behavior MUST be performed by the ITR only when the source host originates a packet with the 'DF' field of the IP header set to 0. When the 'DF' field of the IP header is set to 1, or the packet is an IPv6 packet originated by the source host, the ITR will drop the packet when the size (adding in the size of the encapsulation header) is greater than L and send an ICMPv4 ICMP Unreachable/Fragmentation-Needed or ICMPv6 "Packet Too Big" message to the source with a value of S, where S is (L - H).

When the outer-header encapsulation uses an IPv4 header, an implementation SHOULD set the DF bit to 1 so ETR fragment reassembly can be avoided. An implementation MAY set the DF bit in such headers to 0 if it has good reason to believe there are unresolvable path MTU issues between the sending ITR and the receiving ETR.

This specification RECOMMENDS that L be defined as 1500. Additional information about in-network MTU and fragmentation issues can be found at [RFC4459].

7.2. A Stateful Solution to MTU Handling

An ITR stateful solution to handle MTU issues is described as follows:

1. The ITR will keep state of the effective MTU for each Locator per Map-Cache entry. The effective MTU is what the core network can deliver along the path between the ITR and ETR.

2. When an IPv4-encapsulated packet with the DF bit set to 1, exceeds what the core network can deliver, one of the intermediate routers on the path will send an an ICMPv4 Unreachable/Fragmentation-Needed to the ITR, respectively. The ITR will parse the ICMP message to determine which Locator is affected by the effective MTU change and then record the new effective MTU value in the Map-Cache entry.

3. When a packet is received by the ITR from a source inside of the site and the size of the packet is greater than the effective MTU stored with the Map-Cache entry associated with the destination EID the packet is for, the ITR will send an ICMPv4 ICMP Unreachable/Fragmentation-Needed message back to the source. The packet size advertised by the ITR in the ICMP message is the effective MTU minus the LISP encapsulation length.

Even though this mechanism is stateful, it has advantages over the stateless IP fragmentation mechanism, by not involving the destination host with reassembly of ITR fragmented packets.

Please note that [RFC1191] and [RFC1981], which describe the use of ICMP packets for PMTU discovery, can behave suboptimally in the presence of ICMP black holes or off-path attackers that spoof ICMP. Possible mitigations include ITRs and ETRs cooperating on MTU probe packets ([RFC4821], [I-D.ietf-tsvwg-datagram-plpmtud]), or ITRs storing the beginning of large packets to verify that they match the echoed packet in ICMP Frag Needed/PTB.

8. Using Virtualization and Segmentation with LISP

There are several cases where segregation is needed at the EID level. For instance, this is the case for deployments containing overlapping addresses, traffic isolation policies or multi-tenant virtualization. For these and other scenarios where segregation is needed, Instance IDs are used.

An Instance ID can be carried in a LISP-encapsulated packet. An ITR that prepends a LISP header will copy a 24-bit value used by the LISP router to uniquely identify the address space. The value is copied to the 'Instance ID' field of the LISP header, and the I-bit is set to 1.

When an ETR decapsulates a packet, the Instance ID from the LISP header is used as a table identifier to locate the forwarding table to use for the inner destination EID lookup.

For example, an 802.1Q VLAN tag or VPN identifier could be used as a 24-bit Instance ID. See [I-D.ietf-lisp-vpn] for LISP VPN use-case details. Please note that the Instance ID is not protected, an on-path attacker can modify the tags and for instance, allow communicatons between logically isolated VLANs.

Participants within a LISP deployment must agree on the meaning of Instance ID values. The source and destination EIDs MUST belong to the same Instance ID.

Instance ID SHOULD NOT be used with overlapping IPv6 EID addresses.

9. Routing Locator Selection

The Map-Cache contains the state used by ITRs and PITRs to encapsulate packets. When an ITR/PITR receives a packet from inside the LISP site to a destination outside of the site a longest-prefix match lookup of the EID is done to the Map-Cache (see Section 6). The lookup returns a single Locator-Set containing a list of RLOCs corresponding to the EID's topological location. Each RLOC in the Locator-Set is associated with a 'Priority' and 'Weight', this information is used to select the RLOC to encapsulate.

The RLOC with the lowest 'Priority' is selected. An RLOC with 'Priority' 255 means that MUST NOT be used for forwarding. When multiple RLOCs have the same 'Priority' then the 'Weight' states how to load balance traffic among them. The value of the 'Weight' represents the relative weight of the total packets that match the mapping entry.

The following are different scenarios for choosing RLOCs and the controls that are available:

- o The server-side returns one RLOC. The client-side can only use one RLOC. The server-side has complete control of the selection.

- o The server-side returns a list of RLOCs where a subset of the list has the same best Priority. The client can only use the subset list according to the weighting assigned by the server-side. In this case, the server-side controls both the subset list and load-splitting across its members. The client-side can use RLOCs outside of the subset list if it determines that the subset list is unreachable (unless RLOCs are set to a Priority of 255). Some sharing of control exists: the server-side determines the destination RLOC list and load distribution while the client-side has the option of using alternatives to this list if RLOCs in the list are unreachable.

- o The server-side sets a Weight of zero for the RLOC subset list. In this case, the client-side can choose how the traffic load is spread across the subset list. See Section 12 for details on load-sharing mechanisms. Control is shared by the server-side determining the list and the client-side determining load distribution. Again, the client can use alternative RLOCs if the server-provided list of RLOCs is unreachable.

- o Either side (more likely the server-side ETR) decides to "glean" the RLOCs. For example, if the server-side ETR gleans RLOCs, then the client-side ITR gives the client-side ITR responsibility for bidirectional RLOC reachability and preferability. Server-side ETR gleaning of the client-side ITR RLOC is done by caching the inner-header source EID and the outer-header source RLOC of received packets. The client-side ITR controls how traffic is returned and can alternate using an outer-header source RLOC, which then can be added to the list the server-side ETR uses to return traffic. Since no Priority or Weights are provided using this method, the server-side ETR MUST assume that each client-side ITR RLOC uses the same best Priority with a Weight of zero. In addition, since EID-Prefix encoding cannot be conveyed in data packets, the EID-to-RLOC Cache on Tunnel Routers can grow to be very large. Gleaning has several important considerations. A "gleaned" Map-Cache entry is only stored and used for a RECOMMENDED period of 3 seconds, pending verification. Verification MUST be performed by sending a Map-Request to the source EID (the inner-header IP source address) of the received encapsulated packet. A reply to this "verifying Map-Request" is used to fully populate the Map-Cache entry for the "gleaned" EID and is stored and used for the time indicated from the 'TTL' field of a received Map-Reply. When a verified Map- Cache entry is stored, data gleaning no longer occurs for subsequent packets that have a source EID that matches the EID-Prefix of the verified entry. This "gleaning" mechanism MUST NOT be used over the public Internet and SHOULD only be used in trusted and closed deployments. Refer to Section 16 for security issues regarding this mechanism.

RLOCs that appear in EID-to-RLOC Map-Reply messages are assumed to be reachable when the R-bit [I-D.ietf-lisp-rfc6833bis] for the Locator record is set to 1. When the R-bit is set to 0, an ITR or PITR MUST NOT encapsulate to the RLOC. Neither the information contained in a Map-Reply nor that stored in the mapping database system provides reachability information for RLOCs. Note that reachability is not part of the mapping system and is determined using one or more of the Routing Locator reachability algorithms described in the next section.

10. Routing Locator Reachability

Several Data-Plane mechanisms for determining RLOC reachability are currently defined. Please note that additional Control-Plane based reachability mechanisms are defined in [I-D.ietf-lisp-rfc6833bis].

1. An ETR MAY examine the Locator-Status-Bits in the LISP header of an encapsulated data packet received from an ITR. If the ETR is also acting as an ITR and has traffic to return to the original ITR site, it can use this status information to help select an RLOC.

2. When an ETR receives an encapsulated packet from an ITR, the source RLOC from the outer header of the packet is likely to be reachable. Please note that in some scenarios the RLOC from the outer header can be an spoofable field.

3. An ITR/ETR pair can use the 'Echo-Noncing' Locator reachability algorithms described in this section.

When determining Locator up/down reachability by examining the Locator-Status-Bits from the LISP-encapsulated data packet, an ETR will receive up-to-date status from an encapsulating ITR about reachability for all ETRs at the site. CE-based ITRs at the source site can determine reachability relative to each other using the site IGP as follows:

- o Under normal circumstances, each ITR will advertise a default route into the site IGP.

- o If an ITR fails or if the upstream link to its PE fails, its default route will either time out or be withdrawn.

Each ITR can thus observe the presence or lack of a default route originated by the others to determine the Locator-Status-Bits it sets for them.

When ITRs at the site are not deployed in CE routers, the IGP can still be used to determine the reachability of Locators, provided they are injected into the IGP. This is typically done when a /32 address is configured on a loopback interface.

RLOCs listed in a Map-Reply are numbered with ordinals 0 to n-1. The Locator-Status-Bits in a LISP-encapsulated packet are numbered from 0 to n-1 starting with the least significant bit. For example, if an RLOC listed in the 3rd position of the Map-Reply goes down (ordinal value 2), then all ITRs at the site will clear the 3rd least significant bit (xxxx x0xx) of the 'Locator-Status-Bits' field for the packets they encapsulate.

When an xTR decides to use 'Locator-Status-Bits' to affect reachability information, it acts as follows: ETRs decapsulating a packet will check for any change in the 'Locator-Status-Bits' field. When a bit goes from 1 to 0, the ETR, if acting also as an ITR, will refrain from encapsulating packets to an RLOC that is indicated as down. It will only resume using that RLOC if the corresponding Locator-Status-Bit returns to a value of 1. Locator-Status-Bits are associated with a Locator-Set per EID-Prefix. Therefore, when a Locator becomes unreachable, the Locator-Status-Bit that corresponds to that Locator's position in the list returned by the last Map-Reply will be set to zero for that particular EID-Prefix.

Locator-Status-Bits MUST NOT be used over the public Internet and SHOULD only be used in trusted and closed deployments. In addition Locator-Status-Bits SHOULD be coupled with Map-Versioning (Section 13.2) to prevent race conditions where Locator-Status-Bits are interpreted as referring to different RLOCs than intended. Refer to Section 16 for security issues regarding this mechanism.

If an ITR encapsulates a packet to an ETR and the packet is received and decapsulated by the ETR, it is implied but not confirmed by the ITR that the ETR's RLOC is reachable. In most cases, the ETR can also reach the ITR but cannot assume this to be true, due to the possibility of path asymmetry. In the presence of unidirectional traffic flow from an ITR to an ETR, the ITR SHOULD NOT use the lack of return traffic as an indication that the ETR is unreachable. Instead, it MUST use an alternate mechanism to determine reachability.

The security considerations of Section 16 related to data-plane reachability applies to the data-plane RLOC reachability mechanisms described in this section.

10.1. Echo Nonce Algorithm

When data flows bidirectionally between Locators from different sites, a Data-Plane mechanism called "nonce echoing" can be used to determine reachability between an ITR and ETR. When an ITR wants to solicit a nonce echo, it sets the N- and E-bits and places a 24-bit nonce [RFC4086] in the LISP header of the next encapsulated data packet.

When this packet is received by the ETR, the encapsulated packet is forwarded as normal. When the ETR is an xTR (co-located as an ITR), it then sends a data packet to the ITR (when it is an xTR co-located as an ETR), it includes the nonce received earlier with the N-bit set and E-bit cleared. The ITR sees this "echoed nonce" and knows that the path to and from the ETR is up.

The ITR will set the E-bit and N-bit for every packet it sends while in the echo-nonce-request state. The time the ITR waits to process the echoed nonce before it determines the path is unreachable is variable and is a choice left for the implementation.

If the ITR is receiving packets from the ETR but does not see the nonce echoed while being in the echo-nonce-request state, then the path to the ETR is unreachable. This decision MAY be overridden by other Locator reachability algorithms. Once the ITR determines that the path to the ETR is down, it can switch to another Locator for that EID-Prefix.

Note that "ITR" and "ETR" are relative terms here. Both devices MUST be implementing both ITR and ETR functionality for the echo nonce mechanism to operate.

The ITR and ETR MAY both go into the echo-nonce-request state at the same time. The number of packets sent or the time during which echo nonce requests are sent is an implementation-specific setting. In this case, an xTR receiving the echo-nonce-request packets will suspend the echo-nonce-request state and setup a 'echo-nonce-request-state' timer. After the 'echo-nonce-request-state' timer expires it will resume the echo-nonce-request state.

This mechanism does not completely solve the forward path reachability problem, as traffic may be unidirectional. That is, the ETR receiving traffic at a site MAY not be the same device as an ITR that transmits traffic from that site, or the site-to-site traffic is unidirectional so there is no ITR returning traffic.

The echo-nonce algorithm is bilateral. That is, if one side sets the E-bit and the other side is not enabled for echo-noncing, then the echoing of the nonce does not occur and the requesting side may erroneously consider the Locator unreachable. An ITR SHOULD set the E-bit in an encapsulated data packet when it knows the ETR is enabled for echo-noncing. This is conveyed by the E-bit in the Map-Reply message.

Many implementations default to not advertising they are echo-nonce capable in Map-Reply messages and so RLOC-probing tends to be used for RLOC reachability.

The echo-nonce mechanism MUST NOT be used over the public Internet and MUST only be used in trusted and closed deployments. Refer to Section 16 for security issues regarding this mechanism.

11. EID Reachability within a LISP Site

A site MAY be multihomed using two or more ETRs. The hosts and infrastructure within a site will be addressed using one or more EID-Prefixes that are mapped to the RLOCs of the relevant ETRs in the mapping system. One possible failure mode is for an ETR to lose reachability to one or more of the EID-Prefixes within its own site. When this occurs when the ETR sends Map-Replies, it can clear the R-bit associated with its own Locator. And when the ETR is also an ITR, it can clear its Locator-Status-Bit in the encapsulation data header.

It is recognized that there are no simple solutions to the site partitioning problem because it is hard to know which part of the EID-Prefix range is partitioned and which Locators can reach any sub-ranges of the EID-Prefixes. Note that this is not a new problem introduced by the LISP architecture. The problem exists today when a multihomed site uses BGP to advertise its reachability upstream.

12. Routing Locator Hashing

When an ETR provides an EID-to-RLOC mapping in a Map-Reply message that is stored in the Map-Cache of a requesting ITR, the Locator-Set for the EID-Prefix MAY contain different Priority and Weight values for each locator address. When more than one best Priority Locator exists, the ITR can decide how to load-share traffic against the corresponding Locators.

The following hash algorithm MAY be used by an ITR to select a Locator for a packet destined to an EID for the EID-to-RLOC mapping:

1. Either a source and destination address hash or the traditional 5-tuple hash can be used. The traditional 5-tuple hash includes the source and destination addresses; source and destination TCP, UDP, or Stream Control Transmission Protocol (SCTP) port numbers; and the IP protocol number field or IPv6 next-protocol fields of a packet that a host originates from within a LISP site. When a packet is not a TCP, UDP, or SCTP packet, the source and destination addresses only from the header are used to compute the hash.

2. Take the hash value and divide it by the number of Locators stored in the Locator-Set for the EID-to-RLOC mapping.

3. The remainder will yield a value of 0 to "number of Locators minus 1". Use the remainder to select the Locator in the Locator-Set.

The specific hash algorithm the ITR uses for load-sharing is out of scope for this document and does not prevent interoperability.

The Source port SHOULD be the same for all packets belonging to the same flow. Also note that when a packet is LISP encapsulated, the source port number in the outer UDP header needs to be set. Selecting a hashed value allows core routers that are attached to Link Aggregation Groups (LAGs) to load-split the encapsulated packets across member links of such LAGs. Otherwise, core routers would see a single flow, since packets have a source address of the ITR, for packets that are originated by different EIDs at the source site. A suggested setting for the source port number computed by an ITR is a 5-tuple hash function on the inner header, as described above. The source port SHOULD be the same for all packets belonging to the same flow.

Many core router implementations use a 5-tuple hash to decide how to balance packet load across members of a LAG. The 5-tuple hash includes the source and destination addresses of the packet and the source and destination ports when the protocol number in the packet is TCP or UDP. For this reason, UDP encoding is used for LISP encapsulation. In this scenario, when the outer header is IPv6, the flow label MAY also be set following the procedures specified in [RFC6438]. When the inner header is IPv6 then the flow label is not zero, it MAY be used to compute the hash.

13. Changing the Contents of EID-to-RLOC Mappings

Since the LISP architecture uses a caching scheme to retrieve and store EID-to-RLOC mappings, the only way an ITR can get a more up-to-date mapping is to re-request the mapping. However, the ITRs do not know when the mappings change, and the ETRs do not keep track of which ITRs requested its mappings. For scalability reasons, it is desirable to maintain this approach but need to provide a way for ETRs to change their mappings and inform the sites that are currently communicating with the ETR site using such mappings.

This section defines two Data-Plane mechanism for updating EID-to-RLOC mappings. Additionally, the Solicit-Map Request (SMR) Control-Plane updating mechanism is specified in [I-D.ietf-lisp-rfc6833bis].

13.1. Locator-Status-Bits

Locator-Status-Bits (LSB) can also be used to keep track of the Locator status (up or down) when EID-to-RLOC mappings are changing. When LSB are used in a LISP deployment, all LISP tunnel routers MUST implement both ITR and ETR capabilities (therefore all tunnel routers are effectively xTRs). In this section the term "source xTR" is used to refer to the xTR setting the LSB and "destination xTR" is used to refer to the xTR receiving the LSB. The procedure is as follows:

First, when a Locator record is added or removed from the Locator-Set, the source xTR will signal this by sending a Solicit-Map Request (SMR) Control-Plane message [I-D.ietf-lisp-rfc6833bis] to the destination xTR. At this point the source xTR MUST NOT use LSB (L-bit = 0) since the destination xTR site has outdated information. The source xTR will setup a 'use-LSB' timer.

Second and as defined in [I-D.ietf-lisp-rfc6833bis], upon reception of the SMR message the destination xTR will retrieve the updated EID-to-RLOC mappings by sending a Map-Request.

And third, when the 'use-LSB' timer expires, the source xTR can use again LSB with the destination xTR to signal the Locator status (up or down). The specific value for the 'use-LSB' timer depends on the LISP deployment, the 'use-LSB' timer needs to be large enough for the destination xTR to retreive the updated EID-to-RLOC mappings. A RECOMMENDED value for the 'use-LSB' timer is 5 minutes.

13.2. Database Map-Versioning

When there is unidirectional packet flow between an ITR and ETR, and the EID-to-RLOC mappings change on the ETR, it needs to inform the

ITR so encapsulation to a removed Locator can stop and can instead be started to a new Locator in the Locator-Set.

An ETR, when it sends Map-Reply messages, conveys its own Map-Version Number. This is known as the Destination Map-Version Number. ITRs include the Destination Map-Version Number in packets they encapsulate to the site. When an ETR decapsulates a packet and detects that the Destination Map-Version Number is less than the current version for its mapping, the SMR procedure described in [I-D.ietf-lisp-rfc6833bis] occurs.

An ITR, when it encapsulates packets to ETRs, can convey its own Map-Version Number. This is known as the Source Map-Version Number. When an ETR decapsulates a packet and detects that the Source Map-Version Number is greater than the last Map-Version Number sent in a Map-Reply from the ITR's site, the ETR will send a Map-Request to one of the ETRs for the source site.

A Map-Version Number is used as a sequence number per EID-Prefix, so values that are greater are considered to be more recent. A value of 0 for the Source Map-Version Number or the Destination Map-Version Number conveys no versioning information, and an ITR does no comparison with previously received Map-Version Numbers.

A Map-Version Number can be included in Map-Register messages as well. This is a good way for the Map-Server to assure that all ETRs for a site registering to it will be synchronized according to Map-Version Number.

Map-Version requires that ETRs within the LISP site are synchronized with respect to the Map-Version Number, EID-prefix and the set and status (up/down) of the RLOCs. The use of Map-Versioning without proper synchronization may cause traffic disruption. The synchronization protocol is out-of-the-scope of this document, but MUST keep ETRs synchronized within a 1 minute window.

Map-Versioning MUST NOT be used over the public Internet and SHOULD only be used in trusted and closed deployments. Refer to Section 16 for security issues regarding this mechanism.

See [I-D.ietf-lisp-6834bis] for a more detailed analysis and description of Database Map-Versioning.

14. Multicast Considerations

A multicast group address, as defined in the original Internet architecture, is an identifier of a grouping of topologically independent receiver host locations. The address encoding itself does not determine the location of the receiver(s). The multicast routing protocol, and the network-based state the protocol creates, determine where the receivers are located.

In the context of LISP, a multicast group address is both an EID and a Routing Locator. Therefore, no specific semantic or action needs to be taken for a destination address, as it would appear in an IP header. Therefore, a group address that appears in an inner IP header built by a source host will be used as the destination EID. The outer IP header (the destination Routing Locator address), prepended by a LISP router, can use the same group address as the destination Routing Locator, use a multicast or unicast Routing Locator obtained from a Mapping System lookup, or use other means to determine the group address mapping.

With respect to the source Routing Locator address, the ITR prepends its own IP address as the source address of the outer IP header, just like it would if the destination EID was a unicast address. This source Routing Locator address, like any other Routing Locator address, MUST be routable on the underlay.

There are two approaches for LISP-Multicast, one that uses native multicast routing in the underlay with no support from the Mapping System and the other that uses only unicast routing in the underlay with support from the Mapping System. See [RFC6831] and [RFC8378], respectively, for details. Details for LISP-Multicast and interworking with non-LISP sites are described in [RFC6831] and [RFC6832].

15. Router Performance Considerations

LISP is designed to be very "hardware-based forwarding friendly". A few implementation techniques can be used to incrementally implement LISP:

- o When a tunnel-encapsulated packet is received by an ETR, the outer destination address may not be the address of the router. This makes it challenging for the control plane to get packets from the hardware. This may be mitigated by creating special Forwarding Information Base (FIB) entries for the EID-Prefixes of EIDs served by the ETR (those for which the router provides an RLOC translation). These FIB entries are marked with a flag indicating that Control-Plane processing SHOULD be performed. The forwarding logic of testing for particular IP protocol number values is not necessary. There are a few proven cases where no changes to existing deployed hardware were needed to support the LISP Data-Plane.

- o On an ITR, prepending a new IP header consists of adding more octets to a MAC rewrite string and prepending the string as part of the outgoing encapsulation procedure. Routers that support Generic Routing Encapsulation (GRE) tunneling [RFC2784] or 6to4 tunneling [RFC3056] may already support this action.

- o A packet's source address or interface the packet was received on can be used to select VRF (Virtual Routing/Forwarding). The VRF's routing table can be used to find EID-to-RLOC mappings.

For performance issues related to Map-Cache management, see Section 16.

16. Security Considerations

In what follows we highlight security considerations that apply when LISP is deployed in environments such as those specified in Section 1.1.

The optional mechanisms of gleaning is offered to directly obtain a mapping from the LISP encapsulated packets. Specifically, an xTR can learn the EID-to-RLOC mapping by inspecting the source RLOC and source EID of an encapsulated packet, and insert this new mapping into its Map-Cache. An off-path attacker can spoof the source EID address to divert the traffic sent to the victim's spoofed EID. If the attacker spoofs the source RLOC, it can mount a DoS attack by redirecting traffic to the spoofed victim's RLOC, potentially overloading it.

The LISP Data-Plane defines several mechanisms to monitor RLOC Data-Plane reachability, in this context Locator-Status Bits, Nonce-Present and Echo-Nonce bits of the LISP encapsulation header can be manipulated by an attacker to mount a DoS attack. An off-path attacker able to spoof the RLOC and/or nonce of a victim's xTR can manipulate such mechanisms to declare false information about the RLOC's reachability status.

For example of such attacks, an off-path attacker can exploit the echo-nonce mechanism by sending data packets to an ITR with a random nonce from an ETR's spoofed RLOC. Note the attacker must guess a valid nonce the ITR is requesting to be echoed within a small window of time. The goal is to convince the ITR that the ETR's RLOC is reachable even when it may not be reachable. If the attack is successful, the ITR believes the wrong reachability status of the ETR's RLOC until RLOC-probing detects the correct status. This time frame is on the order of 10s of seconds. This specific attack can be mitigated by preventing RLOC spoofing in the network by deploying uRPF BCP 38 [RFC2827]. In addition and in order to exploit this vulnerability, the off-path attacker must send echo-nonce packets at high rate. If the nonces have never been requested by the ITR, it can protect itself from erroneous reachability attacks.

A LISP-specific uRPF check is also possible. When decapsulating, an ETR can check that the source EID and RLOC are valid EID-to-RLOC mappings by checking the Mapping System.

Map-Versioning is a Data-Plane mechanism used to signal a peering xTR that a local EID-to-RLOC mapping has been updated, so that the peering xTR uses LISP Control-Plane signaling message to retrieve a fresh mapping. This can be used by an attacker to forge the map-versioning field of a LISP encapsulated header and force an excessive amount of signaling between xTRs that may overload them.

Locator-Status-Bits, echo-nonce and map-versioning MUST NOT be used over the public Internet and SHOULD only be used in trusted and closed deployments. In addition Locator-Status-Bits SHOULD be coupled with map-versioning to prevent race conditions where Locator-Status-Bits are interpreted as referring to different RLOCs than intended.

LISP implementations and deployments which permit outer header fragments of IPv6 LISP encapsulated packets as a means of dealing with MTU issues should also use implementation techniques in ETRs to prevent this from being a DoS attack vector. Limits on the number of fragments awaiting reassembly at an ETR, RTR, or PETR, and the rate of admitting such fragments may be used.

17. Network Management Considerations

Considerations for network management tools exist so the LISP protocol suite can be operationally managed. These mechanisms can be found in [RFC7052] and [RFC6835].

18. Changes since RFC 6830

For implementation considerations, the following changes have been made to this document since RFC 6830 was published:

- It is no longer mandated that a maximum number of 2 LISP headers be prepended to a packet. If there is a application need for more than 2 LISP headers, an implementation can support more. However, it is RECOMMENDED that a maximum of two LISP headers can be prepended to a packet.

- The 3 reserved flag bits in the LISP header have been allocated for [RFC8061]. The low-order 2 bits of the 3-bit field (now named the KK bits) are used as a key identifier. The 1 remaining bit is still documented as reserved and unassigned.

- o Data-Plane gleaning for creating map-cache entries has been made optional. Any ITR implementations that depend on or assume the remote ETR is gleaning should not do so. This does not create any interoperability problems since the control-plane map-cache population procedures are unilateral and are the typical method for map-cache population.

- o The bulk of the changes to this document which reduces its length are due to moving the LISP control-plane messaging and procedures to [I-D.ietf-lisp-rfc6833bis].

19. IANA Considerations

This section provides guidance to the Internet Assigned Numbers Authority (IANA) regarding registration of values related to this Data-Plane LISP specification, in accordance with BCP 26 [RFC8126].

19.1. LISP UDP Port Numbers

The IANA registry has allocated UDP port number 4341 for the LISP Data-Plane. IANA has updated the description for UDP port 4341 as follows:

```
lisp-data      4341 udp    LISP Data Packets
```

20. References

20.1. Normative References

[I-D.ietf-lisp-6834bis]
Iannone, L., Saucez, D., and O. Bonaventure, "Locator/ID Separation Protocol (LISP) Map-Versioning", draft-ietf-lisp-6834bis-07 (work in progress), October 2020.

[I-D.ietf-lisp-rfc6833bis]
Farinacci, D., Maino, F., Fuller, V., and A. Cabellos-Aparicio, "Locator/ID Separation Protocol (LISP) Control-Plane", draft-ietf-lisp-rfc6833bis-29 (work in progress), September 2020.

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

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <https://www.rfc-editor.org/info/rfc791>.

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

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <https://www.rfc-editor.org/info/rfc2474>.

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

[RFC2983] Black, D., "Differentiated Services and Tunnels", RFC 2983, DOI 10.17487/RFC2983, October 2000, <https://www.rfc-editor.org/info/rfc2983>.

[RFC6040] Briscoe, B., "Tunnelling of Explicit Congestion Notification", RFC 6040, DOI 10.17487/RFC6040, November 2010, <https://www.rfc-editor.org/info/rfc6040>.

[RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, DOI 10.17487/RFC6438, November 2011, <https://www.rfc-editor.org/info/rfc6438>.

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

[RFC6831] Farinacci, D., Meyer, D., Zwiebel, J., and S. Venaas, "The Locator/ID Separation Protocol (LISP) for Multicast Environments", RFC 6831, DOI 10.17487/RFC6831, January 2013, <https://www.rfc-editor.org/info/rfc6831>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8378] Moreno, V. and D. Farinacci, "Signal-Free Locator/ID Separation Protocol (LISP) Multicast", RFC 8378, DOI 10.17487/RFC8378, May 2018, <https://www.rfc-editor.org/info/rfc8378>.

20.2. Informative References

[AFN] IANA, "Address Family Numbers", August 2016, <http://www.iana.org/assignments/address-family-numbers>.

[CHIAPPA] Chiappa, J., "Endpoints and Endpoint names: A Proposed", 1999, <http://mercury.lcs.mit.edu/~jnc/tech/endpoints.txt>.

[I-D.ietf-lisp-introduction] Cabellos-Aparicio, A. and D. Saucez, "An Architectural Introduction to the Locator/ID Separation Protocol (LISP)", draft-ietf-lisp-introduction-13 (work in progress), April 2015.

[I-D.ietf-lisp-vpn] Moreno, V. and D. Farinacci, "LISP Virtual Private Networks (VPNs)", draft-ietf-lisp-vpn-06 (work in progress), August 2020.

[I-D.ietf-tsvwg-datagram-plpmtud] Fairhurst, G., Jones, T., Tuexen, M., Ruengeler, I., and T. Voelker, "Packetization Layer Path MTU Discovery for Datagram Transports", draft-ietf-tsvwg-datagram-plpmtud-22 (work in progress), June 2020.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

[RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <https://www.rfc-editor.org/info/rfc1191>.

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <https://www.rfc-editor.org/info/rfc1918>.

[RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, DOI 10.17487/RFC1981, August 1996, <https://www.rfc-editor.org/info/rfc1981>.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <https://www.rfc-editor.org/info/rfc2784>.

[RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, DOI 10.17487/RFC3056, February 2001, <https://www.rfc-editor.org/info/rfc3056>.

[RFC3232] Reynolds, J., Ed., "Assigned Numbers: RFC 1700 is Replaced by an On-line Database", RFC 3232, DOI 10.17487/RFC3232, January 2002, <https://www.rfc-editor.org/info/rfc3232>.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <https://www.rfc-editor.org/info/rfc3261>.

[RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <https://www.rfc-editor.org/info/rfc4086>.

[RFC4459] Savola, P., "MTU and Fragmentation Issues with In-the-Network Tunneling", RFC 4459, DOI 10.17487/RFC4459, April 2006, <https://www.rfc-editor.org/info/rfc4459>.

[RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, DOI 10.17487/RFC4821, March 2007, <https://www.rfc-editor.org/info/rfc4821>.

[RFC4984] Meyer, D., Ed., Zhang, L., Ed., and K. Fall, Ed., "Report from the IAB Workshop on Routing and Addressing", RFC 4984, DOI 10.17487/RFC4984, September 2007, <https://www.rfc-editor.org/info/rfc4984>.

[RFC6832] Lewis, D., Meyer, D., Farinacci, D., and V. Fuller, "Interworking between Locator/ID Separation Protocol (LISP) and Non-LISP Sites", RFC 6832, DOI 10.17487/RFC6832, January 2013, <https://www.rfc-editor.org/info/rfc6832>.

[RFC6835] Farinacci, D. and D. Meyer, "The Locator/ID Separation Protocol Internet Groper (LIG)", RFC 6835, DOI 10.17487/RFC6835, January 2013, <https://www.rfc-editor.org/info/rfc6835>.

[RFC6935] Eubanks, M., Chimento, P., and M. Westerlund, "IPv6 and UDP Checksums for Tunneled Packets", RFC 6935, DOI 10.17487/RFC6935, April 2013, <https://www.rfc-editor.org/info/rfc6935>.

[RFC6936] Fairhurst, G. and M. Westerlund, "Applicability Statement for the Use of IPv6 UDP Datagrams with Zero Checksums", RFC 6936, DOI 10.17487/RFC6936, April 2013, <https://www.rfc-editor.org/info/rfc6936>.

[RFC7052] Schudel, G., Jain, A., and V. Moreno, "Locator/ID Separation Protocol (LISP) MIB", RFC 7052, DOI 10.17487/RFC7052, October 2013, <https://www.rfc-editor.org/info/rfc7052>.

[RFC7215] Jakab, L., Cabellos-Aparicio, A., Coras, F., Domingo-Pascual, J., and D. Lewis, "Locator/Identifier Separation Protocol (LISP) Network Element Deployment Considerations", RFC 7215, DOI 10.17487/RFC7215, April 2014, <https://www.rfc-editor.org/info/rfc7215>.

[RFC8060] Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format (LCAF)", RFC 8060, DOI 10.17487/RFC8060, February 2017, <https://www.rfc-editor.org/info/rfc8060>.

[RFC8061] Farinacci, D. and B. Weis, "Locator/ID Separation Protocol (LISP) Data-Plane Confidentiality", RFC 8061, DOI 10.17487/RFC8061, February 2017, <https://www.rfc-editor.org/info/rfc8061>.

[RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <https://www.rfc-editor.org/info/rfc8085>.

Appendix A. Acknowledgments

An initial thank you goes to Dave Oran for planting the seeds for the initial ideas for LISP. His consultation continues to provide value to the LISP authors.

A special and appreciative thank you goes to Noel Chiappa for providing architectural impetus over the past decades on separation of location and identity, as well as detailed reviews of the LISP architecture and documents, coupled with enthusiasm for making LISP a practical and incremental transition for the Internet.

The original authors would like to gratefully acknowledge many people who have contributed discussions and ideas to the making of this proposal. They include Scott Brim, Andrew Partan, John Zwiebel, Jason Schiller, Lixia Zhang, Dorian Kim, Peter Schoenmaker, Vijay Gill, Geoff Huston, David Conrad, Mark Handley, Ron Bonica, Ted Seely, Mark Townsley, Chris Morrow, Brian Weis, Dave McGrew, Peter Lothberg, Dave Thaler, Eliot Lear, Shane Amante, Ved Kafle, Olivier Bonaventure, Luigi Iannone, Robin Whittle, Brian Carpenter, Joel Halpern, Terry Manderson, Roger Jorgensen, Ran Atkinson, Stig Venaas, Iljitsch van Beijnum, Roland Bless, Dana Blair, Bill Lynch, Marc Woolward, Damien Saucez, Damian Lezama, Attilla De Groot, Parantap Lahiri, David Black, Roque Gagliano, Isidor Kouvelas, Jesper Skriver, Fred Templin, Margaret Wasserman, Sam Hartman, Michael Hofling, Pedro Marques, Jari Arkko, Gregg Schudel, Srinivas Subramanian, Amit Jain, Xu Xiaohu, Dhirendra Trivedi, Yakov Rekhter, John Scudder, John Drake, Dimitri Papadimitriou, Ross Callon, Selina Heimlich, Job Snijders, Vina Ermagan, Fabio Maino, Victor Moreno, Chris White, Clarence Filsfils, Alia Atlas, Florin Coras and Alberto Rodriguez.

This work originated in the Routing Research Group (RRG) of the IRTF. An individual submission was converted into the IETF LISP working group document that became this RFC.

The LISP working group would like to give a special thanks to Jari Arkko, the Internet Area AD at the time that the set of LISP documents were being prepared for IESG last call, and for his meticulous reviews and detailed commentaries on the 7 working group last call documents progressing toward standards-track RFCs.

The current authors would like to give a sincere thank you to the people who help put LISP on standards track in the IETF. They include Joel Halpern, Luigi Iannone, Deborah Brungard, Fabio Maino, Scott Bradner, Kyle Rose, Takeshi Takahashi, Sarah Banks, Pete Resnick, Colin Perkins, Mirja Kuhlewind, Francis Dupont, Benjamin Kaduk, Eric Rescorla, Alvaro Retana, Alexey Melnikov, Alissa Cooper, Suresh Krishnan, Alberto Rodriguez-Natal, Vina Ermagen, Mohamed

Boucadair, Brian Trammell, Sabrina Tanamal, and John Drake. The contributions they offered greatly added to the security, scale, and robustness of the LISP architecture and protocols.

Appendix B. Document Change Log

[RFC Editor: Please delete this section on publication as RFC.]

B.1. Changes to draft-ietf-lisp-rfc6830bis-27

- o Posted November 2019.
- o Fixed how LSB behave in the presence of new/removed locators.
- o Added ETR synchronization requirements when using Map-Versioning.
- o Fixed a large set of minor comments and edits.

B.2. Changes to draft-ietf-lisp-rfc6830bis-27

- o Posted April 2019 post telechat.
- o Made editorial corrections per Warren's suggestions.
- o Put in suggested text from Luigi that Mirja agreed with.
- o LSB, Echo-Nonce and Map-Versioning SHOULD be only used in closed environments.
- o Removed paragraph stating that Instance-ID can be 32-bit in the control-plane.
- o 6831/8378 are now normative.
- o Rewritten Security Considerations according to the changes.
- o Stated that LSB SHOULD be coupled with Map-Versioning.

B.3. Changes to draft-ietf-lisp-rfc6830bis-26

- o Posted late October 2018.
- o Changed description about "reserved" bits to state "reserved and unassigned".

B.4. Changes to draft-ietf-lisp-rfc6830bis-25

- o Posted mid October 2018.

- o Added more to the Security Considerations section with discussion about echo-nonce attacks.

B.5. Changes to draft-ietf-lisp-rfc6830bis-24

- o Posted mid October 2018.

- o Final editorial changes for Eric and Ben.

B.6. Changes to draft-ietf-lisp-rfc6830bis-23

- o Posted early October 2018.

- o Added an applicability statement in section 1 to address security concerns from Telechat.

B.7. Changes to draft-ietf-lisp-rfc6830bis-22

- o Posted early October 2018.

- o Changes to reflect comments post Telechat.

B.8. Changes to draft-ietf-lisp-rfc6830bis-21

- o Posted late-September 2018.

- o Changes to reflect comments from Sep 27th Telechat.

B.9. Changes to draft-ietf-lisp-rfc6830bis-20

- o Posted late-September 2018.

- o Fix old reference to RFC3168, changed to RFC6040.

B.10. Changes to draft-ietf-lisp-rfc6830bis-19

- o Posted late-September 2018.

- o More editorial changes.

B.11. Changes to draft-ietf-lisp-rfc6830bis-18

- o Posted mid-September 2018.

- o Changes to reflect comments from Secdir review (Mirja).

B.12. Changes to draft-ietf-lisp-rfc6830bis-17

- o Posted September 2018.

- o Indicate in the "Changes since RFC 6830" section why the document has been shortened in length.

- o Make reference to RFC 8085 about UDP congestion control.

- o More editorial changes from multiple IESG reviews.

B.13. Changes to draft-ietf-lisp-rfc6830bis-16

- o Posted late August 2018.

- o Distinguish the message type names between ICMP for IPv4 and ICMP for IPv6 for handling MTU issues.

B.14. Changes to draft-ietf-lisp-rfc6830bis-15

- o Posted August 2018.

- o Final editorial changes before RFC submission for Proposed Standard.

- o Added section "Changes since RFC 6830" so implementers are informed of any changes since the last RFC publication.

B.15. Changes to draft-ietf-lisp-rfc6830bis-14

- o Posted July 2018 IETF week.

- o Put obsolete of RFC 6830 in Intro section in addition to abstract.

B.16. Changes to draft-ietf-lisp-rfc6830bis-13

- o Posted March IETF Week 2018.

- o Clarified that a new nonce is required per RLOC.

- o Removed 'Clock Sweep' section. This text must be placed in a new OAM document.

- Some references changed from normative to informative

B.17. Changes to draft-ietf-lisp-rfc6830bis-12

- Posted July 2018.

- Fixed Luigi editorial comments to ready draft for RFC status.

B.18. Changes to draft-ietf-lisp-rfc6830bis-11

- Posted March 2018.

- Removed sections 16, 17 and 18 (Mobility, Deployment and Traceroute considerations). This text must be placed in a new OAM document.

B.19. Changes to draft-ietf-lisp-rfc6830bis-10

- Posted March 2018.

- Updated section 'Router Locator Selection' stating that the Data-Plane MUST follow what's stored in the Map-Cache (priorities and weights).

- Section 'Routing Locator Reachability': Removed bullet point 2 (ICMP Network/Host Unreachable),3 (hints from BGP),4 (ICMP Port Unreachable),5 (receive a Map-Reply as a response) and RLOC probing

- Removed 'Solicit-Map Request'.

B.20. Changes to draft-ietf-lisp-rfc6830bis-09

- Posted January 2018.

- Add more details in section 5.3 about DSCP processing during encapsulation and decapsulation.

- Added clarity to definitions in the Definition of Terms section from various commenters.

- Removed PA and PI definitions from Definition of Terms section.

- More editorial changes.

- Removed 4342 from IANA section and move to RFC6833 IANA section.

B.21. Changes to draft-ietf-lisp-rfc6830bis-08

- Posted January 2018.
- Remove references to research work for any protocol mechanisms.
- Document scanned to make sure it is RFC 2119 compliant.
- Made changes to reflect comments from document WG shepherd Luigi Iannone.
- Ran IDNITs on the document.

B.22. Changes to draft-ietf-lisp-rfc6830bis-07

- Posted November 2017.
- Rephrase how Instance-IDs are used and don't refer to [RFC1918] addresses.

B.23. Changes to draft-ietf-lisp-rfc6830bis-06

- Posted October 2017.
- Put RTR definition before it is used.
- Rename references that are now working group drafts.
- Remove "EIDs MUST NOT be used as used by a host to refer to other hosts. Note that EID blocks MAY LISP RLOCs".
- Indicate what address-family can appear in data packets.
- ETRs may, rather than will, be the ones to send Map-Replies.
- Recommend, rather than mandate, max encapsulation headers to 2.
- Reference VPN draft when introducing Instance-ID.
- Indicate that SMRs can be sent when ITR/ETR are in the same node.
- Clarify when private addresses can be used.

B.24. Changes to draft-ietf-lisp-rfc6830bis-05

- Posted August 2017.
- Make it clear that a Re-encapsulating Tunnel Router is an RTR.

B.25. Changes to draft-ietf-lisp-rfc6830bis-04

- Posted July 2017.
- Changed reference of IPv6 RFC2460 to RFC8200.
- Indicate that the applicability statement for UDP zero checksums over IPv6 adheres to RFC6936.

B.26. Changes to draft-ietf-lisp-rfc6830bis-03

- Posted May 2017.
- Move the control-plane related codepoints in the IANA Considerations section to RFC6833bis.

B.27. Changes to draft-ietf-lisp-rfc6830bis-02

- Posted April 2017.
- Reflect some editorial comments from Damien Sausez.

B.28. Changes to draft-ietf-lisp-rfc6830bis-01

- Posted March 2017.
- Include references to new RFCs published.
- Change references from RFC6833 to RFC6833bis.
- Clarified LCAF text in the IANA section.
- Remove references to "experimental".

B.29. Changes to draft-ietf-lisp-rfc6830bis-00

- Posted December 2016.
- Created working group document from draft-farinacci-lisp -rfc6830-00 individual submission. No other changes made.

Authors' Addresses

Dino Farinacci
lispers.net

EMail: farinacci@gmail.com

Vince Fuller
vaf.net Internet Consulting

EMail: vince.fuller@gmail.com

Dave Meyer
1-4-5.net

EMail: dmm@1-4-5.net

Darrel Lewis
Cisco Systems
170 Tasman Drive
San Jose, CA
USA

EMail: darlewis@cisco.com

Albert Cabellos
UPC/BarcelonaTech
Campus Nord, C. Jordi Girona 1-3
Barcelona, Catalunya
Spain

EMail: acabello@ac.upc.edu

Network Working Group
Internet-Draft
Obsoletes: 6830, 6833 (if approved)
Intended status: Standards Track
Expires: May 22, 2021

D. Farinacci
lispers.net
F. Maino
Cisco Systems
V. Fuller
vaf.net Internet Consulting
A. Cabellos (Ed.)
UPC/BarcelonaTech
November 18, 2020

Locator/ID Separation Protocol (LISP) Control-Plane
draft-ietf-lisp-rfc6833bis-30

Abstract

This document describes the Control-Plane and Mapping Service for the Locator/ID Separation Protocol (LISP), implemented by two types of LISP-speaking devices -- the LISP Map-Resolver and LISP Map-Server -- that provides a simplified "front end" for one or more Endpoint ID to Routing Locator mapping databases.

By using this Control-Plane service interface and communicating with Map-Resolvers and Map-Servers, LISP Ingress Tunnel Routers (ITRs) and Egress Tunnel Routers (ETRs) are not dependent on the details of mapping database systems, which facilitates modularity with different database designs. Since these devices implement the "edge" of the LISP Control-Plane infrastructure, connecting EID addressable nodes of a LISP site, it the implementation and operational complexity of the overall cost and effort of deploying LISP.

This document obsoletes RFC 6830 and RFC 6833.

Status of This Memo

This Internet-Draft will expire on May 22, 2021.

Copyright Notice

Table of Contents

1. Introduction

The Locator/ID Separation Protocol [I-D.ietf-lisp-rfc6830bis] (see also [I-D.ietf-lisp-introduction]) specifies an architecture and mechanism for dynamic tunneling by logically separating the addresses currently used by IP in two separate name spaces: Endpoint IDs (EIDs), used within sites; and Routing Locators (RLOCs), used on the transit networks that make up the Internet infrastructure. To achieve this separation, LISP defines protocol mechanisms for mapping from EIDs to RLOCs. In addition, LISP assumes the existence of a database to store and propagate those mappings across mapping system nodes. Several such databases have been proposed; among them are the Content distribution Overlay Network Service for LISP-NERD (a Not-so-novel EID-to-RLOC Database) [RFC6837], LISP Alternative Logical Topology (LISP-ALT) [RFC6836], and LISP Delegated Database Tree (LISP-DDT) [RFC8111].

The LISP Mapping Service defines two types of LISP-speaking devices: the Map-Resolver, which accepts Map-Requests from an Ingress Tunnel Router (ITR) and "resolves" the EID-to-RLOC mapping using a mapping database; and the Map-Server, which learns authoritative EID-to-RLOC mappings from an Egress Tunnel Router (ETR) and publishes them in a database.

This LISP Control-Plane Mapping Service can be used by many different encapsulation-based or translation-based Data-Planes which include but are not limited to the ones defined in LISP RFC 6830bis [I-D.ietf-lisp-rfc6830bis], LISP-GPE [I-D.ietf-lisp-gpe], VXLAN [RFC7348], VXLAN-GPE [I-D.ietf-nvo3-vxlan-gpe], GRE [RFC2890], GTP [GTP-3GPP], ILA [I-D.herbert-intarea-ila], and Segment Routing (SRv6) [RFC8402].

Conceptually, LISP Map-Servers share some of the same basic configuration and maintenance properties as Domain Name System (DNS) [RFC1035] servers; likewise, Map-Resolvers are conceptually similar to DNS caching resolvers. With this in mind, this specification borrows familiar terminology (resolver and server) from the DNS specifications.

Note this document doesn't assume any particular database mapping infrastructure to illustrate certain aspects of Map-Server and Map-Resolver operation. The Mapping Service interface can (and likely will) be used by ITRs and ETRs to access other mapping database systems as the LISP infrastructure evolves.

LISP is not intended to address problems of connectivity and scaling on behalf of arbitrary communicating parties. Relevant situations are described in the scoping section of the introduction to [I-D.ietf-lisp-rfc6830bis].

This document obsoletes RFC 6830 and 6833.

1.1. Scope of Applicability

LISP was originally developed to address the Internet-wide route scaling problem [RFC4984]. While there are a number of approaches of interest for that problem, as LISP as been developed and refined, a large number of other LISP uses have been found and are being used. As such, the design and development of LISP has changed so as to focus on these use cases. The common property of these uses is a large set of cooperating entities seeking to communicate over the public Internet or other large underlay IP infrastructures, while keeping the addressing and topology of the cooperating entities separate from the underlay and Internet topology, routing, and addressing.

When communicating over the public Internet, deployers MUST consider the following guidelines:

1. LISP-SEC MUST be implemented [I-D.ietf-lisp-sec]. This means that the S-bit MUST be set in the Map-Reply (Section 5.4), Map-Register (Section 5.6) and Encapsulated Control messages (Section 5.8).

2. Implementations SHOULD use the 'HMAC-SHA256-128+HKDF-SHA256' as the Algorithm ID (Section 12.5) in Map-Register message (Section 5.6), and MUST NOT use 'None' or 'HMAC-SHA-1-96-None' as Algorithm ID (Section 12.5) in the Map-Register message (Section 5.6)

2. Requirements Notation

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Definition of Terms

Map-Server: A network infrastructure component that learns of EID-Prefix mapping entries from an ETR, via the registration mechanism described below, or some other authoritative source if one exists. A Map-Server publishes these EID-Prefixes in a mapping database.

Map-Request: A LISP Map-Request is a Control-Plane message to query the mapping system to resolve an EID. A LISP Map-Request can also be sent to an RLOC to test for reachability and to exchange security keys between an encapsulator and a decapsulator. This type of Map-Request is also known as an RLOC-Probe Request.

Map-Reply: A LISP Map-Reply is a Control-Plane message returned in response to a Map-Request sent to the mapping system when resolving an EID. A LISP Map-Reply can also be returned by a decapsulator in response to a Map-Request sent by an encapsulator to test for reachability. This type of Map-Reply is known as a RLOC-Probe Reply.

Encapsulated Map-Request: A LISP Map-Request carried within an Encapsulated Control Message (ECM), which has an additional LISP header prepended. Sent to UDP destination port 4342. The "outer" addresses are routable IP addresses, also known as RLOCs. Used by an ITR when sending to a Map-Resolver and by a Map-Server when forwarding a Map-Request to an ETR.

Map-Resolver: A network infrastructure component that accepts LISP Encapsulated (ECM) Map-Requests, typically from an ITR, and determines whether or not the destination IP address is part of the EID namespace; if it is not, a Negative Map-Reply is returned. Otherwise, the Map-Resolver finds the appropriate EID-to-RLOC mapping by consulting a mapping database system.

Negative Map-Reply: A LISP Map-Reply that contains an empty Locator-Set. Returned in response to a Map-Request if the destination EID is not registered in the mapping system, is policy denied or fails authentication.

Map-Register message: A LISP message sent by an ETR to a Map-Server to register its associated EID-Prefixes. In addition to the set of EID-Prefixes to register, the message includes one or more RLOCs to reach ETR(s). The Map-Server uses these RLOCs when forwarding Map-Requests (re-formatted as Encapsulated Map-Requests). An ETR MAY request that the Map-Server answer Map-Requests on its behalf by setting the "proxy Map-Reply" flag (P-bit) in the message.

Map-Notify message: A LISP message sent by a Map-Server to an ETR to confirm that a Map-Register has been received and processed. An ETR requests that a Map-Notify be returned by setting the "want-map-notify" flag (M-bit) in the Map-Register message. Unlike a Map-Reply, a Map-Notify uses UDP port 4342 for both source and destination. Map-Notify messages are also sent to ITRs by Map-Servers when there are RLOC-set changes.

For definitions of other terms, notably Ingress Tunnel Router (ITR), Egress Tunnel Router (ETR), and Re-encapsulating Tunnel Router (RTR), refer to the LISP Data-Plane specification [I-D.ietf-lisp-rfc6830bis].

4. Basic Overview

A Map-Server is a device that publishes EID-Prefixes in a LISP mapping database on behalf of a set of ETRs. When it receives a Map Request (typically originating from an ITR), it consults the mapping database to find an ETR that can answer with the set of RLOCs for an EID-Prefix. To publish its EID-Prefixes, an ETR periodically sends Map-Register messages to the Map-Server. A Map-Register message contains a list of EID-Prefixes plus a set of RLOCs that can be used to reach the ETRs.

When LISP-ALT [RFC6836] is used as the mapping database, a Map-Server connects to the ALT network and acts as a "last-hop" ALT-Router. Intermediate ALT-Routers forward Map-Requests to the Map-Server that advertises a particular EID-Prefix, and the Map-Server forwards them to the owning ETR, which responds with Map-Reply messages.

When LISP-DDT [RFC8111] is used as the mapping database, a Map-Server sends the final Map-Referral messages from the Delegated Database Tree.

A Map-Resolver receives Encapsulated Map-Requests from its client ITRs and uses a mapping database system to find the appropriate ETR to answer those requests. On a LISP-ALT network, a Map-Resolver acts as a "first-hop" ALT-Router. It has Generic Routing Encapsulation (GRE) tunnels configured to other ALT-Routers and uses BGP to learn paths to ETRs for different prefixes in the LISP-ALT database. The Map-Resolver uses this path information to forward Map-Requests over the ALT to the correct ETRs. On a LISP-DDT network [RFC8111], a Map-Resolver maintains a referral-cache and acts as a "first-hop" DDT-node. The Map-Resolver uses the referral information to forward Map-Requests.

Note that while it is conceivable that a Map-Resolver could cache responses to improve performance, issues surrounding cache management would need to be resolved so that doing so will be reliable and practical. In this specification, Map-Resolvers will operate only in a non-caching mode, decapsulating and forwarding Encapsulated Map Requests received from ITRs. Any specification of caching functionality is out of scope for this document.

Note that a single device can implement the functions of both a Map-Server and a Map-Resolver, and in many cases the functions will be co-located in that way. Also, there can be ALT-only nodes and DDT-only nodes, when LISP-ALT and LISP-DDT are used, respectively, to connecting Map-Resolvers and Map-Servers together to make up the Mapping System.

5. LISP IPv4 and IPv6 Control-Plane Packet Formats

The following UDP packet formats are used by the LISP control plane.

```
        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|  IHL  |Type of Service|          Total Length         |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |         Identification        |Flags|      Fragment Offset    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |  Time to Live | Protocol = 17 |         Header Checksum       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                    Source Routing Locator                     |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                 Destination Routing Locator                   |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |           Source Port         |         Dest Port             |
   UDP +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     \ |           UDP Length          |        UDP Checksum           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                               |
       |                         LISP Message                          |
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPv4 UDP LISP Control Message

```
        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   |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                               |
       +                                                               +
       |                                                               |
       +                     Source Routing Locator                    +
       |                                                               |
       +                                                               +
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                               |
       +                                                               +
       |                                                               |
       +                  Destination Routing Locator                  +
       |                                                               |
       +                                                               +
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |           Source Port         |         Dest Port             |
   UDP +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     \ |           UDP Length          |        UDP Checksum           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                               |
       |                         LISP Message                          |
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPv6 UDP LISP Control Message

When a UDP Map-Request, Map-Register, or Map-Notify (when used as a notification message) are sent, the UDP source port is chosen by the sender and the destination UDP port number is set to 4342. When a UDP Map-Reply, Map-Notify (when used as an acknowledgement to a Map-Register), or Map-Notify-Ack are sent, the source UDP port number is set to 4342 and the destination UDP port number is copied from the source port of either the Map-Request or the invoking data packet. Implementations MUST be prepared to accept packets when either the source port or destination UDP port is set to 4342 due to NATs changing port number values.

The 'UDP Length' field will reflect the length of the UDP header and the LISP Message payload. LISP is expected to be deployed by cooperating entities communicating over underlays. Deployers are expected to set the MTU according to the specific deployment guidelines to prevent fragmentation of either the inner packet or the outer encapsulated packet. For deployments not aware of the underlay restrictions on path MTU, the message size MUST be limited to 576 bytes for IPv4 or 1280 bytes for IPv6 -considering the entire IP packet- as outlined in [RFC8085].

The UDP checksum is computed and set to non-zero for all messages sent to or from port 4342. It MUST be checked on receipt, and if the checksum fails, the control message MUST be dropped [RFC1071].

The format of control messages includes the UDP header so the checksum and length fields can be used to protect and delimit message boundaries.

5.1. LISP Control Packet Type Allocations

This section defines the LISP control message formats and summarizes for IANA the LISP Type codes assigned by this document. For completeness, the summary below includes the LISP Shared Extension Message assigned by [I-D.ietf-lisp-rfc8113bis]. Message type definitions are:

```
Reserved:                          0     b'0000'
LISP Map-Request:                  1     b'0001'
LISP Map-Reply:                    2     b'0010'
LISP Map-Register:                 3     b'0011'
LISP Map-Notify:                   4     b'0100'
LISP Map-Notify-Ack:               5     b'0101'
LISP Map-Referral:                 6     b'0110'
Unassigned                         7     b'0111'
LISP Encapsulated Control Message: 8     b'1000'
Unassigned                         9-14  b'1001'- b'1110'
LISP Shared Extension Message:     15    b'1111'
```

Protocol designers experimenting with new message formats are recommended to use the LISP Shared Extension Message Type described in [I-D.ietf-lisp-rfc8113bis].

All LISP Control-Plane messages use Address Family Identifiers (AFI) [AFI] or LISP Canonical Address Format (LCAF) [RFC8060] formats to encode either fixed or variable length addresses. This includes explicit fields in each control message or part of EID-records or RLOC-records in commonly formatted messages. LISP control-plane messages that include an unrecognized AFI MUST be dropped and the event MUST be logged.

The LISP control-plane describes how other data-planes can encode messages to support the Soliciting of Map-Requests as well as RLOC-probing procedures.

5.2. Map-Request Message Format

```
        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
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |Type=1 |A|M|P|S|p|s|R|R|  Rsvd   |L|D|   IRC   | Record Count  |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         Nonce . . .                           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         . . . Nonce                           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |         Source-EID-AFI        |   Source EID Address  ...     |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |         ITR-RLOC-AFI 1        |    ITR-RLOC Address 1  ...    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                              ...                              |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |         ITR-RLOC-AFI n        |    ITR-RLOC Address n  ...    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |   Reserved    | EID mask-len  |        EID-Prefix-AFI         |
   Rec +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     \ |                       EID-Prefix  ...                         |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                   Map-Reply Record  ...                       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Packet field descriptions:

Type: 1 (Map-Request)

A: This is an authoritative bit, it is set to 1 when an ITR wants the destination site to return the Map-Reply rather than the mapping database system returning a Map-Reply, and set to 0 otherwise.

M: This is the map-data-present bit. When set, it indicates that a Map-Reply Record segment is included in the Map-Request.

P: This is the probe-bit, which indicates that a Map-Request MUST be treated as a Locator reachability probe. The receiver MUST respond with a Map-Reply with the probe-bit set, indicating that the Map-Reply is a Locator reachability probe reply, with the nonce copied from the Map-Request. See RLOC-Probing Section 7.1 for more details. This RLOC-probe Map-Request MUST NOT be sent to the mapping system. If a Map-Resolver or Map-Server receives a Map-Request with the probe-bit set, it MUST drop the message.

S: This is the Solicit-Map-Request (SMR) bit. See Solicit-Map-Request (SMRs) Section 6.1 for details.

p: This is the PITR bit. This bit is set to 1 when a PITR sends a Map-Request. The use of this bit is deployment-specific.

s: This is the SMR-invoked bit. This bit is set to 1 when an xTR is sending a Map-Request in response to a received SMR-based Map-Request.

R: This reserved and unassigned bit MUST be set to 0 on transmit and MUST be ignored on receipt.

Rsvd: This field MUST be set to 0 on transmit and MUST be ignored on receipt.

L: This is the local-xtr bit. It is used by an xTR in a LISP site to tell other xTRs in the same site that it is part of the RLOC-set for the LISP site. The L-bit is set to 1 when the RLOC is the sender's IP address.

D: This is the dont-map-reply bit. It is used in the SMR procedure described in Section 6.1. When an xTR sends an SMR message, it doesn't need a Map-Reply returned. When this bit is set, the receiver of the Map-Request does not return a Map-Reply.

IRC: This 5-bit field is the ITR-RLOC Count, which encodes the additional number of ('ITR-RLOC-AFI', 'ITR-RLOC Address') fields present in this message. At least one (ITR-RLOC-AFI, ITR-RLOC-Address) pair MUST be encoded. Multiple 'ITR-RLOC Address' fields are used, so a Map-Replier can select which destination address to use for a Map-Reply. The IRC value ranges from 0 to 31. For a value of 0, there is 1 ITR-RLOC address encoded; for a value of 1, there are 2 ITR-RLOC addresses encoded, and so on up to 31, which encodes a total of 32 ITR-RLOC addresses.

Record Count: This is the number of records in this Map-Request message. A record is comprised of the portion of the packet that is labeled 'Rec' above and occurs the number of times equal to Record Count. For this version of the protocol, a receiver MUST accept and process Map-Requests that contain one or more records, but a sender MUST only send Map-Requests containing one record.

Nonce: This is an 8-octet random value created by the sender of the Map-Request. This nonce will be returned in the Map-Reply. The nonce is used as an index to identify the corresponding Map-Request when a Map-Reply message is received. The nonce MUST be generated by a properly seeded pseudo-random source, see as an example [RFC4086].

Source-EID-AFI: This is the address family of the 'Source EID Address' field.

Source EID Address: This is the EID of the source host that originated the packet that caused the Map-Request. When Map-Requests are used for refreshing a Map-Cache entry or for RLOC-Probing, an AFI value 0 is used and this field is of zero length.

ITR-RLOC-AFI: This is the address family of the 'ITR-RLOC Address' field that follows this field.

ITR-RLOC Address: This is used to give the ETR the option of selecting the destination address from any address family for the Map-Reply message. This address MUST be a routable RLOC address of the sender of the Map-Request message.

EID mask-len: This is the mask length for the EID-Prefix.

EID-Prefix-AFI: This is the address family of the EID-Prefix according to [AFI] and [RFC8060].

EID-Prefix: This prefix address length is 4 octets for an IPv4 address family and 16 octets for an IPv6 address family when the EID-Prefix-AFI is 1 or 2, respectively. For other AFIs [AFI], the address length varies and for the LCAF AFI the format is defined in [RFC8060]. When a Map-Request is sent by an ITR because a data packet is received for a destination where there is no mapping entry, the EID-Prefix is set to the destination IP address of the data packet, and the 'EID mask-len' is set to 32 or 128 for IPv4 or IPv6, respectively. When an xTR wants to query a site about the status of a mapping it already has cached, the EID-Prefix used in the Map-Request has the same mask-length as the EID-Prefix returned from the site when it sent a Map-Reply message.

Map-Reply Record: When the M-bit is set, this field is the size of a single "Record" in the Map-Reply format. This Map-Reply record contains the EID-to-RLOC mapping entry associated with the Source EID. This allows the ETR that will receive this Map-Request to cache the data if it chooses to do so. It is important to note that this mapping has not been validated by the Mapping System.

5.3. EID-to-RLOC UDP Map-Request Message

A Map-Request is sent from an ITR when it needs a mapping for an EID, wants to test an RLOC for reachability, or wants to refresh a mapping before TTL expiration. For the initial case, the destination IP address used for the Map-Request is the data packet's destination address (i.e., the destination EID) that had a mapping cache lookup failure. For the latter two cases, the destination IP address used for the Map-Request is one of the RLOC addresses from the Locator-Set of the Map-Cache entry. The source address is either an IPv4 or IPv6 RLOC address, depending on whether the Map-Request is using an IPv4 or IPv6 header, respectively. In all cases, the UDP source port number for the Map-Request message is a 16-bit value selected by the ITR/PITR, and the UDP destination port number is set to the well-known destination port number 4342. A successful Map-Reply, which is one that has a nonce that matches an outstanding Map-Request nonce, will update the cached set of RLOCs associated with the EID-Prefix range.

One or more Map-Request ('ITR-RLOC-AFI', 'ITR-RLOC-Address') fields MUST be filled in by the ITR. The number of fields (minus 1) encoded MUST be placed in the 'IRC' field. The ITR MAY include all locally configured Locators in this list or just provide one locator address from each address family it supports. If the ITR erroneously provides no ITR-RLOC addresses, the Map-Replier MUST drop the Map-Request.

Map-Requests can also be LISP encapsulated using UDP destination port 4342 with a LISP Type value set to "Encapsulated Control Message", when sent from an ITR to a Map-Resolver. Likewise, Map-Requests are LISP encapsulated the same way from a Map-Server to an ETR. Details on Encapsulated Map-Requests and Map-Resolvers can be found in Section 5.8.

Map-Requests MUST be rate-limited to 1 per second per EID-prefix. After 10 retransmits without receiving the corresponding Map-Reply the sender MUST wait 30 seconds.

An ITR that is configured with mapping database information (i.e., it is also an ETR) MAY optionally include those mappings in a Map-Request. When an ETR configured to accept and verify such "piggybacked" mapping data receives such a Map-Request and it does not have this mapping in the Map-Cache, it MUST originate a "verifying Map-Request" through the mapping database to validate thge "piggybacked" mapping data.

5.4. Map-Reply Message Format

```
        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
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |Type=2 |P|E|S|          Reserved               | Record Count  |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         Nonce . . .                           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         . . . Nonce                           |
   +-> +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   |                          Record TTL                           |
   |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   R   | Locator Count | EID mask-len  | ACT |A|       Reserved        |
   e   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   c   | Rsvd  |  Map-Version Number   |       EID-Prefix-AFI          |
   o   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   r   |                          EID-Prefix                           |
   d   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  /|    Priority   |    Weight     |  M Priority   |   M Weight    |
   | L +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | o |        Unused Flags     |L|p|R|           Loc-AFI             |
   | c +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  \|                             Locator                           |
   +-> +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Packet field descriptions:

Type: 2 (Map-Reply)

P: This is the probe-bit, which indicates that the Map-Reply is in response to a Locator reachability probe Map-Request. The 'Nonce' field must contain a copy of the nonce value from the original Map-Request. See RLOC-probing Section 7.1 for more details. When the probe-bit is set to 1 in a Map-Reply message, the A-bit in each EID-record included in the message MUST be set to 1, otherwise MUST be silently discarded.

E: This bit indicates that the ETR that sends this Map-Reply message is advertising that the site is enabled for the Echo-Nonce Locator reachability algorithm. See Echo-Nonce [I-D.ietf-lisp-rfc6830bis] for more details.

S: This is the Security bit. When set to 1, the following authentication information will be appended to the end of the Map-Reply. The details can be found in [I-D.ietf-lisp-sec].

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    AD Type    |       Authentication Data Content . . .       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Reserved: This unassigned field MUST be set to 0 on transmit and MUST be ignored on receipt.

Record Count: This is the number of records in this reply message. A record is comprised of that portion of the packet labeled 'Record' above and occurs the number of times equal to Record Count. Note that the reply count can be larger than the requested count, for instance when more-specifics are present.

Nonce: This 64-bit value from the Map-Request is echoed in this 'Nonce' field of the Map-Reply.

Record TTL: This is the time in minutes the recipient of the Map-Reply can store the mapping. If the TTL is 0, the entry MUST be removed from the cache immediately. If the value is 0xffffffff, the recipient can decide locally how long to store the mapping.

Locator Count: This is the number of Locator entries in the given Record. A Locator entry comprises what is labeled above as 'Loc'. The Locator count can be 0, indicating that there are no Locators for the EID-Prefix.

EID mask-len: This is the mask length for the EID-Prefix.

ACT: This 3-bit field describes Negative Map-Reply actions. In any other message type, these bits are set to 0 and ignored on receipt. These bits are used only when the 'Locator Count' field is set to 0. The action bits are encoded only in Map-Reply messages. They are used to tell an ITR or PITR why a empty locator-set was returned from the mapping system and how it stores the map-cache entry. See Section 12.3 for additional information.

(0) No-Action: The Map-Cache is kept alive, and no packet encapsulation occurs.

(1) Natively-Forward: The packet is not encapsulated or dropped but natively forwarded.

(2) Send-Map-Request: The Map-Cache entry is created and flagged that any packet matching this entry invokes sending a Map-Request.

(3) Drop/No-Reason: A packet that matches this Map-Cache entry is dropped. An ICMP Destination Unreachable message SHOULD be sent.

(4) Drop/Policy-Denied: A packet that matches this Map-Cache entry is dropped. The reason for the Drop action is that a Map-Request for the target-EID is being policy denied by either an xTR or the mapping system.

(5) Drop/Authentication-Failure: A packet that matches this Map-Cache entry is dropped. The reason for the Drop action is that a Map-Request for the target-EID fails an authentication verification-check by either an xTR or the mapping system.

A: The Authoritative bit MAY only be set to 1 by an ETR. A Map-Server generating Map-Reply messages as a proxy MUST NOT set the A-bit to 1. This bit indicates to the requesting ITRs if the Map-Reply was originated by a LISP node managed at the site that owns the EID-Prefix.

Map-Version Number: When this 12-bit value is non-zero, the Map-Reply sender is informing the ITR what the version number is for the EID record contained in the Map-Reply. The ETR can allocate this number internally but MUST coordinate this value with other ETRs for the site. When this value is 0, there is no versioning information conveyed. The Map-Version Number can be included in Map-Request and Map-Register messages. See Map-Versioning [I-D.ietf-lisp-6834bis] for more details.

EID-Prefix-AFI: Address family of the EID-Prefix according to [AFI] and [RFC8060].

EID-Prefix: This prefix is 4 octets for an IPv4 address family and 16 octets for an IPv6 address family.

Priority: Each RLOC is assigned a unicast Priority. Lower values are more preferable. When multiple RLOCs have the same Priority, they may be used in a load-split fashion. A value of 255 means the RLOC MUST NOT be used for unicast forwarding.

Weight: When priorities are the same for multiple RLOCs, the Weight indicates how to balance unicast traffic between them. Weight is encoded as a relative weight of total unicast packets that match the mapping entry. For example, if there are 4 Locators in a

Locator-Set, where the Weights assigned are 30, 20, 20, and 10, the first Locator will get 37.5% of the traffic, the 2nd and 3rd Locators will each get 25% of the traffic, and the 4th Locator will get 12.5% of the traffic. If all Weights for a Locator-Set are equal, the receiver of the Map-Reply will decide how to load-split the traffic. See RLOC-hashing [I-D.ietf-lisp-rfc6830bis] for a suggested hash algorithm to distribute the load across Locators with the same Priority and equal Weight values.

M Priority: Each RLOC is assigned a multicast Priority used by an ETR in a receiver multicast site to select an ITR in a source multicast site for building multicast distribution trees. A value of 255 means the RLOC MUST NOT be used for joining a multicast distribution tree. For more details, see [RFC6831].

M Weight: When priorities are the same for multiple RLOCs, the Weight indicates how to balance building multicast distribution trees across multiple ITRs. The Weight is encoded as a relative weight (similar to the unicast Weights) of the total number of trees built to the source site identified by the EID-Prefix. If all Weights for a Locator-Set are equal, the receiver of the Map-Reply will decide how to distribute multicast state across ITRs. For more details, see [RFC6831].

Unused Flags: These are set to 0 when sending and ignored on receipt.

L: When this bit is set, the Locator is flagged as a local Locator to the ETR that is sending the Map-Reply. When a Map-Server is doing proxy Map-Replying for a LISP site, the L-bit is set to 0 for all Locators in this Locator-Set.

p: When this bit is set, an ETR informs the RLOC-Probing ITR that the locator address for which this bit is set is the one being RLOC-probed and may be different from the source address of the Map-Reply. An ITR that RLOC-probes a particular Locator MUST use this Locator for retrieving the data structure used to store the fact that the Locator is reachable. The p-bit is set for a single Locator in the same Locator-Set. If an implementation sets more than one p-bit erroneously, the receiver of the Map-Reply MUST select the first set p-bit Locator. The p-bit MUST NOT be set for Locator-Set records sent in Map-Request and Map-Register messages.

R: This is set when the sender of a Map-Reply has a route to the Locator in the Locator data record. This receiver may find this useful to know if the Locator is up but not necessarily reachable from the receiver's point of view.

Locator: This is an IPv4 or IPv6 address (as encoded by the 'Loc-AFI' field) assigned to an ETR and used by an ITR as a destination RLOC address in the outer header of a LISP encapsulated packet. Note that the destination RLOC address of a LISP encapsulated packet MAY be an anycast address. A source RLOC of a LISP encapsulated packet can be an anycast address as well. The source or destination RLOC MUST NOT be the broadcast address (255.255.255.255 or any subnet broadcast address known to the router) and MUST NOT be a link-local multicast address. The source RLOC MUST NOT be a multicast address. The destination RLOC SHOULD be a multicast address if it is being mapped from a multicast destination EID.

Map-Reply MUST be rate-limited, it is RECOMMENDED that a Map-Reply for the same destination RLOC be sent no more than one packets per 3 seconds.

The Record format, as defined here, is used both in the Map-Reply and Map-Register messages, this includes all the field definitions.

5.5. EID-to-RLOC UDP Map-Reply Message

A Map-Reply returns an EID-Prefix with a mask-length that is less than or equal to the EID being requested. The EID being requested is either from the destination field of an IP header of a Data-Probe or the EID of a record of a Map-Request. The RLOCs in the Map-Reply are routable IP addresses of all ETRs for the LISP site. Each RLOC conveys status reachability but does not convey path reachability from a requester's perspective. Separate testing of path reachability is required. See RLOC-reachability Section 7.1 for details.

Note that a Map-Reply MAY contain different EID-Prefix granularity (prefix + mask-length) than the Map-Request that triggers it. This might occur if a Map-Request were for a prefix that had been returned by an earlier Map-Reply. In such a case, the requester updates its cache with the new prefix information and granularity. For example, a requester with two cached EID-Prefixes that are covered by a Map-Reply containing one less-specific prefix replaces the entry with the less-specific EID-Prefix. Note that the reverse, replacement of one less-specific prefix with multiple more-specific prefixes, can also occur, not by removing the less-specific prefix but rather by adding the more-specific prefixes that, during a lookup, will override the less-specific prefix.

When an EID moves out of a LISP site [I-D.ietf-lisp-eid-mobility], the database mapping system may have overlapping EID-prefixes. Or when a LISP site is configured with multiple sets of ETRs that support different EID-prefix mask-lengths, the database mapping system may have overlapping EID-prefixes. When overlapping EID-prefixes exist, a Map-Request with an EID that best matches any EID-Prefix MUST be returned in a single Map-Reply message. For instance, if an ETR had database mapping entries for EID-Prefixes:

```
2001:db8::/32
2001:db8:1::/48
2001:db8:1:1::/64
2001:db8:1:2::/64
```

A Map-Request for EID 2001:db8:1:1::1 would cause a Map-Reply with a record count of 1 to be returned with a mapping record EID-Prefix of 2001:db8:1:1::/64.

A Map-Request for EID 2001:db8:1:5::5 would cause a Map-Reply with a record count of 3 to be returned with mapping records for EID-Prefixes 2001:db8:1::/48, 2001:db8:1:1::/64, 2001:db8:1:2::/64, filling out the /48 with more-specifics that exist in the mapping system.

Note that not all overlapping EID-Prefixes need to be returned but only the more-specific entries (note that in the second example above 2001:db8::/32 was not returned for requesting EID 2001:db8:1:5::5) for the matching EID-Prefix of the requesting EID. When more than one EID-Prefix is returned, all SHOULD use the same Time to Live value so they can all time out at the same time. When a more-specific EID-Prefix is received later, its Time to Live value in the Map-Reply record can be stored even when other less-specific entries exist. When a less-specific EID-Prefix is received later, its Map-Cache expiration time SHOULD be set to the minimum expiration time of any more-specific EID-Prefix in the Map-Cache. This is done so the integrity of the EID-Prefix set is wholly maintained and so no more-specific entries are removed from the Map-Cache while keeping less-specific entries.

For scalability, it is expected that aggregation of EID addresses into EID-Prefixes will allow one Map-Reply to satisfy a mapping for the EID addresses in the prefix range, thereby reducing the number of Map-Request messages.

Map-Reply records can have an empty Locator-Set. A Negative Map-Reply is a Map-Reply with an empty Locator-Set. Negative Map-Replies convey special actions by the sender to the ITR or PITR that have solicited the Map-Reply. There are two primary applications for Negative Map-Replies. The first is for a Map-Resolver to instruct an ITR or PITR when a destination is for a LISP site versus a non-LISP site, and the other is to source quench Map-Requests that are sent for non-allocated EIDs.

For each Map-Reply record, the list of Locators in a Locator-Set MUST be sorted in order of ascending IP address where an IPv4 locator address is considered numerically 'less than' an IPv6 locator address.

When sending a Map-Reply message, the destination address is copied from one of the 'ITR-RLOC' fields from the Map-Request. The ETR can choose a locator address from one of the address families it supports. For Data-Probes, the destination address of the Map-Reply is copied from the source address of the Data-Probe message that is invoking the reply. The source address of the Map-Reply is one of the local IP addresses chosen, to allow Unicast Reverse Path Forwarding (uRPF) checks to succeed in the upstream service provider. The destination port of a Map-Reply message is copied from the source port of the Map-Request or Data-Probe, and the source port of the Map-Reply message is set to the well-known UDP port 4342.

5.6. Map-Register Message Format

This section specifies the encoding format for the Map-Register message. The message is sent in UDP with a destination UDP port of 4342 and a randomly selected UDP source port number.

The fields below are used in multiple control messages. They are defined for Map-Register, Map-Notify and Map-Notify-Ack message types.

The Map-Register message format is:

```
        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
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |Type=3 |P|S|I|        Reserved       |E|T|a|R|M| Record Count  |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         Nonce . . .                           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         . . . Nonce                           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |    Key ID     |  Algorithm ID  |  Authentication Data Length  |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       ~                     Authentication Data                       ~
   +-> +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   |                          Record TTL                           |
   |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   R   | Locator Count | EID mask-len  | ACT |A|       Reserved        |
   e   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   c   | Rsvd  |  Map-Version Number   |        EID-Prefix-AFI         |
   o   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   r   |                          EID-Prefix                           |
   d   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  /|    Priority   |    Weight     |  M Priority   |   M Weight    |
   | L +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | o |        Unused Flags     |L|p|R|           Loc-AFI             |
   | c +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  \|                             Locator                           |
   +-> +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Packet field descriptions:

Type: 3 (Map-Register)

P: This is the proxy Map-Reply bit. When set to 1, the ETR sending the Map-Register message is requesting the Map-Server to proxy a Map-Reply. The Map-Server will send non-authoritative Map-Replies on behalf of the ETR.

S: This is the security-capable bit. When set, the procedures from [I-D.ietf-lisp-sec] are supported.

I: This is the ID-present bit. This bit is set to 1 to indicate that a 128 bit xTR-ID and a 64 bit Site-ID fields are present at the end of the Map-Register message. If an xTR is configured with an xTR-ID and Site-ID, it MUST set the I bit to 1 and include its xTR-ID and Site-ID in the Map-Register messages it generates. The combination of Site-ID plus xTR-ID uniquely identifies an xTR in a LISP domain and serves to track its last seen nonce.

Reserved: This unassigned field MUST be set to 0 on transmit and MUST be ignored on receipt.

E: This is the Map-Register EID-notify bit. This is used by a First-Hop-Router (FHR) which discovers a dynamic-EID. This EID-notify based Map-Register is sent by the FHR to a same site xTR that propogates the Map-Register to the mapping system. The site xTR keeps state to later Map-Notify the FHR after the EID has moves away. See [I-D.ietf-lisp-eid-mobility] for a detailed use-case.

T: This is the use-TTL for timeout bit. When set to 1, the xTR wants the Map-Server to time out registrations based on the value in the "Record TTL" field of this message. Otherwise, the default timeout described in Section 8.2 is used.

a: This is the merge-request bit. When set to 1, the xTR requests to merge RLOC-records from different xTRs registering the same EID-record. See signal-free multicast [RFC8378] for one use case example.

R: This reserved and unassigned bit MUST be set to 0 on transmit and MUST be ignored on receipt.

M: This is the want-map-notify bit. When set to 1, an ETR is requesting a Map-Notify message to be returned in response to sending a Map-Register message. The Map-Notify message sent by a Map-Server is used to acknowledge receipt of a Map-Register message.

Record Count: This is the number of records in this Map-Register message. A record is comprised of that portion of the packet labeled 'Record' above and occurs the number of times equal to Record Count.

Nonce: This 8-octet 'Nonce' field is incremented each time a Map-Register message is sent. When a Map-Register acknowledgement is requested, the nonce is returned by Map-Servers in Map-Notify messages. Since the entire Map-Register message is authenticated, the 'Nonce' field serves to protect against Map-Register replay attacks. An ETR that registers to the mapping system SHOULD store the last nonce sent in persistent storage so when it restarts it can continue using an incrementing nonce. If the ETR cannot support saving the nonce, then when it restarts it MUST use a new authentication key to register to the mapping system. A Map-Server MUST track and save in persistent storage the last nonce received for each ETR xTR-ID and key pair. If a Map-Register is received with a nonce value that is not greater than the saved nonce, it MUST drop the Map-Register message and SHOULD log the fact a replay attack could have occurred.

Key ID: A key-id value that identifies a pre-shared secret between an ETR and a Map-Server. Per-message keys are derived from the pre-shared secret to authenticate the origin and protect the integrity of the Map-Register. The Key ID allows to rotate between multiple pre-shared secrets in a non disruptive way. The pre-shared secret MUST be unique per each LISP "Site-ID"

Algorithm ID: This field identifies the Key Derivation Function (KDF) and Message Authentication Code (MAC) algorithms used to derive the key and to compute the Authentication Data of a Map-Register. This 8-bit field identifies the KDF and MAC algorithm pair. See Section 12.5 for codepoint assignments.

Authentication Data Length: This is the length in octets of the 'Authentication Data' field that follows this field. The length of the 'Authentication Data' field is dependent on the MAC algorithm used. The length field allows a device that doesn't know the MAC algorithm to correctly parse the packet.

Authentication Data: This is the output of the MAC algorithm placed in this field after the MAC computation. The MAC output is computed as follows:

1: The KDF algorithm is identified by the field 'Algorithm ID' according to the table in Section 12.5. Implementations of this specification MUST implement HMAC-SHA-256-128 [RFC4868] and SHOULD implement HMAC-SHA-256-128+HKDF-SHA256 [RFC5869] .

2: The MAC algorithm is identified by the field 'Algorithm ID' according to the table in Section 12.5.

3: The pre-shared secret used to derive the per-message key is represented by PSK[Key ID], that is the pre-shared secret identified by the 'Key ID'.

4: The derived per-message key is computed as: per-msg-key=KDF(nonce+PSK[Key ID],s). Where the nonce is the value in the Nonce field of the Map-Register, '+' denotes concatenation and 's' (the salt) is a string that corresponds to the message type being authenticated. For Map-Register messages, it is equal to "Map-Register Authentication". Similarly, for Map-Notify and Map-Notify-Ack messages, it is "Map-Notify Authentication" and "Map-Notify-Ack Authentication", respectively. For those Algorithm IDs defined in Section 12.5 that specify a 'none' KDF, the per-message key is computed as: per-msg-key = PSK[Key ID]. This means that the same key is used across multiple protocol messages.

5: The MAC output is computed using the MAC algorithm and the per-msg-key over the entire Map-Register payload (from and including the LISP message type field through the end of the last RLOC record) with the authenticated data field preset to 0.

The definition of the rest of the Map-Register can be found in EID-record description in Section 5.4. When the I-bit is set, the following fields are added to the end of the Map-Register message:

xTR-ID: xTR-ID is a 128 bit field at the end of the Map-Register message, starting after the final Record in the message. The xTR-ID is used to uniquely identify a xTR. The same xTR-ID value MUST NOT be used in two different xTRs in the scope of the Site-ID.

Site-ID: Site-ID is a 64 bit field at the end of the Map- Register message, following the xTR-ID. Site-ID is used to uniquely identify to which site the xTR that sent the message belongs. This document does not specify a strict meaning for the Site-ID field. Informally it provides an indication that a group of xTRs have some relation, either administratively, topologically or otherwise.

5.7. Map-Notify/Map-Notify-Ack Message Format

This section specifies the encoding format for the Map-Notify and Map-Notify-Ack messages. The messages are sent inside a UDP packet with source and destination UDP ports equal to 4342.

The Map-Notify and Map-Notify-Ack message formats are:

```
        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
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |Type=4/5|             Reserved               | Record Count  |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         Nonce . . .                           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         . . . Nonce                           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |    Key ID     |  Algorithm ID |  Authentication Data Length   |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       ~                     Authentication Data                       ~
   +-> +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   |                          Record TTL                           |
   |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   R   | Locator Count | EID mask-len  | ACT |A|       Reserved        |
   e   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   c   | Rsvd  |  Map-Version Number   |        EID-Prefix-AFI         |
   o   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   r   |                          EID-Prefix                           |
   d   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  /|    Priority   |    Weight     |  M Priority   |   M Weight    |
   | L +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | o |        Unused Flags     |L|p|R|           Loc-AFI             |
   | c +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  \|                             Locator                           |
   +-> +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Packet field descriptions:

Type: 4/5 (Map-Notify/Map-Notify-Ack)

The Map-Notify message has the same contents as a Map-Register message. See the Map-Register section for field descriptions and the Map-Reply section for EID-record and RLOC-record descriptions.

The fields of the Map-Notify are copied from the corresponding Map-Register to acknowledge its correct processing. In the Map-Notfiy, the 'Authentication Data' field is recomputed using the corresponding per-message key and according to the procedure defined in the previous section. The Map-Notify message can also used, outside the scope of this specification, in an unsolicited manner, such as is specified in [I-D.ietf-lisp-pubsub].

After sending a Map-Register, if a Map-Notify is not received after 1 second the transmitter MUST re-transmit the original Map-Register with an exponential backoff (base of 2, that is, the next backoff timeout interval is doubled), the maximum backoff is 1 minute. Map-Notify messages are only transmitted upon the reception of a Map-Register with the M-bit set, Map-Notify messages are not retransmitted. The only exeption to this is for unsolicited Map-Notify messages, see below.

A Map-Server sends an unsolicited Map-Notify message (one that is not used as an acknowledgment to a Map-Register message) only in conformance with the Congestion Control And Relability Guideline sections of [RFC8085]. A Map-Notify is retransmitted until a Map-Notify-Ack is received by the Map-Server with the same nonce used in the Map-Notify message. An implementation SHOULD retransmit up to 3 times at 3 second retransmission intervals, after which time the retransmission interval is exponentially backed-off (base of 2, that is, the next backoff timeout interval is doubled) for another 3 retransmission attempts. Map-Notify-Ack messages are only transmitted upon the reception of an unsolicited Map-Notify, Map-Notify-Ack messages are not retransmitted.

The Map-Notify-Ack message has the same contents as a Map-Notify message. It is used to acknowledge the receipt of an unsolicited Map-Notify and, once the the authentication data is validated, allows for the sender to stop retransmitting a Map-Notify with the same nonce and the authentication data validates. The fields of the Map-Notify-Ack are copied from the corresponding Map-Notify message to acknowledge its correct processing. The 'Authentication Data' field is recomputed using the corresponding per-message key and according to the procedure defined in the previous section.

Upon reception of Map-Register, Map-Notify or Map-Notifiy-Ack, the receiver verifies the authentication data. If the authentication data fails to validate, the message is dropped without further processing.

5.8. Encapsulated Control Message Format

An Encapsulated Control Message (ECM) is used to encapsulate control packets sent between xTRs and the mapping database system or internal to the mapping database system.

```
        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
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |                       IPv4 or IPv6 Header                     |
   OH  |                      (uses RLOC addresses)                    |
     \ |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |       Source Port = xxxx      |       Dest Port = 4342        |
   UDP +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     \ |           UDP Length          |        UDP Checksum           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  LISP |Type=8 |S|D|R|R|            Reserved                           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |                       IPv4 or IPv6 Header                     |
   IH  |                  (uses RLOC or EID addresses)                 |
     \ |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |       Source Port = xxxx      |       Dest Port = yyyy        |
   UDP +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     \ |           UDP Length          |        UDP Checksum           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   LCM |                      LISP Control Message                     |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Packet header descriptions:

OH: The outer IPv4 or IPv6 header, which uses RLOC addresses in the source and destination header address fields.

UDP: The outer UDP header with destination port 4342. The source port is randomly allocated. The checksum field MUST be non-zero.

LISP: Type 8 is defined to be a "LISP Encapsulated Control Message", and what follows is either an IPv4 or IPv6 header as encoded by the first 4 bits after the 'Reserved' field, or the Authentication Data field [I-D.ietf-lisp-sec] if the S-bit (see below) is set.

Type: 8 (Encapsulated Control Message (ECM))

S: This is the Security bit. When set to 1, the field following the 'Reserved' field will have the following Authentication Data format and follow the procedures from [I-D.ietf-lisp-sec].

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    AD Type    |       Authentication Data Content . . .       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

D: This is the DDT-bit. When set to 1, the sender is requesting a Map-Referral message to be returned. The details of this procedure are described in [RFC8111].

R: This reserved and unassigned bit MUST be set to 0 on transmit and MUST be ignored on receipt.

IH: The inner IPv4 or IPv6 header, which can use either RLOC or EID addresses in the header address fields. When a Map-Request is encapsulated in this packet format, the destination address in this header is an EID.

UDP: The inner UDP header, where the port assignments depend on the control packet being encapsulated. When the control packet is a Map-Request or Map-Register, the source port is selected by the ITR/PITR and the destination port is 4342. When the control packet is a Map-Reply, the source port is 4342 and the destination port is assigned from the source port of the invoking Map-Request. Port number 4341 MUST NOT be assigned to either port. The checksum field MUST be non-zero.

LCM: The format is one of the control message formats described in Section 5. Map-Request messages are allowed to be Control-Plane (ECM) encapsulated. When Map-Requests are sent for RLOC-Probing purposes (i.e. the probe-bit is set), they MUST NOT be sent inside Encapsulated Control Messages. PIM Join/Prune messages [RFC6831] are also allowed to be Control-Plane (ECM) encapsulated.

6. Changing the Contents of EID-to-RLOC Mappings

In the LISP architecture ITRs/PITRs use a local Map-Cache to store EID-to-RLOC mappings for forwarding. When an ETR updates a mapping a mechanism is required to inform ITRs/PITRs that are using such mappings.

The LISP Data-Plane defines several mechanism to update mappings [I-D.ietf-lisp-rfc6830bis]. This document specifies the Solicit-Map Request (SMR), a Control-Plane push-based mechanism. An additional Control-Plane mechanism based on the Publish/subscribe paradigm is specified in [I-D.ietf-lisp-pubsub].

6.1. Solicit-Map-Request (SMR)

Soliciting a Map-Request is a selective way for ETRs, at the site where mappings change, to control the rate they receive requests for Map-Reply messages. SMRs are also used to tell remote ITRs to update the mappings they have cached.

Since ETRs are not required to keep track of remote ITRs that have cached their mappings, they do not know which ITRs need to have their mappings updated. As a result, an ETR will solicit Map-Requests to those sites to which it has been sending LISP encapsulated data packets for the last minute. As a result, when an ETR is also acting as ITR, it will send an SMR to an ITR to which it has recently sent encapsulated data.

An SMR message is simply a bit set in a Map-Request message. An ITR or PITR will send a Map-Request (SMR-invoked Map-Request) when they receive an SMR message. While the SMR message is sent through the data-plane, the SMR-invoked Map-Request MUST be sent through the Mapping System (not directly).

Both the SMR sender and the SMR responder MUST rate-limit these messages. It is RECOMMENDED that the SMR sender rate-limits Map-Request for the same destination RLOC to no more than one packet per 3 seconds. It is RECOMMENDED that the SMR responder rate-limits Map-Request for the same EID-Prefix to no more than once per 3 seconds.

When an ITR receives an SMR message for which it does not have a cached mapping for the EID in the SMR message, it SHOULD NOT send an SMR-invoked Map-Request. This scenario can occur when an ETR sends SMR messages to all Locators in the Locator-Set it has stored in its Map-Cache but the remote ITRs that receive the SMR may not be sending packets to the site. There is no point in updating the ITRs until they need to send, in which case they will send Map-Requests to obtain a Map-Cache entry.

7. Routing Locator Reachability

This document defines several Control-Plane mechanisms for determining RLOC reachability. Please note that additional Data-Plane reachability mechanisms are defined in [I-D.ietf-lisp-rfc6830bis].

1. An ITR may receive an ICMP Network Unreachable or Host Unreachable message for an RLOC it is using. This indicates that the RLOC is likely down. Note that trusting ICMP messages may not be desirable, but neither is ignoring them completely. Implementations are encouraged to follow current best practices in treating these conditions [I-D.ietf-opsec-icmp-filtering].

2. When an ITR participates in the routing protocol that operates in the underlay routing system, it can determine that an RLOC is down when no Routing Information Base (RIB) entry exists that matches the RLOC IP address.

3. An ITR may receive an ICMP Port Unreachable message from a destination host. This occurs if an ITR attempts to use interworking [RFC6832] and LISP-encapsulated data is sent to a non-LISP-capable site.

4. An ITR may receive a Map-Reply from an ETR in response to a previously sent Map-Request. The RLOC source of the Map-Reply is likely up, since the ETR was able to send the Map-Reply to the ITR. Please note that in some scenarios the RLOC -from the outer header- can be an spoofable field.

5. An ITR/ETR pair can use the 'RLOC-Probing' mechanism described below.

When ITRs receive ICMP Network Unreachable or Host Unreachable messages as a method to determine unreachability, they will refrain from using Locators that are described in Locator lists of Map-Replies. However, using this approach is unreliable because many network operators turn off generation of ICMP Destination Unreachable messages.

If an ITR does receive an ICMP Network Unreachable or Host Unreachable message, it MAY originate its own ICMP Destination Unreachable message destined for the host that originated the data packet the ITR encapsulated.

This assumption does create a dependency: Locator unreachability is detected by the receipt of ICMP Host Unreachable messages. When a Locator has been determined to be unreachable, it is not used for active traffic; this is the same as if it were listed in a Map-Reply with Priority 255.

The ITR can test the reachability of the unreachable Locator by sending periodic Map-Requests. Both Map-Requests and Map-Replies MUST be rate-limited, see Section 5.3 and Section 5.4 for information about rate-limiting. Locator reachability testing is never done with data packets, since that increases the risk of packet loss for end-to-end sessions.

7.1. RLOC-Probing Algorithm

RLOC-Probing is a method that an ITR or PITR can use to determine the reachability status of one or more Locators that it has cached in a Map-Cache entry. The probe-bit of the Map-Request and Map-Reply messages is used for RLOC-Probing.

RLOC-Probing is done in the control plane on a timer basis, where an ITR or PITR will originate a Map-Request destined to a locator address from one of its own locator addresses. A Map-Request used as an RLOC-probe is NOT encapsulated and NOT sent to a Map-Server or to the mapping database system as one would when requesting mapping data. The EID record encoded in the Map-Request is the EID-Prefix of the Map-Cache entry cached by the ITR or PITR. The ITR MAY include a mapping data record for its own database mapping information that contains the local EID-Prefixes and RLOCs for its site. RLOC-probes are sent periodically using a jittered timer interval.

When an ETR receives a Map-Request message with the probe-bit set, it returns a Map-Reply with the probe-bit set. The source address of the Map-Reply is set to the IP address of the outgoing interface the Map-Reply destination address routes to. The Map-Reply SHOULD contain mapping data for the EID-Prefix contained in the Map-Request. This provides the opportunity for the ITR or PITR that sent the RLOC-probe to get mapping updates if there were changes to the ETR's database mapping entries.

There are advantages and disadvantages of RLOC-Probing. The main benefit of RLOC-Probing is that it can handle many failure scenarios allowing the ITR to determine when the path to a specific Locator is reachable or has become unreachable, thus providing a robust mechanism for switching to using another Locator from the cached Locator. RLOC-Probing can also provide rough Round-Trip Time (RTT) estimates between a pair of Locators, which can be useful for network management purposes as well as for selecting low delay paths. The major disadvantage of RLOC-Probing is in the number of control messages required and the amount of bandwidth used to obtain those benefits, especially if the requirement for failure detection times is very small.

8. Interactions with Other LISP Components

8.1. ITR EID-to-RLOC Mapping Resolution

An ITR is configured with one or more Map-Resolver addresses. These addresses are "Locators" (or RLOCs) and MUST be routable on the underlying core network; they MUST NOT need to be resolved through LISP EID-to-RLOC mapping, as that would introduce a circular dependency. When using a Map-Resolver, an ITR does not need to connect to any other database mapping system.

An ITR sends an Encapsulated Map-Request to a configured Map-Resolver when it needs an EID-to-RLOC mapping that is not found in its local Map-Cache. Using the Map-Resolver greatly reduces both the complexity of the ITR implementation and the costs associated with its operation.

In response to an Encapsulated Map-Request, the ITR can expect one of the following:

- An immediate Negative Map-Reply (with action code of "Natively-Forward", 15-minute Time to Live (TTL)) from the Map-Resolver if the Map-Resolver can determine that the requested EID does not exist. The ITR saves the EID-Prefix returned in the Map-Reply in its cache, marks it as non-LISP-capable, and knows not to attempt LISP encapsulation for destinations matching it.

- A Negative Map-Reply, with action code of "Natively-Forward", from a Map-Server that is authoritative (within the LISP deployment Section 1.1) for an EID-Prefix that matches the requested EID but that does not have an actively registered, more-specific EID-prefix. In this case, the requested EID is said to match a "hole" in the authoritative EID-Prefix. If the requested EID matches a more-specific EID-Prefix that has been delegated by the Map-Server but for which no ETRs are currently registered, a 1-minute TTL is returned. If the requested EID matches a non-delegated part of the authoritative EID-Prefix, then it is not a LISP EID and a 15-minute TTL is returned. See Section 8.2 for discussion of aggregate EID-Prefixes and details of Map-Server EID-Prefix matching.

- A LISP Map-Reply from the ETR that owns the EID-to-RLOC mapping or possibly from a Map-Server answering on behalf of the ETR. See Section 8.4 for more details on Map-Resolver message processing.

Note that an ITR may be configured to both use a Map-Resolver and to participate in a LISP-ALT logical network. In such a situation, the ITR SHOULD send Map-Requests through the ALT network for any EID-Prefix learned via ALT BGP. Such a configuration is expected to be very rare, since there is little benefit to using a Map-Resolver if an ITR is already using LISP-ALT. There would be, for example, no need for such an ITR to send a Map-Request to a possibly non-existent EID (and rely on Negative Map-Replies) if it can consult the ALT database to verify that an EID-Prefix is present before sending that Map-Request.

8.2. EID-Prefix Configuration and ETR Registration

An ETR publishes its EID-Prefixes on a Map-Server by sending LISP Map-Register messages. A Map-Register message includes authentication data, so prior to sending a Map-Register message, the ETR and Map-Server MUST be configured with a pre-shared secret used to derive Map-Register authentication keys. A Map-Server's configuration SHOULD also include a list of the EID-Prefixes for which each ETR is authoritative. Upon receipt of a Map-Register from an ETR, a Map-Server accepts only EID-Prefixes that are configured for that ETR. Failure to implement such a check would leave the mapping system vulnerable to trivial EID-Prefix hijacking attacks.

In addition to the set of EID-Prefixes defined for each ETR that may register, a Map-Server is typically also configured with one or more aggregate prefixes that define the part of the EID numbering space assigned to it. When LISP-ALT is the database in use, aggregate EID-Prefixes are implemented as discard routes and advertised into ALT BGP. The existence of aggregate EID-Prefixes in a Map-Server's database means that it may receive Map Requests for EID-Prefixes that match an aggregate but do not match a registered prefix; Section 8.3 describes how this is handled.

Map-Register messages are sent periodically from an ETR to a Map-Server with a suggested interval between messages of one minute. A Map-Server SHOULD time out and remove an ETR's registration if it has not received a valid Map-Register message within the past three minutes. When first contacting a Map-Server after restart or changes to its EID-to-RLOC database mappings, an ETR MAY initially send Map-Register messages at an increased frequency, up to one every 20 seconds. This "quick registration" period is limited to five minutes in duration.

An ETR MAY request that a Map-Server explicitly acknowledge receipt and processing of a Map-Register message by setting the "want-map-notify" (M-bit) flag. A Map-Server that receives a Map-Register with this flag set will respond with a Map-Notify message. Typical use of this flag by an ETR would be to set it for Map-Register messages sent during the initial "quick registration" with a Map-Server but then set it only occasionally during steady-state maintenance of its association with that Map-Server. Note that the Map-Notify message is sent to UDP destination port 4342, not to the source port specified in the original Map-Register message.

Note that a one-minute minimum registration interval during maintenance of an ETR-Map-Server association places a lower bound on how quickly and how frequently a mapping database entry can be updated. This may have implications for what sorts of mobility can be supported directly by the mapping system; shorter registration intervals or other mechanisms might be needed to support faster mobility in some cases. For a discussion on one way that faster mobility may be implemented for individual devices, please see [I-D.ietf-lisp-mn].

An ETR MAY also request, by setting the "proxy Map-Reply" flag (P-bit) in the Map-Register message, that a Map-Server answer Map-Requests instead of forwarding them to the ETR. See Section 7.1 for details on how the Map-Server sets certain flags (such as those indicating whether the message is authoritative and how returned Locators SHOULD be treated) when sending a Map-Reply on behalf of an ETR. When an ETR requests proxy reply service, it SHOULD include all RLOCs for all ETRs for the EID-Prefix being registered, along with the routable flag ("R-bit") setting for each RLOC. The Map-Server includes all of this information in Map-Reply messages that it sends on behalf of the ETR. This differs from a non-proxy registration, since the latter need only provide one or more RLOCs for a Map-Server to use for forwarding Map-Requests; the registration information is not used in Map-Replies, so it being incomplete is not incorrect.

An ETR that uses a Map-Server to publish its EID-to-RLOC mappings does not need to participate further in the mapping database protocol(s). When using a LISP-ALT mapping database, for example, this means that the ETR does not need to implement GRE or BGP, which greatly simplifies its configuration and reduces its cost of operation.

Note that use of a Map-Server does not preclude an ETR from also connecting to the mapping database (i.e., it could also connect to the LISP-ALT network), but doing so doesn't seem particularly useful, as the whole purpose of using a Map-Server is to avoid the complexity of the mapping database protocols.

8.3. Map-Server Processing

Once a Map-Server has EID-Prefixes registered by its client ETRs, it can accept and process Map-Requests for them.

In response to a Map-Request, the Map-Server first checks to see if the destination EID matches a configured EID-Prefix. If there is no match, the Map-Server returns a Negative Map-Reply with action code "Natively-Forward" and a 15-minute TTL. This can occur if a Map Request is received for a configured aggregate EID-Prefix for which no more-specific EID-Prefix exists; it indicates the presence of a non-LISP "hole" in the aggregate EID-Prefix.

Next, the Map-Server checks to see if any ETRs have registered the matching EID-Prefix. If none are found, then the Map-Server returns a Negative Map-Reply with action code "Natively-Forward" and a 1-minute TTL.

If the EID-prefix is either registered or not registered to the mapping system and there is a policy in the Map-Server to have the requestor drop packets for the matching EID-prefix, then a Drop/ Policy-Denied action is returned. If the EID-prefix is registered or not registered and there is a authentication failure, then a Drop/ Authentication- failure action is returned. If either of these actions result as a temporary state in policy or authentication then a Send-Map-Request action with 1-minute TTL MAY be returned to allow the requestor to retry the Map-Request.

If any of the registered ETRs for the EID-Prefix have requested proxy reply service, then the Map-Server answers the request instead of forwarding it. It returns a Map-Reply with the EID-Prefix, RLOCs, and other information learned through the registration process.

If none of the ETRs have requested proxy reply service, then the Map-Server re-encapsulates and forwards the resulting Encapsulated Map-Request to one of the registered ETRs. It does not otherwise alter the Map-Request, so any Map-Reply sent by the ETR is returned to the RLOC in the Map-Request, not to the Map-Server. Unless also acting as a Map-Resolver, a Map-Server should never receive Map-Replies; any such messages SHOULD be discarded without response, perhaps accompanied by the logging of a diagnostic message if the rate of Map-Replies is suggestive of malicious traffic.

8.4. Map-Resolver Processing

Upon receipt of an Encapsulated Map-Request, a Map-Resolver decapsulates the enclosed message and then searches for the requested EID in its local database of mapping entries (statically configured or learned from associated ETRs if the Map-Resolver is also a Map-Server offering proxy reply service). If it finds a matching entry, it returns a LISP Map-Reply with the known mapping.

If the Map-Resolver does not have the mapping entry and if it can determine that the EID is not in the mapping database (for example, if LISP-ALT is used, the Map-Resolver will have an ALT forwarding table that covers the full EID space), it immediately returns a negative LISP Map-Reply, with action code "Natively-Forward" and a 15-minute TTL. To minimize the number of negative cache entries needed by an ITR, the Map-Resolver SHOULD return the least-specific prefix that both matches the original query and does not match any EID-Prefix known to exist in the LISP-capable infrastructure.

If the Map-Resolver does not have sufficient information to know whether the EID exists, it needs to forward the Map-Request to another device that has more information about the EID being requested. To do this, it forwards the unencapsulated Map-Request, with the original ITR RLOC as the source, to the mapping database system. Using LISP-ALT, the Map-Resolver is connected to the ALT network and sends the Map-Request to the next ALT hop learned from its ALT BGP neighbors. The Map-Resolver does not send any response to the ITR; since the source RLOC is that of the ITR, the ETR or Map-Server that receives the Map-Request over the ALT and responds will do so directly to the ITR.

8.4.1. Anycast Operation

A Map-Resolver can be set up to use "anycast", where the same address is assigned to multiple Map-Resolvers and is propagated through IGP routing, to facilitate the use of a topologically close Map-Resolver by each ITR.

ETRs MAY have anycast RLOC addresses which are registered as part of their RLOC-set to the mapping system. However, registrations MUST use their unique RLOC addresses, distinct authentication keys or different XTR-IDs to identify security associations with the Map-Servers.

9. Security Considerations

A LISP threat analysis can be found in [RFC7835]. In what follows we highlight security considerations that apply when LISP is deployed in environments such as those specified in Section 1.1, where the following assumptions hold:

1. The Mapping System is secure and trusted, and for the purpose of this security considerations the Mapping System is considered as one trusted element.

2. The ETRs have a pre-configured trust relationship with the Mapping System, which includes some form of shared secret, and the Mapping System is aware of which EIDs an ETR can advertise. How those keys and mappings gets established is out of the scope of this document.

3. LISP-SEC [I-D.ietf-lisp-sec] MUST be implemented. Network operators should carefully weight how the LISP-SEC threat model applies to their particular use case or deployment. If they decide to ignore a particular recommendation, they should make sure the risk associated with the corresponding threats is well understood.

The Map-Request/Map-Reply message exchange can be exploited by an attacker to mount DoS and/or amplification attacks. Attackers can send Map-Requests at high rates to overload LISP nodes and increase the state maintained by such nodes or consume CPU cycles. Such threats can be mitigated by systematically applying filters and rate limiters.

The Map-Request/Map-Reply message exchange can also be exploited to inject forged mappings directly in the ITR EID-to-RLOC map-cache. This can lead to traffic being redirected to the attacker, see further details in [RFC7835]. In addition, valid ETRs in the system can perform overclaiming attacks. In this case, attackers can claim to own an EID-prefix that is larger than the prefix owned by the ETR. Such attacks can be addressed by using LISP-SEC [I-D.ietf-lisp-sec]. The LISP-SEC protocol defines a mechanism for providing origin authentication, integrity protection, and prevention of 'man-in-the-middle' and 'prefix overclaiming' attacks on the Map-Request/Map-Reply exchange. In addition and while beyond the scope of securing an individual Map-Server or Map-Resolver, it should be noted that LISP-SEC can be complemented by additional security mechanisms defined by the Mapping System Infrastructure. For instance, BGP-based LISP-ALT [RFC6836] can take advantage of standards work on adding security to BGP while LISP-DDT [RFC8111] defines its own additional security mechanisms.

To publish an authoritative EID-to-RLOC mapping with a Map-Server using the Map-Register message, an ETR includes authentication data that is a MAC of the entire message using a key derived from the pre-shared secret. An implementation SHOULD support HMAC-SHA256-128+HKDF-SHA256 [RFC5869]. The Map-Register message includes protection for replay attacks by a man-in-the-middle. However, there is a potential attack where a compromised ETR could overclaim the prefix it owns and successfully register it on its corresponding Map-Server. To mitigate this and as noted in Section 8.2, a Map-Server MUST verify that all EID-Prefixes registered by an ETR match the configuration stored on the Map-Server.

Deployments concerned about manipulations of Map-Request and Map-Reply messages, and malicious ETR EID prefix overclaiming MUST drop LISP Control Plane messages that do not contain LISP-SEC material (S-bit, EID-AD, OTK-AD, PKT-AD).

Mechanisms to encrypt, support privacy, prevent eavesdropping and packet tampering for messages exchanged between xTRs, xTRs and the mapping system, and nodes that make up the mapping system, SHOULD be deployed. Examples of this are DTLS [RFC6347] or LISP-crypto [RFC8061].

10. Privacy Considerations

As noted by [RFC6973] privacy is a complex issue that greatly depends on the specific protocol use-case and deployment. As noted in section 1.1 of [I-D.ietf-lisp-rfc6830bis] LISP focuses on use-cases where entities communicate over the public Internet while keeping separate addressing and topology. In what follows we detail the privacy threats introduced by the LISP Control Plane, the analysis is based on the guidelines detailed in [RFC6973].

LISP can use long-lived identifiers (EIDs) that survive mobility events. Such identifiers bind to the RLOCs of the nodes, which represents the topological location with respect to the specific LISP deployments. In addition, EID-to-RLOC mappings are typically considered public information within the LISP deployment when control-plane messages are not encrypted, and can be eavesdropped while Map-Request messages are sent to the corresponding Map-Resolvers or Map-Register messages to Map-Servers.

In this context, attackers can correlate the EID with the RLOC and track the corresponding user topological location and/or mobility. This can be achieved by off-path attackers, if they are authenticated, by querying the mapping system. Deployments concerned about this threat can use access control-lists or stronger authentication mechanisms [I-D.ietf-lisp-ecdsa-auth] in the mapping system to make sure that only authorized users can access this information (data minimization). Use of ephemeral EIDs [I-D.ietf-lisp-eid-anonymity] to achieve anonymity is another mechanism to lessen persistency and identity tracking.

11. Changes since RFC 6833

For implementation considerations, the following major changes have been made to this document since RFC 6833 was published:

- A Map-Notify-Ack message is added in this document to provide reliability for Map-Notify messages. Any receiver of a Map-Notify message must respond with a Map-Notify-Ack message. Map-Servers who are senders of Map-Notify messages, must queue the Map-Notify contents until they receive a Map-Notify-Ack with the nonce used in the Map-Notify message. Note that implementations for Map-Notify-Ack support already exist and predate this document.

- This document is incorporating the codepoint for the Map-Referral message from the LISP-DDT specification [RFC8111] to indicate that a Map-Server must send the final Map-Referral message when it participates in the LISP-DDT mapping system procedures.

- The L" and "D" bits are added to the Map-Request message. See Section 5.3 for details.

- The "S", "I", "E", "T", "a", "R", and "M" bits are added to the Map-Register message. See Section 5.6 for details.

- The 16-bit Key-ID field of the Map-Register message has been split into a 8-bit Key-ID field and a 8-bit Algorithm-ID field.

- The nonce and the authentication data in the Map-Register message have a different behaviour, see Section 5.6 for details.

- This document adds two new Action values that are in an EID-record that appear in Map-Reply, Map-Register, Map-Notify, and Map-Notify-Ack messages. The Drop/Policy-Denied and Drop/Auth-Failure are the descriptions for the two new action values. See Section 5.4 for details.

12. IANA Considerations

This section provides guidance to the Internet Assigned Numbers Authority (IANA) regarding registration of values related to this LISP Control-Plane specification, in accordance with BCP 26 [RFC8126].

There are three namespaces (listed in the sub-sections below) in LISP that have been registered.

- LISP IANA registry allocations should not be made for purposes unrelated to LISP routing or transport protocols.

- o The following policies are used here with the meanings defined in BCP 26: "Specification Required", "IETF Review", "Experimental Use", and "First Come First Served".

12.1. LISP UDP Port Numbers

The IANA registry has allocated UDP port number 4342 for the LISP Control-Plane. IANA has updated the description for UDP port 4342 as follows:

Keyword	Port	Transport Layer	Description
lisp-control	4342	udp	LISP Control Packets

12.2. LISP Packet Type Codes

It is being requested that the IANA be authoritative for LISP Packet Type definitions and it is requested to replace the [RFC6830] registry message references with the RFC number assigned to this document.

Based on deployment experience of [RFC6830], the Map-Notify-Ack message, message type 5, was added by this document. This document requests IANA to add it to the LISP Packet Type Registry.

Name	Number	Defined in
LISP Map-Notify-Ack	5	RFC6833bis

12.3. LISP Map-Reply EID-Record Action Codes

New ACT values can be allocated through IETF review or IESG approval. Four values have already been allocated by [RFC6830]. IANA is requested to replace the [RFC6830] reference for this registry with the RFC number assigned to this document and [RFC6830]. This specification changes the name of ACT type 3 value from "Drop" to "Drop/No-Reason" as well as adding two new ACT values, the "Drop/Policy-Denied" (type 4) and "Drop/Authentication-Failure" (type 5).

Value	Action	Description	Raeference
4	Drop/Policy-Denied	A packet matching this Map-Cache entry is dropped because the target EWID is policy-denied by the xTR or the mapping system.	RFC6833bis
5	Drop/Auth-Failure	Packet matching the Map-Cache entry is dropped beacuse the Map-Request for the target EID fails an authentication check by the xTR or the mapping system.	RFC6833bis

LISP Map-Reply Action Values

In addition, LISP has a number of flag fields and reserved fields, such as the LISP header flags field [I-D.ietf-lisp-rfc6830bis]. New bits for flags in these fields can be implemented after IETF review or IESG approval, but these need not be managed by IANA.

12.4. LISP Address Type Codes

LISP Canonical Address Format (LCAF) [RFC8060] is an 8-bit field that defines LISP-specific encodings for AFI value 16387. LCAF encodings are used for specific use-cases where different address types for EID-records and RLOC-records are required.

The IANA registry "LISP Canonical Address Format (LCAF) Types" is used for LCAF types. The registry for LCAF types use the Specification Required policy [RFC8126]. Initial values for the registry as well as further information can be found in [RFC8060].

Therefore, there is no longer a need for the "LISP Address Type Codes" registry requested by [RFC6830]. This document requests to remove it.

12.5. LISP Algorithm ID Numbers

In [RFC6830], a request for a "LISP Key ID Numbers" registry was submitted. This document renames the registry to "LISP Algorithm ID Numbers" and requests the IANA to make the name change.

The following Algorithm ID values are defined by this specification as used in any packet type that references a 'Algorithm ID' field:

Name	Number	MAC	KDF
None	0	None	None
HMAC-SHA-1-96-None	1	[RFC2404]	None
HMAC-SHA-256-128-None	2	[RFC4868]	None
HMAC-SHA256-128+HKDF-SHA256	3	[RFC4868]	[RFC4868]

Number values are in the range of 0 to 255. The allocation of values is on a first come first served basis.

12.6. LISP Bit Flags

This document asks IANA to create a registry for allocation of bits in several headers of the LISP control plane, namely in the Map-Request, Map-Reply, Map-Register, Encapsulated Control Message (ECM) messages. Bit allocations are also requested for EID-records and RLOC-records. The registry created should be named "LISP Control Plane Header Bits". A sub-registry needs to be created per each message and EID-record. The name of each sub-registry is indicated below, along with its format and allocation of bits defined in this document. Any additional bits allocation, requires a specification, according with [RFC8126] policies.

Sub-Registry: Map-Request Header Bits [Section 5.2]:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Type=1 |A|M|P|S|p|s|R|R|  Rsvd   |L|D|    IRC  | Record Count  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Spec Name	IANA Name	Bit Position	Description
A	map-request-A	4	Authoritative Bit
M	map-request-M	5	Map Data Present Bit
P	map-request-P	6	RLOC-Probe Request Bit
S	map-request-S	7	Solicit Map-Request (SMR) Bit
p	map-request-p	8	Proxy-ITR Bit
s	map-request-s	9	Solicit Map-Request Invoked Bit
L	map-request-L	17	Local xTR Bit
D	map-request-D	18	Don't Map-Reply Bit

LISP Map-Request Header Bits

Sub-Registry: Map-Reply Header Bits [Section 5.4]:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Type=2 |P|E|S|          Reserved               | Record Count  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Spec Name	IANA Name	Bit Position	Description
P	map-reply-P	4	RLOC-Probe Bit
E	map-reply-E	5	Echo Nonce Capable Bit
S	map-reply-S	6	Security Bit

LISP Map-Reply Header Bits

Sub-Registry: Map-Register Header Bits [Section 5.6]:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Type=3 |P|S|I|        Reserved       |E|T|a|R|M| Record Count  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Spec Name	IANA Name	Bit Position	Description
P	map-register-P	4	Proxy Map-Reply Bit
S	map-register-S	5	LISP-SEC Capable Bit
I	map-register-I	6	xTR-ID present flag

LISP Map-Register Header Bits

Sub-Registry: Encapsulated Control Message (ECM) Header Bits [Section 5.8]:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Type=8 |S|D|E|M|            Reserved                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Spec Name	IANA Name	Bit Position	Description
S	ecm-S	4	Security Bit
D	ecm-D	5	LISP-DDT Bit
E	ecm-E	6	Forward to ETR Bit
M	ecm-M	7	Destined to Map-Server Bit

LISP Encapsulated Control Message (ECM) Header Bits

Sub-Registry: EID-Record Header Bits [Section 5.4]:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Locator Count | EID mask-len  | ACT |A|       Reserved        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Spec Name	IANA Name	Bit Position	Description
A	eid-record-A	19	Authoritative Bit

LISP EID-Record Header Bits

Sub-Registry: RLOC-Record Header Bits [Section 5.4]:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Unused Flags     |L|p|R|           Loc-AFI             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Spec Name	IANA Name	Bit Position	Description
L	rloc-record-L	13	Local RLOC Bit
p	rloc-record-p	19	RLOC-Probe Reply Bit
R	rloc-record-R	19	RLOC Reachable Bit

LISP RLOC-Record Header Bits

13. References

13.1. Normative References

[I-D.ietf-lisp-6834bis]
Iannone, L., Saucez, D., and O. Bonaventure, "Locator/ID Separation Protocol (LISP) Map-Versioning", draft-ietf-lisp-6834bis-07 (work in progress), October 2020.

[I-D.ietf-lisp-rfc6830bis]
Farinacci, D., Fuller, V., Meyer, D., Lewis, D., and A. Cabellos-Aparicio, "The Locator/ID Separation Protocol (LISP)", draft-ietf-lisp-rfc6830bis-35 (work in progress), September 2020.

[I-D.ietf-lisp-rfc8113bis]
Boucadair, M. and C. Jacquenet, "Locator/ID Separation Protocol (LISP): Shared Extension Message & IANA Registry for Packet Type Allocations", draft-ietf-lisp-rfc8113bis-03 (work in progress), January 2019.

[I-D.ietf-lisp-sec]
Maino, F., Ermagan, V., Cabellos-Aparicio, A., and D. Saucez, "LISP-Security (LISP-SEC)", draft-ietf-lisp-sec-21 (work in progress), July 2020.

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

[RFC2404] Madson, C. and R. Glenn, "The Use of HMAC-SHA-1-96 within ESP and AH", RFC 2404, DOI 10.17487/RFC2404, November 1998, <https://www.rfc-editor.org/info/rfc2404>.

[RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <https://www.rfc-editor.org/info/rfc4086>.

[RFC4868] Kelly, S. and S. Frankel, "Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec", RFC 4868, DOI 10.17487/RFC4868, May 2007, <https://www.rfc-editor.org/info/rfc4868>.

[RFC5869] Krawczyk, H. and P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)", RFC 5869, DOI 10.17487/RFC5869, May 2010, <https://www.rfc-editor.org/info/rfc5869>.

[RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <https://www.rfc-editor.org/info/rfc8085>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

13.2. Informative References

[AFI] "Address Family Identifier (AFIs)", ADDRESS FAMILY NUMBERS http://www.iana.org/assignments/address-family-numbers/address-family-numbers.xhtml?, Febuary 2007.

[GTP-3GPP] "General Packet Radio System (GPRS) Tunnelling Protocol User Plane (GTPv1-U)", TS.29.281 https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1699, January 2015.

[I-D.herbert-intarea-ila]
Herbert, T. and P. Lapukhov, "Identifier-locator addressing for IPv6", draft-herbert-intarea-ila-01 (work in progress), March 2018.

[I-D.ietf-lisp-ecdsa-auth]
Farinacci, D. and E. Nordmark, "LISP Control-Plane ECDSA Authentication and Authorization", draft-ietf-lisp-ecdsa-auth-04 (work in progress), September 2020.

[I-D.ietf-lisp-eid-anonymity]
Farinacci, D., Pillay-Esnault, P., and W. Haddad, "LISP EID Anonymity", draft-ietf-lisp-eid-anonymity-09 (work in progress), October 2020.

[I-D.ietf-lisp-eid-mobility]
Portoles-Comeras, M., Ashtaputre, V., Moreno, V., Maino, F., and D. Farinacci, "LISP L2/L3 EID Mobility Using a Unified Control Plane", draft-ietf-lisp-eid-mobility-06 (work in progress), May 2020.

[I-D.ietf-lisp-gpe]
Maino, F., Lemon, J., Agarwal, P., Lewis, D., and M. Smith, "LISP Generic Protocol Extension", draft-ietf-lisp-gpe-19 (work in progress), July 2020.

[I-D.ietf-lisp-introduction]
Cabellos-Aparicio, A. and D. Saucez, "An Architectural Introduction to the Locator/ID Separation Protocol (LISP)", draft-ietf-lisp-introduction-13 (work in progress), April 2015.

[I-D.ietf-lisp-mn]
Farinacci, D., Lewis, D., Meyer, D., and C. White, "LISP Mobile Node", draft-ietf-lisp-mn-08 (work in progress), August 2020.

[I-D.ietf-lisp-pubsub]
Rodriguez-Natal, A., Ermagan, V., Cabellos-Aparicio, A., Barkai, S., and M. Boucadair, "Publish/Subscribe Functionality for LISP", draft-ietf-lisp-pubsub-06 (work in progress), July 2020.

[I-D.ietf-nvo3-vxlan-gpe]
Maino, F., Kreeger, L., and U. Elzur, "Generic Protocol Extension for VXLAN (VXLAN-GPE)", draft-ietf-nvo3-vxlan-gpe-10 (work in progress), July 2020.

[I-D.ietf-opsec-icmp-filtering]
Gont, F., Gont, G., and C. Pignataro, "Recommendations for filtering ICMP messages", draft-ietf-opsec-icmp-filtering-04 (work in progress), July 2013.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[RFC1071] Braden, R., Borman, D., and C. Partridge, "Computing the Internet checksum", RFC 1071, DOI 10.17487/RFC1071, September 1988, <https://www.rfc-editor.org/info/rfc1071>.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <https://www.rfc-editor.org/info/rfc2104>.

[RFC2890] Dommety, G., "Key and Sequence Number Extensions to GRE", RFC 2890, DOI 10.17487/RFC2890, September 2000, <https://www.rfc-editor.org/info/rfc2890>.

[RFC4984] Meyer, D., Ed., Zhang, L., Ed., and K. Fall, Ed., "Report from the IAB Workshop on Routing and Addressing", RFC 4984, DOI 10.17487/RFC4984, September 2007, <https://www.rfc-editor.org/info/rfc4984>.

[RFC6234] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <https://www.rfc-editor.org/info/rfc6234>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

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

[RFC6831] Farinacci, D., Meyer, D., Zwiebel, J., and S. Venaas, "The Locator/ID Separation Protocol (LISP) for Multicast Environments", RFC 6831, DOI 10.17487/RFC6831, January 2013, <https://www.rfc-editor.org/info/rfc6831>.

[RFC6832] Lewis, D., Meyer, D., Farinacci, D., and V. Fuller, "Interworking between Locator/ID Separation Protocol (LISP) and Non-LISP Sites", RFC 6832, DOI 10.17487/RFC6832, January 2013, <https://www.rfc-editor.org/info/rfc6832>.

[RFC6836] Fuller, V., Farinacci, D., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol Alternative Logical Topology (LISP+ALT)", RFC 6836, DOI 10.17487/RFC6836, January 2013, <https://www.rfc-editor.org/info/rfc6836>.

[RFC6837] Lear, E., "NERD: A Not-so-novel Endpoint ID (EID) to Routing Locator (RLOC) Database", RFC 6837, DOI 10.17487/RFC6837, January 2013, <https://www.rfc-editor.org/info/rfc6837>.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <https://www.rfc-editor.org/info/rfc6973>.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <https://www.rfc-editor.org/info/rfc7348>.

[RFC7835] Saucez, D., Iannone, L., and O. Bonaventure, "Locator/ID Separation Protocol (LISP) Threat Analysis", RFC 7835, DOI 10.17487/RFC7835, April 2016, <https://www.rfc-editor.org/info/rfc7835>.

[RFC8060] Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format (LCAF)", RFC 8060, DOI 10.17487/RFC8060, February 2017, <https://www.rfc-editor.org/info/rfc8060>.

[RFC8061] Farinacci, D. and B. Weis, "Locator/ID Separation Protocol (LISP) Data-Plane Confidentiality", RFC 8061, DOI 10.17487/RFC8061, February 2017, <https://www.rfc-editor.org/info/rfc8061>.

[RFC8111] Fuller, V., Lewis, D., Ermagan, V., Jain, A., and A. Smirnov, "Locator/ID Separation Protocol Delegated Database Tree (LISP-DDT)", RFC 8111, DOI 10.17487/RFC8111, May 2017, <https://www.rfc-editor.org/info/rfc8111>.

[RFC8378] Moreno, V. and D. Farinacci, "Signal-Free Locator/ID Separation Protocol (LISP) Multicast", RFC 8378, DOI 10.17487/RFC8378, May 2018, <https://www.rfc-editor.org/info/rfc8378>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

Appendix A. Acknowledgments

The original authors would like to thank Greg Schudel, Darrel Lewis, John Zwiebel, Andrew Partan, Dave Meyer, Isidor Kouvelas, Jesper Skriver, Fabio Maino, and members of the lisp@ietf.org mailing list for their feedback and helpful suggestions.

Special thanks are due to Noel Chiappa for his extensive work and thought about caching in Map-Resolvers.

The current authors would like to give a sincere thank you to the people who help put LISP on standards track in the IETF. They include Joel Halpern, Luigi Iannone, Deborah Brungard, Fabio Maino, Scott Bradner, Kyle Rose, Takeshi Takahashi, Sarah Banks, Pete Resnick, Colin Perkins, Mirja Kuhlewind, Francis Dupont, Benjamin Kaduk, Eric Rescorla, Alvaro Retana, Alexey Melnikov, Alissa Cooper, Suresh Krishnan, Alberto Rodriguez-Natal, Vina Ermagen, Mohamed Boucadair, Brian Trammell, Sabrina Tanamal, and John Drake. The contributions they offered greatly added to the security, scale, and robustness of the LISP architecture and protocols.

Appendix B. Document Change Log

[RFC Editor: Please delete this section on publication as RFC.]

B.1. Changes to draft-ietf-lisp-rfc6833bis-26

- o Posted November 2019.
- o Fixed the required (MUST implement) authentcation algorithms.
- o Fixed a large set of minor comments and edits.

B.2. Changes to draft-ietf-lisp-rfc6833bis-25

- o Posted June 2019.
- o Added change requested by Mirja describing Record Count in an EID-record.
- o Fixed Requirements Notation section per Pete.
- o Added KDF for shared-secret
- o Specified several rate-limiters for control messages

B.3. Changes to draft-ietf-lisp-rfc6833bis-24

- Posted February 2019.
- Added suggested text from Albert that Benjamin Kaduk agreed with.
- Added suggested editorial comments from Alvaro's rewview.
- Ran document through IDnits. Fixed bugs found.

B.4. Changes to draft-ietf-lisp-rfc6833bis-23

- Posted December 2018.
- Added to Security Considerations section that deployments that care about prefix over claiming should use LISP-SEC.
- Added to Security Considerations section that DTLS or LISP-crypto be used for control-plane privacy.
- Make LISP-SEC a normative reference.
- Make it more clear where field descriptions are spec'ed when referencing to the same fields in other packet types.

B.5. Changes to draft-ietf-lisp-rfc6833bis-22

- Posted week after IETF November 2018.
- No longer need to use IPSEC for replay attacks.

B.6. Changes to draft-ietf-lisp-rfc6833bis-21

- Posted early November 2018.
- Added I-bit back in because its necessary to use for Map-Register replay attack scenarios. The Map-Server tracks the nonce per xTR-ID to detect duplicate or replayed Map-Register messages.

B.7. Changes to draft-ietf-lisp-rfc6833bis-20

- Posted late October 2018.
- Changed description about "reserved" bits to state "reserved and unassigned".
- Make it more clear how Map-Register nonce processing is performed in an ETR and Map-Server.

B.8. Changes to draft-ietf-lisp-rfc6833bis-19

- Posted mid October 2018.
- Added Fabio text to the Security Considerations section.

B.9. Changes to draft-ietf-lisp-rfc6833bis-18

- Posted mid October 2018.
- Fixed comments from Eric after more email clarity.

B.10. Changes to draft-ietf-lisp-rfc6833bis-17

- Posted early October 2018.
- Changes to reflect comments from Sep 27th Telechat.
- Added all flag bit definitions as request for allocation in IANA Considersations section.
- Added an applicability statement in section 1 to address security concerns from Telechat.
- Moved m-bit description and IANA request to draft-ietf-lisp-mn.
- Moved I-bit description and IANA request to draft-ietf-lisp-pubsub.

B.11. Changes to draft-ietf-lisp-rfc6833bis-16

- Posted Late-September 2018.
- Re-wrote Security Considerations section. Thanks Albert.
- Added Alvaro text to be more clear about IANA actions.

B.12. Changes to draft-ietf-lisp-rfc6833bis-15

- Posted mid-September 2018.
- Changes to reflect comments from Colin and Mirja.

B.13. Changes to draft-ietf-lisp-rfc6833bis-14

- Posted September 2018.

- o Changes to reflect comments from Genart, RTGarea, and Secdir reviews.

B.14. Changes to draft-ietf-lisp-rfc6833bis-13

- o Posted August 2018.

- o Final editorial changes before RFC submission for Proposed Standard.

- o Added section "Changes since RFC 6833" so implementators are informed of any changes since the last RFC publication.

B.15. Changes to draft-ietf-lisp-rfc6833bis-12

- o Posted late July 2018.

- o Moved RFC6830bis and RFC6834bis to Normative References.

B.16. Changes to draft-ietf-lisp-rfc6833bis-11

- o Posted July 2018.

- o Fixed Luigi editorial comments to ready draft for RFC status and ran through IDNITs again.

B.17. Changes to draft-ietf-lisp-rfc6833bis-10

- o Posted after LISP WG at IETF week March.

- o Move AD field encoding after S-bit in the ECM packet format description section.

- o Say more about when the new Drop actions should be sent.

B.18. Changes to draft-ietf-lisp-rfc6833bis-09

- o Posted March IETF week 2018.

- o Fixed editorial comments submitted by document shepherd Luigi Iannone.

B.19. Changes to draft-ietf-lisp-rfc6833bis-08

- o Posted March 2018.

- o Added RLOC-probing algorithm.

- o Added Solicit-Map Request algorithm.

- o Added several mechanisms (from 6830bis) regarding Routing Locator Reachability.

- o Added port 4342 to IANA Considerations section.

B.20. Changes to draft-ietf-lisp-rfc6833bis-07

- o Posted December 2017.

- o Make it more clear in a couple of places that RLOCs are used to locate ETRs more so than for Map-Server Map-Request forwarding.

- o Make it clear that "encapsualted" for a control message is an ECM based message.

- o Make it more clear what messages use source-port 4342 and which ones use destinatino-port 4342.

- o Don't make DDT references when the mapping transport system can be of any type and the referneced text is general to it.

- o Generalize text when referring to the format of an EID-prefix. Can use othe AFIs then IPv4 and IPv6.

- o Many editorial changes to clarify text.

- o Changed some "must", "should", and "may" to capitalized.

- o Added definitions for Map-Request and Map-Reply messages.

- o Ran document through IDNITs.

B.21. Changes to draft-ietf-lisp-rfc6833bis-06

- o Posted October 2017.

- o Spec the I-bit to include the xTR-ID in a Map-Request message to be consistent with the Map-Register message and to anticipate the introduction of pubsub functionality to allow Map-Requests to subscribe to RLOC-set changes.

- o Updated references for individual submissions that became working group documents.

- o Updated references for working group documents that became RFCs.

B.22. Changes to draft-ietf-lisp-rfc6833bis-05

- o Posted May 2017.

- o Update IANA Considerations section based on new requests from this document and changes from what was requested in [RFC6830].

B.23. Changes to draft-ietf-lisp-rfc6833bis-04

- o Posted May 2017.

- o Clarify how the Key-ID field is used in Map-Register and Map-Notify messages. Break the 16-bit field into a 8-bit Key-ID field and a 8-bit Algorithm-ID field.

- o Move the Control-Plane codepoints from the IANA Considerations section of RFC6830bis to the IANA Considerations section of this document.

- o In the "LISP Control Packet Type Allocations" section, indicate how message Types are IANA allocated and how experimental RFC8113 sub-types should be requested.

B.24. Changes to draft-ietf-lisp-rfc6833bis-03

- o Posted April 2017.

- o Add types 9-14 and specify they are not assigned.

- o Add the "LISP Shared Extension Message" type and point to RFC8113.

B.25. Changes to draft-ietf-lisp-rfc6833bis-02

- o Posted April 2017.

- o Clarify that the LISP Control-Plane document defines how the LISP Data-Plane uses Map-Requests with either the SMR-bit set or the P-bit set supporting mapping updates and RLOC-probing. Indicating that other Data-Planes can use the same mechanisms or their own defined mechanisms to achieve the same functionality.

B.26. Changes to draft-ietf-lisp-rfc6833bis-01

- o Posted March 2017.

- o Include references to new RFCs published.

- o Remove references to self.

- o Change references from RFC6830 to RFC6830bis.

- o Add two new action/reasons to a Map-Reply has posted to the LISP WG mailing list.

- o In intro section, add refernece to I-D.ietf-lisp-introduction.

- o Removed Open Issues section and references to "experimental".

B.27. Changes to draft-ietf-lisp-rfc6833bis-00

- o Posted December 2016.

- o Created working group document from draft-farinacci-lisp -rfc6833-00 individual submission. No other changes made.

B.28. Changes to draft-farinacci-lisp-rfc6833bis-00

- o Posted November 2016.

- o This is the initial draft to turn RFC 6833 into RFC 6833bis.

- o The document name has changed from the "Locator/ID Separation Protocol (LISP) Map-Server Interface" to the "Locator/ID Separation Protocol (LISP) Control-Plane".

- o The fundamental change was to move the Control-Plane messages from RFC 6830 to this document in an effort so any IETF developed or industry created Data-Plane could use the LISP mapping system and Control-Plane.

- o Update Control-Plane messages to incorporate what has been implemented in products during the early phase of LISP development but wasn't able to make it into RFC6830 and RFC6833 to make the Experimental RFC deadline.

- o Indicate there may be nodes in the mapping system that are not MRs or MSs, that is a ALT-node or a DDT-node.

- o Include LISP-DDT in Map-Resolver section and explain how they maintain a referral-cache.

- o Removed open issue about additional state in Map-Servers. With [RFC8111], Map-Servers have the same registration state and can give Map-Resolvers complete information in ms-ack Map-Referral messages.

- o Make reference to the LISP Threats Analysis RFC [RFC7835].

Authors' Addresses

Dino Farinacci
lispers.net

EMail: farinacci@gmail.com

Fabio Maino
Cisco Systems

EMail: fmaino@cisco.com

Vince Fuller
vaf.net Internet Consulting

EMail: vaf@vaf.net

Albert Cabellos
UPC/BarcelonaTech
Campus Nord, C. Jordi Girona 1-3
Barcelona, Catalunya
Spain

EMail: acabello@ac.upc.edu