Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 6, 2018

D. Farinacci
V. Fuller
D. Meyer
D. Lewis
Cisco Systems
A. Cabellos (Ed.)
UPC/BarcelonaTech
March 5, 2018

The Locator/ID Separation Protocol (LISP) draft-ietf-lisp-rfc6830bis-11

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.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

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

This Internet-Draft will expire on September 6, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction 3
2. Requirements Notation 4
3. Definition of Terms 4
4. Basic Overview 8
4.1. Packet Flow Sequence 10
5. LISP Encapsulation Details 12
5.1. LISP IPv4-in-IPv4 Header Format 12
5.2. LISP IPv6-in-IPv6 Header Format 13
5.3. Tunnel Header Field Descriptions 14
6. LISP EID-to-RLOC Map-Cache 19
7. Dealing with Large Encapsulated Packets 19
7.1. A Stateless Solution to MTU Handling 20
7.2. A Stateful Solution to MTU Handling 21
8. Using Virtualization and Segmentation with LISP 21
9. Routing Locator Selection 22
10. Routing Locator Reachability 24
10.1. Echo Nonce Algorithm 25
11. EID Reachability within a LISP Site 26
12. Routing Locator Hashing 27
13. Changing the Contents of EID-to-RLOC Mappings 28
13.1. Clock Sweep 29
13.2. Database Map-Versioning 29
14. Multicast Considerations 30
15. Router Performance Considerations 31
16. Security Considerations 31
17. Network Management Considerations 32
18. IANA Considerations 32
18.1. LISP UDP Port Numbers 33
19. References 33
19.1. Normative References 33
19.2. Informative References 34

Appendix A. Acknowledgments 39
Appendix B. Document Change Log 39
B.1. Changes to draft-ietf-lisp-rfc6830bis-11 40
B.2. Changes to draft-ietf-lisp-rfc6830bis-10 40
B.3. Changes to draft-ietf-lisp-rfc6830bis-09 40
B.4. Changes to draft-ietf-lisp-rfc6830bis-08 40
B.5. Changes to draft-ietf-lisp-rfc6830bis-07 41
B.6. Changes to draft-ietf-lisp-rfc6830bis-06 41
B.7. Changes to draft-ietf-lisp-rfc6830bis-05 41
B.8. Changes to draft-ietf-lisp-rfc6830bis-04 41
B.9. Changes to draft-ietf-lisp-rfc6830bis-03 42
B.10. Changes to draft-ietf-lisp-rfc6830bis-02 42
B.11. Changes to draft-ietf-lisp-rfc6830bis-01 42
B.12. Changes to draft-ietf-lisp-rfc6830bis-00 42
Authors' Addresses 42

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, 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 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.

2. Requirements Notation

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

Address Family Identifier (AFI): AFI is a term used to describe an address encoding in a packet. An address family that pertains to the data-plane. 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 is a term used in this document to refer 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.

Data-Probe: A Data-Probe is a LISP-encapsulated data packet where the inner-header destination address equals the outer-header destination address used to trigger a Map-Reply by a decapsulating ETR. In addition, the original packet is decapsulated and delivered to the destination host if the destination EID is in the EID-Prefix range configured on the ETR. Otherwise, the packet is discarded. A Data-Probe is used in some of the mapping database designs to "probe" or request a Map-Reply from an ETR; in other cases, Map-Requests are used. See each mapping database design for details. When using Data-Probes, by sending Map-Requests on the underlying routing system, EID-Prefixes must be advertised.

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 global 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 globally visible IP addresses. Note that there MAY be transient conditions when the EID-Prefix for the 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 global in scope.

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 globally (i.e., 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 used 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 globally 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 global 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.

Specifically, when a service provider prepends a LISP header for Traffic Engineering purposes, the router that does this is also regarded as an ITR. The outer RLOC the ISP ITR uses can be based on the outer destination address (the originating ITR's supplied RLOC) or the inner destination address (the originating host's supplied EID).

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.

Negative Mapping Entry: A negative mapping entry, also known as a negative cache entry, is an EID-to-RLOC entry where an EID-Prefix is advertised or stored with no RLOCs. That is, the Locator-Set for the EID-to-RLOC entry is empty or has an encoded Locator count of 0. This type of entry could be used to describe a prefix from a non-LISP site, which is explicitly not in the mapping database. There are a set of well-defined actions that are encoded in a Negative Map-Reply.

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.

TE-ETR: A TE-ETR is an ETR that is deployed in a service provider network that strips an outer LISP header for Traffic Engineering purposes.

TE-ITR: A TE-ITR is an ITR that is deployed in a service provider network that prepends an additional LISP header for Traffic Engineering purposes.

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:

- 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.

- LISP routers mostly deal with Routing Locator addresses. See details in Section 4.1 to clarify what is meant by "mostly".

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

- 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.

- 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.

- 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 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. 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:

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

- 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.

- 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.

- 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].

- 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. The specific method used to do this is not described in this example. See [I-D.ietf-lisp-rfc6833bis] for further information.

3. The ITR sends a LISP Map-Request as specified in [I-D.ietf-lisp-rfc6833bis]. Map-Requests SHOULD be rate-limited.

4. 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.

5. 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.

6. The ITR receives the Map-Reply message, parses the message (to check for format validity), 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.

7. 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.

8. 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.

9. 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.

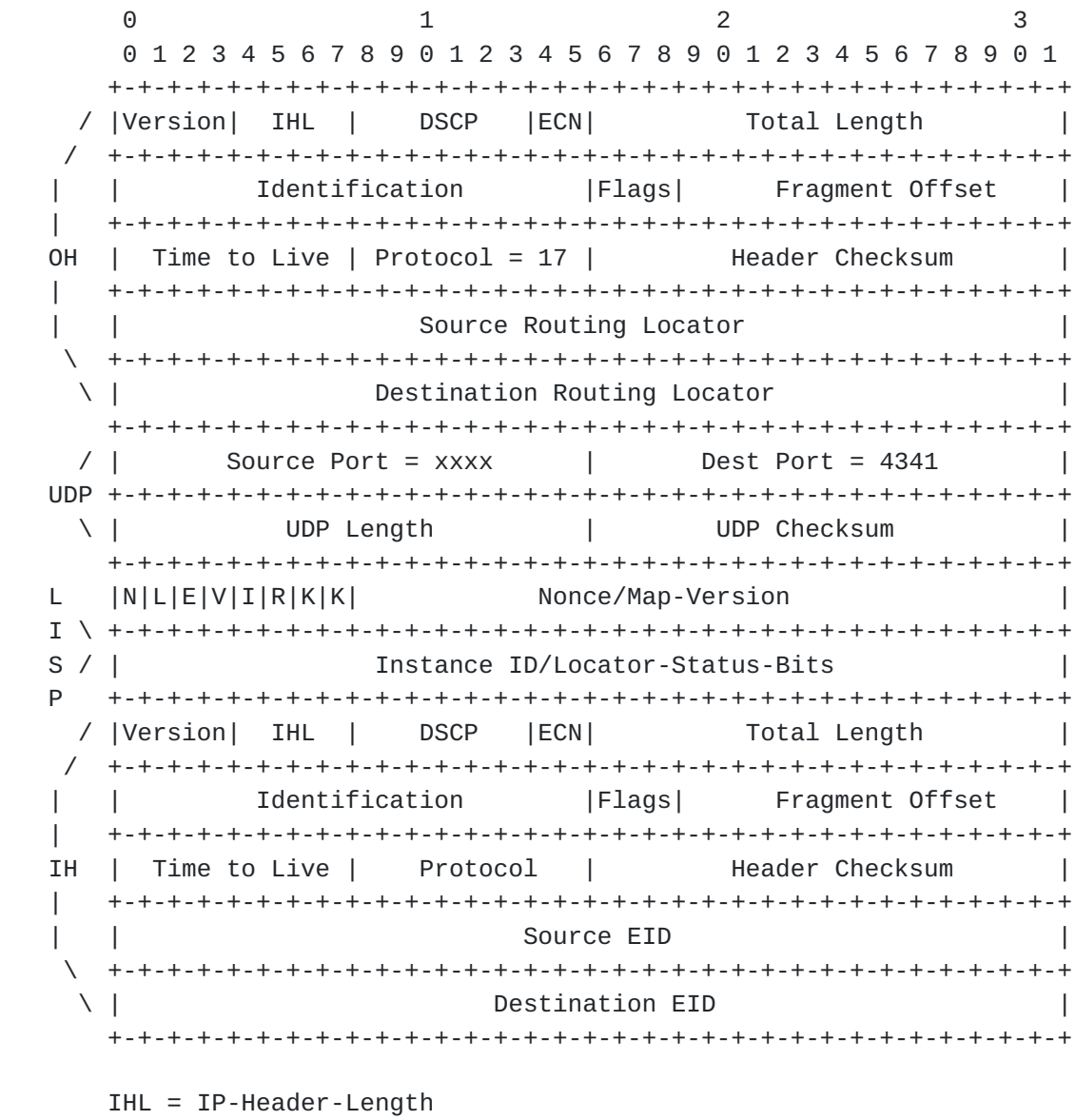
5. LISP Encapsulation Details

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.

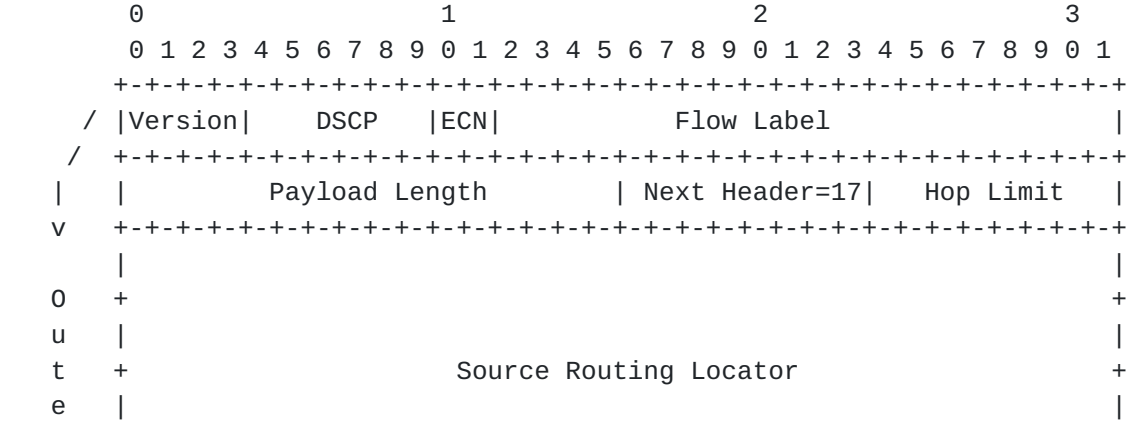
In the case when fragmentation is needed, this specification RECOMMENDS that implementations provide support for one of the proposed fragmentation and reassembly schemes. Two existing schemes are detailed in Section 7.

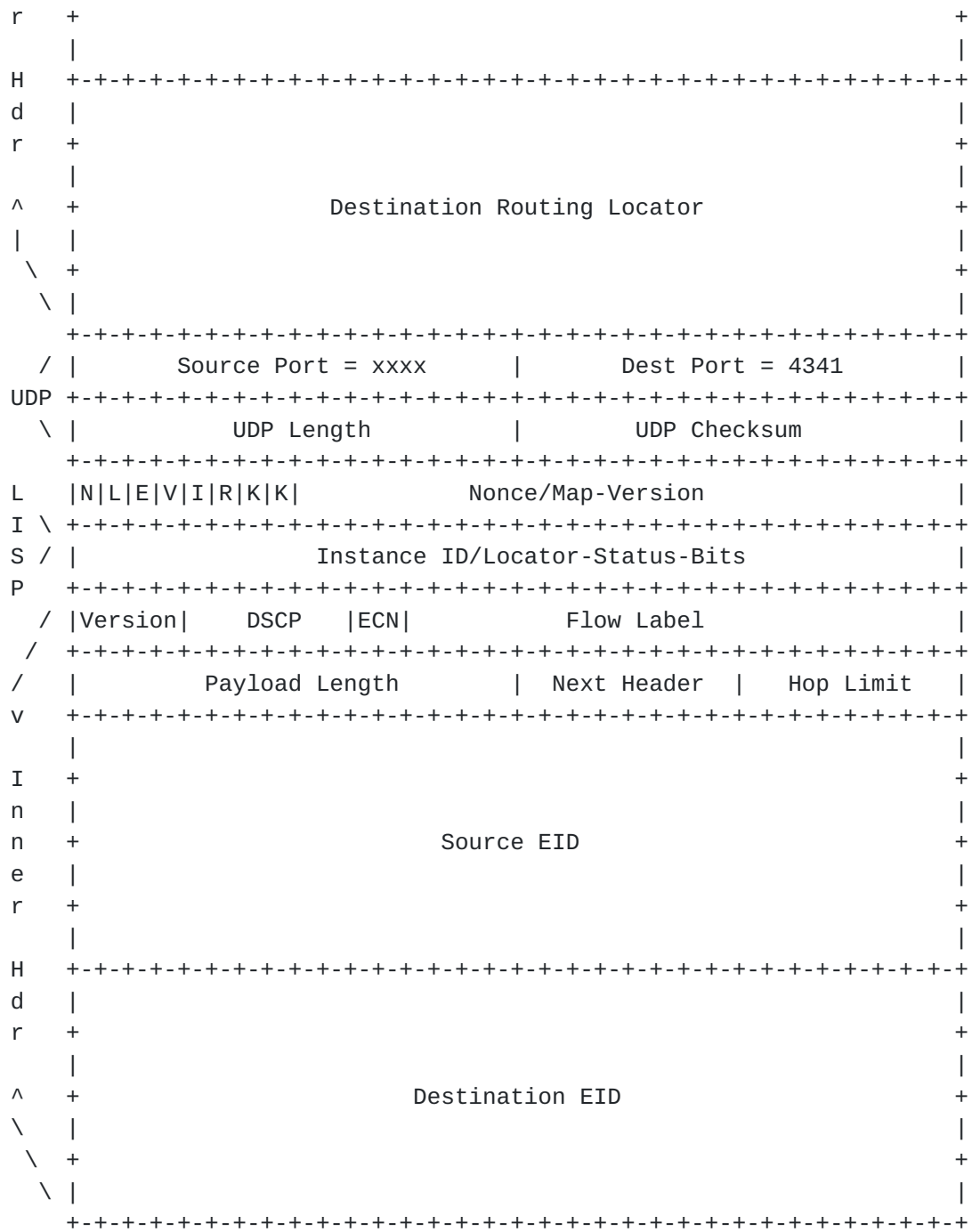
Since IPv4 or IPv6 addresses can be either EIDs or RLOCs, the LISP architecture supports IPv4 EIDs with IPv6 RLOCs (where the inner header is in IPv4 packet format and the outer header is in IPv6 packet format) or IPv6 EIDs with IPv4 RLOCs (where the inner header is in IPv6 packet format and the outer 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].

5.1. LISP IPv4-in-IPv4 Header Format



5.2. LISP IPv6-in-IPv6 Header Format





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.

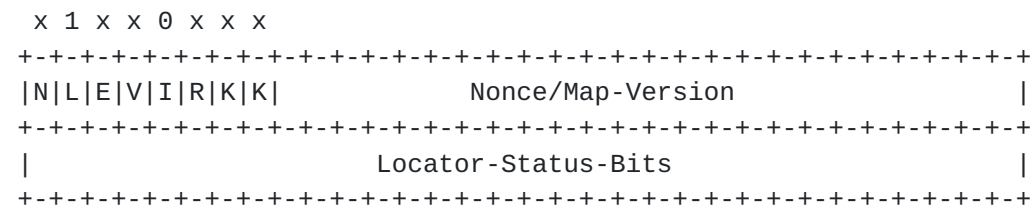
UDP Header: The UDP header contains an ITR selected source port when encapsulating a packet. See Section 12 for details on the hash algorithm used to select a source port based on the 5-tuple of the inner header. The destination port MUST be set to the well-known IANA-assigned port value 4341.

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.

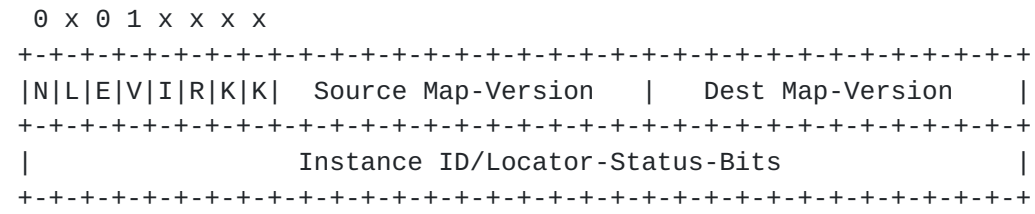
N: The N-bit is the nonce-present bit. When this bit is set to 1, the low-order 24 bits of the first 32 bits of the LISP header contain a Nonce. See Section 10.1 for details. Both N- and V-bits MUST NOT be set in the same packet. If they are, a decapsulating ETR MUST treat the 'Nonce/Map-Version' field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

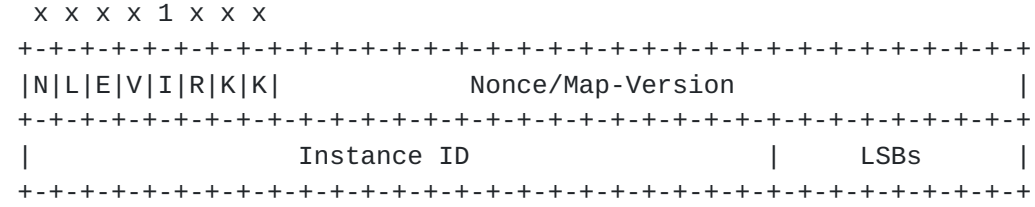


E: The E-bit is the echo-nonce-request bit. This bit MUST be ignored and has no meaning when the N-bit is set to 0. When the N-bit is set to 1 and this bit is set to 1, an ITR is requesting that the nonce value in the 'Nonce' field be echoed back in LISP-encapsulated packets when the ITR is also an ETR.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



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:



R: The R-bit is a Reserved 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 destinations. However, the same nonce can be used for a period of time when encapsulating to the same ETR. 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.

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. 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 'Differentiated Services Code Point' (DSCP) field (or the 'Traffic Class' field, in the case of IPv6) SHOULD be copied from the inner-header DSCP field ('Traffic Class' field, in the case of IPv6) considering the exception listed below.

- The 'Explicit Congestion Notification' (ECN) field (bits 6 and 7 of the IPv6 'Traffic Class' field) requires special treatment in order to avoid discarding indications of congestion [RFC3168]. ITR encapsulation MUST copy the 2-bit 'ECN' field from the inner header to the outer header. Re-encapsulation MUST copy the 2-bit

'ECN' field from the stripped outer header to the new outer header.

When doing ETR/PETR decapsulation:

- The inner-header 'Time to Live' field (or 'Hop Limit' field, in the case of IPv6) SHOULD be copied from the outer-header 'Time to Live' field, when the Time to Live value of the outer header is less than the Time to Live value of the inner header. Failing to perform this check can cause the Time to Live 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.

- The inner-header 'Differentiated Services Code Point' (DSCP) field (or the 'Traffic Class' field, in the case of IPv6) SHOULD be copied from the outer-header DSCP field ('Traffic Class' field, in the case of IPv6) considering the exception listed below.

- The 'Explicit Congestion Notification' (ECN) field (bits 6 and 7 of the IPv6 'Traffic Class' field) requires special treatment in order to avoid discarding indications of congestion [RFC3168]. If the 'ECN' field contains a congestion indication codepoint (the value is '11', the Congestion Experienced (CE) codepoint), then ETR decapsulation MUST copy the 2-bit 'ECN' field from the stripped outer header to the surviving inner header that is used to forward the packet beyond the ETR. These requirements preserve CE indications when a packet that uses ECN traverses a LISP tunnel and becomes marked with a CE indication due to congestion between the tunnel endpoints.

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.

The Explicit Congestion Notification ('ECN') field occupies bits 6 and 7 of both the IPv4 'Type of Service' field and the IPv6 'Traffic Class' field [RFC3168]. The 'ECN' field requires special treatment in order to avoid discarding indications of congestion [RFC3168]. An ITR/PITR encapsulation MUST copy the 2-bit 'ECN' field from the inner header to the outer header. Re-encapsulation MUST copy the 2-bit

'ECN' field from the stripped outer header to the new outer header. If the 'ECN' field contains a congestion indication codepoint (the value is '11', the Congestion Experienced (CE) codepoint), then ETR/ PETR decapsulation MUST copy the 2-bit 'ECN' field from the stripped outer header to the surviving inner header that is used to forward the packet beyond the ETR. These requirements preserve CE indications when a packet that uses ECN traverses a LISP tunnel and becomes marked with a CE indication due to congestion between the tunnel endpoints.

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]. The mapping 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.

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 is performed by the ITR 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 is greater than L and send an ICMP Unreachable/ Fragmentation-Needed 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.

7.2. A Stateful Solution to MTU Handling

An ITR stateful solution to handle MTU issues is described as follows and was first introduced in [OPENLISP]:

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 IPv6-encapsulated packet, or 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 ICMP Unreachable/Fragmentation-Needed message to the ITR. 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 ICMP Unreachable/ Fragmentation-Needed message back to the source. The packet size advertised by the ITR in the ICMP Unreachable/Fragmentation-Needed 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.

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.

The Instance ID that is stored in the mapping database when LISP-DDT [RFC8111] is used is 32 bits in length. That means the control-plane can store more instances than a given data-plane can use. Multiple data-planes can use the same 32-bit space as long as the low-order 24 bits don't overlap among xTRs.

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 RLOC 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 maping entry.

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

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

- 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.

- 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. 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.

- Either side (more likely the server-side ETR) decides not to send a Map-Request. For example, if the server-side ETR does not send Map-Requests, it gleans RLOCs from the client-side ITR, giving 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.

Alternatively, RLOC information MAY be gleaned from received tunneled packets or EID-to-RLOC Map-Request messages. A "gleaned" Map-Cache entry, one learned from the source RLOC of a received encapsulated packet, is only stored and used for a few seconds, pending verification. Verification is 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 is OPTIONAL, 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 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 up.

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:

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

- 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 ETR decapsulates a packet, it 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. Refer to Section 16 for security related issues regarding Locator-Status-Bits.

When an ETR decapsulates a packet, it knows that it is reachable from the encapsulating ITR because that is how the packet arrived. 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.

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 next sends a data packet to the ITR, 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. However, when an ITR is in the echo-nonce-request state, it can echo the ETR's nonce in the next set of packets that it encapsulates and subsequently continue sending echo-nonce-request packets.

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 only 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 RLOC-probe Map-Reply message.

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.

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.

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.

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 data-plane mechanisms for updating EID-to-RLOC mappings. Additionally, the Solicit-Map Request (SMR) control-plane updating mechanism is specified in [I-D.ietf-lisp-rfc6833bis].

When adding a new Locator record in lexicographic order to the end of a Locator-Set, it is easy to update mappings. We assume that new mappings will maintain the same Locator ordering as the old mapping but will just have new Locators appended to the end of the list. So, some ITRs can have a new mapping while other ITRs have only an old mapping that is used until they time out. When an ITR has only an old mapping but detects bits set in the Locator-Status-Bits that correspond to Locators beyond the list it has cached, it simply ignores them. However, this can only happen for locator addresses that are lexicographically greater than the locator addresses in the existing Locator-Set.

When a Locator record is inserted in the middle of a Locator-Set, to maintain lexicographic order, SMR procedure [I-D.ietf-lisp-rfc6833bis] is used to inform ITRs and PITRs of the new Locator-Status-Bit mappings.

When a Locator record is removed from a Locator-Set, ITRs that have the mapping cached will not use the removed Locator because the xTRs will set the Locator-Status-Bit to 0. So, even if the Locator is in the list, it will not be used. For new mapping requests, the xTRs can set the Locator AFI to 0 (indicating an unspecified address), as well as setting the corresponding Locator-Status-Bit to 0. This forces ITRs with old or new mappings to avoid using the removed Locator.

If many changes occur to a mapping over a long period of time, one will find empty record slots in the middle of the Locator-Set and new records appended to the Locator-Set. At some point, it would be useful to compact the Locator-Set so the Locator-Status-Bit settings can be efficiently packed.

We propose here two approaches for Locator-Set compaction: one operational mechanism (clock sweep) and one protocol mechanisms (Map-Versioning). Please note that in addition the Solicit-Map Request (specified in [I-D.ietf-lisp-rfc6833bis]) is a control-plane mechanisms that can be used to update EID-to-RLOC mappings.

13.1. Clock Sweep

The clock sweep approach uses planning in advance and the use of count-down TTLs to time out mappings that have already been cached. The default setting for an EID-to-RLOC mapping TTL is 24 hours. So, there is a 24-hour window to time out old mappings. The following clock sweep procedure is used:

1. 24 hours before a mapping change is to take effect, a network administrator configures the ETRs at a site to start the clock sweep window.

2. During the clock sweep window, ETRs continue to send Map-Reply messages with the current (unchanged) mapping records. The TTL for these mappings is set to 1 hour.

3. 24 hours later, all previous cache entries will have timed out, and any active cache entries will time out within 1 hour. During this 1-hour window, the ETRs continue to send Map-Reply messages with the current (unchanged) mapping records with the TTL set to 1 minute.

4. At the end of the 1-hour window, the ETRs will send Map-Reply messages with the new (changed) mapping records. So, any active caches can get the new mapping contents right away if not cached, or in 1 minute if they had the mapping cached. The new mappings are cached with a TTL equal to the TTL in the Map-Reply.

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.

See [RFC6834] 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 globally routable.

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 [I-D.ietf-lisp-signal-free-multicast], 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:

- 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.

- 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.

- 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

Security considerations for LISP are discussed in [RFC7833].

A complete LISP threat analysis can be found in [RFC7835], in what follows we provide a summary.

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 of a victim's xTR can manipulate such mechanisms to declare a set of RLOCs unreachable. This can be used also, for instance, to declare only one RLOC reachable with the aim of overload it.

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.

Most of the attack vectors can be mitigated with careful deployment and configuration, information learned opportunistically (such as LSB or gleaning) SHOULD be verified with other reachability mechanisms. In addition, systematic rate-limitation and filtering is an effective technique to mitigate attacks that aim to overload the control-plane.

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. 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].

18.1. 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

19. References

19.1. 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.

[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>.

[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>.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <https://www.rfc-editor.org/info/rfc3168>.

[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>.

[RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July 2011, <https://www.rfc-editor.org/info/rfc6275>.

[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>.

19.2. 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-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-01 (work in progress), November 2017.

[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-01 (work in progress), October 2017.

[I-D.ietf-lisp-predictive-rlocs] Farinacci, D. and P. Pillay-Esnault, "LISP Predictive RLOCs", draft-ietf-lisp-predictive-rlocs-01 (work in progress), November 2017.

[I-D.ietf-lisp-sec] Maino, F., Ermagan, V., Cabellos-Aparicio, A., and D. Saucez, "LISP-Security (LISP-SEC)", draft-ietf-lisp-sec-14 (work in progress), October 2017.

[I-D.ietf-lisp-signal-free-multicast] Moreno, V. and D. Farinacci, "Signal-Free LISP Multicast", draft-ietf-lisp-signal-free-multicast-08 (work in progress), February 2018.

[I-D.ietf-lisp-vpn]
Moreno, V. and D. Farinacci, "LISP Virtual Private Networks (VPNs)", draft-ietf-lisp-vpn-01 (work in progress), November 2017.

[LISA96] Lear, E., Tharp, D., Katinsky, J., and J. Coffin, "Renumbering: Threat or Menace?", Usenix Tenth System Administration Conference (LISA 96), October 1996.

[OPENLISP]
Iannone, L., Saucez, D., and O. Bonaventure, "OpenLISP Implementation Report", Work in Progress, July 2008.

[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>.

[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>.

[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>.

[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>.

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

[RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, DOI 10.17487/RFC4632, August 2006, <https://www.rfc-editor.org/info/rfc4632>.

[RFC4866] Arkko, J., Vogt, C., and W. Haddad, "Enhanced Route Optimization for Mobile IPv6", RFC 4866, DOI 10.17487/RFC4866, May 2007, <https://www.rfc-editor.org/info/rfc4866>.

[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>.

[RFC5944] Perkins, C., Ed., "IP Mobility Support for IPv4, Revised", RFC 5944, DOI 10.17487/RFC5944, November 2010, <https://www.rfc-editor.org/info/rfc5944>.

[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>.

[RFC6834] Iannone, L., Saucez, D., and O. Bonaventure, "Locator/ID Separation Protocol (LISP) Map-Versioning", RFC 6834, DOI 10.17487/RFC6834, January 2013, <https://www.rfc-editor.org/info/rfc6834>.

[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>.

[RFC7833] Howlett, J., Hartman, S., and A. Perez-Mendez, Ed., "A RADIUS Attribute, Binding, Profiles, Name Identifier Format, and Confirmation Methods for the Security Assertion Markup Language (SAML)", RFC 7833, DOI 10.17487/RFC7833, May 2016, <https://www.rfc-editor.org/info/rfc7833>.

[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>.

[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>.

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 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.

Appendix B. Document Change Log

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

B.1. 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.2. 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.3. 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.4. 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.5. Changes to draft-ietf-lisp-rfc6830bis-07

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

B.6. 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 addreses can be used.

B.7. Changes to draft-ietf-lisp-rfc6830bis-05

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

B.8. 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.9. Changes to draft-ietf-lisp-rfc6830bis-03

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

B.10. Changes to draft-ietf-lisp-rfc6830bis-02

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

B.11. 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.12. 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
Cisco Systems
Tasman Drive
San Jose, CA 95134
USA

EMail: farinacci@gmail.com

Vince Fuller
Cisco Systems
Tasman Drive
San Jose, CA 95134
USA

EMail: vince.fuller@gmail.com

Dave Meyer
Cisco Systems
170 Tasman Drive
San Jose, CA
USA

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