ï»¿

Internet Engineering Task Force (IETF)
Request for Comments: 9300
Obsoletes: 6830
Category: Standards Track
ISSN: 2070-1721

D. Farinacci
lispers.net
V. Fuller
vaf.net Internet Consulting
D. Meyer
1-4-5.net
D. Lewis
Cisco Systems
A. Cabellos, Ed.
Universitat Politecnica de Catalunya

October 2022

The Locator/ID Separation Protocol (LISP)

Abstract

This document describes the data plane protocol for the Locator/ID Separation Protocol (LISP). LISP defines two namespaces: Endpoint Identifiers (EIDs), which identify end hosts; and Routing Locators (RLOCs), which 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 underlay routers and offers Traffic Engineering (TE), multihoming, and mobility, among other features.

This document obsoletes RFC 6830.

Status of This Memo

This is an Internet Standards Track document.

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

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9300.

Table of Contents

1. Introduction

This document describes the Locator/ID 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), which are used to identify end hosts (e.g., nodes or Virtual Machines); and routable Routing Locators (RLOCs), which are 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; 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 the Map-Cache, that contains EID-to-RLOC mappings. The Map-Cache is populated using the LISP control plane protocol [RFC9301].

LISP does not require changes to either the host protocol stack or underlay routers. By separating the EID from the RLOC space, LISP offers native Traffic Engineering (TE), 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 [RFC9301] specifies the LISP control plane. LISP deployment guidelines can be found in [RFC7215], and [RFC6835] describes considerations for network operational management. Finally, [RFC9299] 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 has been developed and refined, a large number of other ways to use LISP have been found and are being implemented. As such, the design and development of LISP have 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. Definitions of Terms

Address Family Identifier (AFI): "AFI" is a term used to describe an address encoding in a packet. An address family is an address format found in data plane packet headers, for example, an IPv4 address or an IPv6 address. See [AFN], [RFC2453], [RFC2677], and [RFC4760] 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 a generally short-lived, on-demand table in an Ingress Tunnel Router (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 (innermost) LISP header of a packet. The host obtains a destination EID through a Domain Name System (DNS) [RFC1034] lookup or Session Initiation Protocol (SIP) [RFC3261] exchange. This behavior does not change when LISP is in use. 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 discussing other Locator/ID separation proposals, any references to an EID in this document will refer to a LISP EID.

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, all of which follow the UDP header. An ITR prepends LISP headers on packets, and an ETR strips them.

LISP Router: A LISP router is a router that performs the functions of any or all of the following: ITRs, ETRs, Re-encapsulating Tunneling Routers (RTRs), Proxy-ITRs (PITRs), or Proxy-ETRs (PETRs).

LISP Site: A 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 rerouting 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 rerouted 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 address [RFC0791] or IPv6 address [RFC8200] 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 when LISP is not in use as well as when LISP is in use. 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 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 EIDs 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 prepend and strip outer headers with RLOC addresses. See Section 4.2 for details.

* RLOCs are always IP addresses assigned to routers, preferably topologically oriented addresses from provider Classless Inter-Domain Routing (CIDR) 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 internal BGP (iBGP) sessions to other routers in a site while at the same time using RLOCs to terminate external BGP (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 rerouting 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, it is RECOMMENDED 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 separation of location and identity 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. As 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:

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

* MUST NOT use Locator-Status-Bits and Echo-Nonce, as described in Section 10, for RLOC reachability. Instead, they MUST rely solely on control plane methods.

* MUST NOT use gleaning or Locator-Status-Bits and Map-Versioning, as described in Section 13, to update the EID-to-RLOC mappings. Instead, they MUST rely solely on control plane methods.

4.2. Packet Flow Sequence

This section provides an example of the unicast packet flow, also including control plane information as specified in [RFC9301]. The example also assumes the following conditions:

* Source host "host1.abc.example.com" is sending a packet to "host2.xyz.example.com", exactly as it would 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 [RFC9301].

* Map-Replies are sent on the underlying routing system topology, using the control plane protocol [RFC9301].

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 for doing this is to send a LISP Map-Request, as specified in [RFC9301].

3. The Mapping System helps forward 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, it is OPTIONAL for an ETR to 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 ICMPv4 Unreachable / Fragmentation Needed message is returned to the source.

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

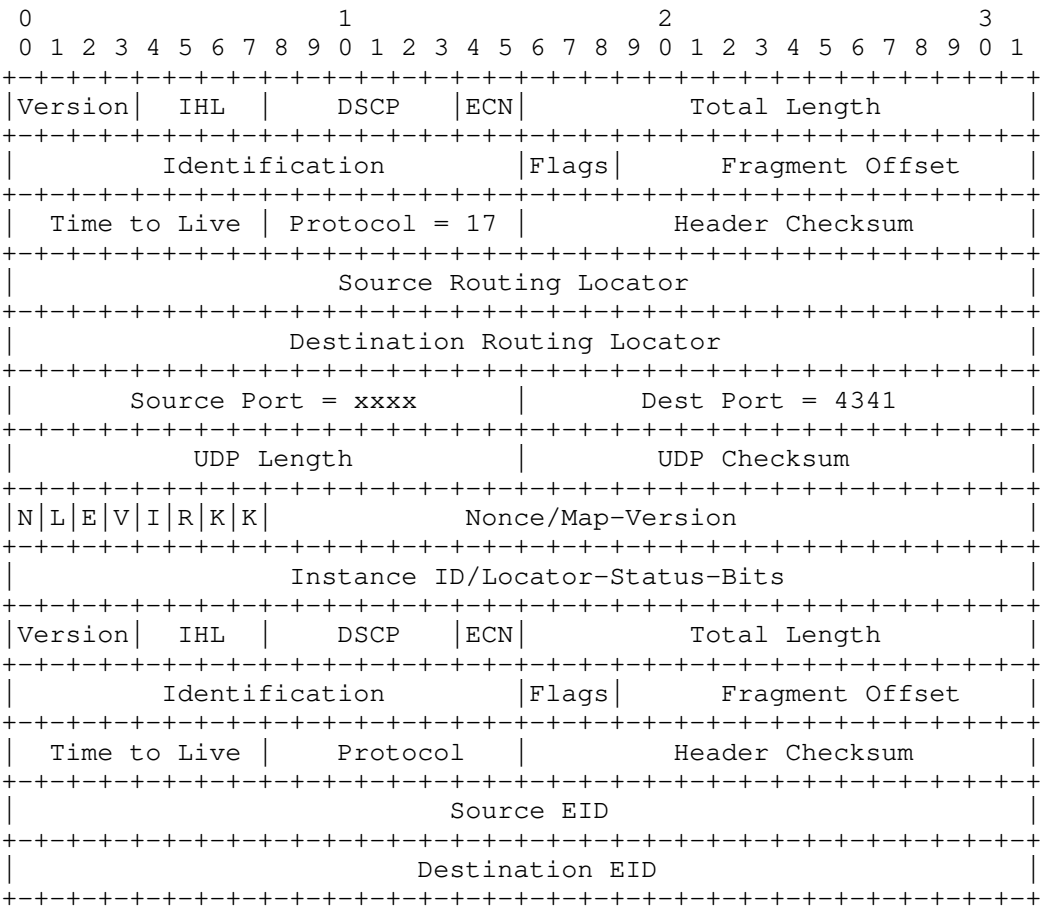
Since IPv4 or IPv6 addresses can be either EIDs or RLOCs, the LISP architecture supports IPv4 EIDs with IPv6 RLOCs (where the inner header is in IPv4 packet format and the outer header is in IPv6 packet format) or IPv6 EIDs with IPv4 RLOCs (where the inner header is in IPv6 packet format and the outer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

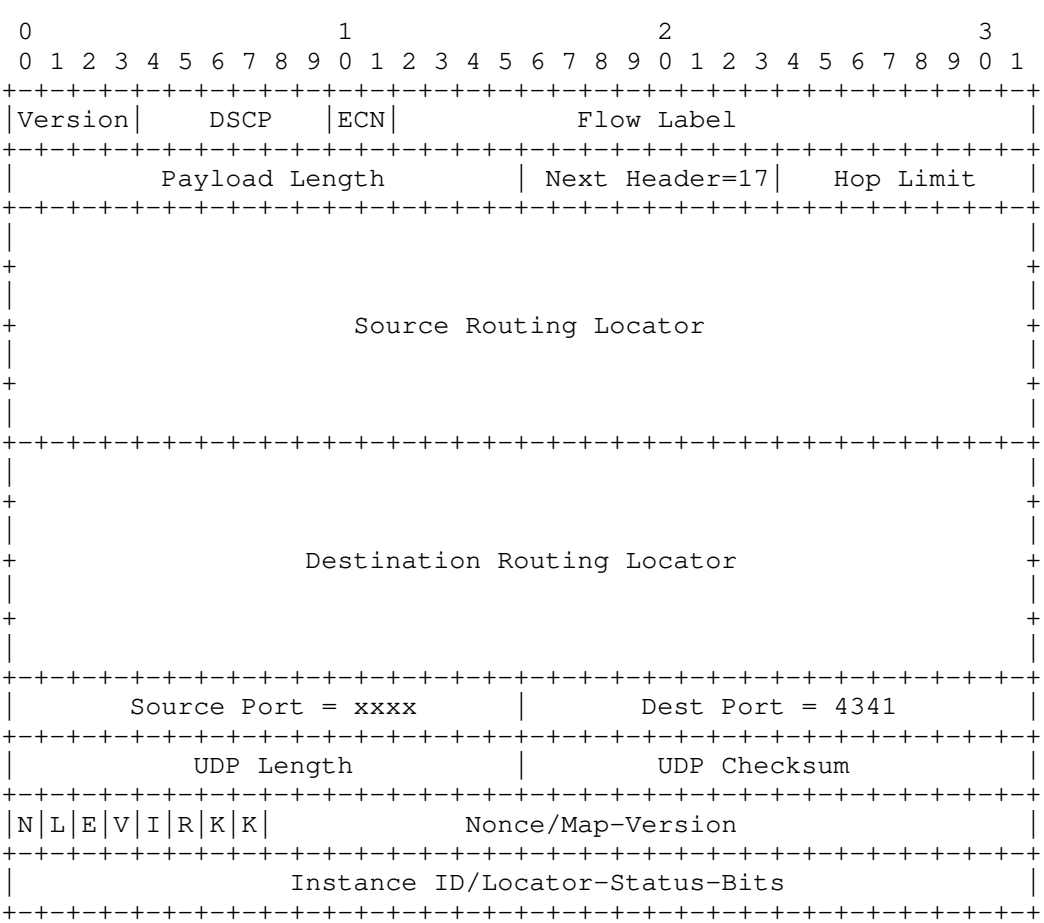
LISP headers against corruption.

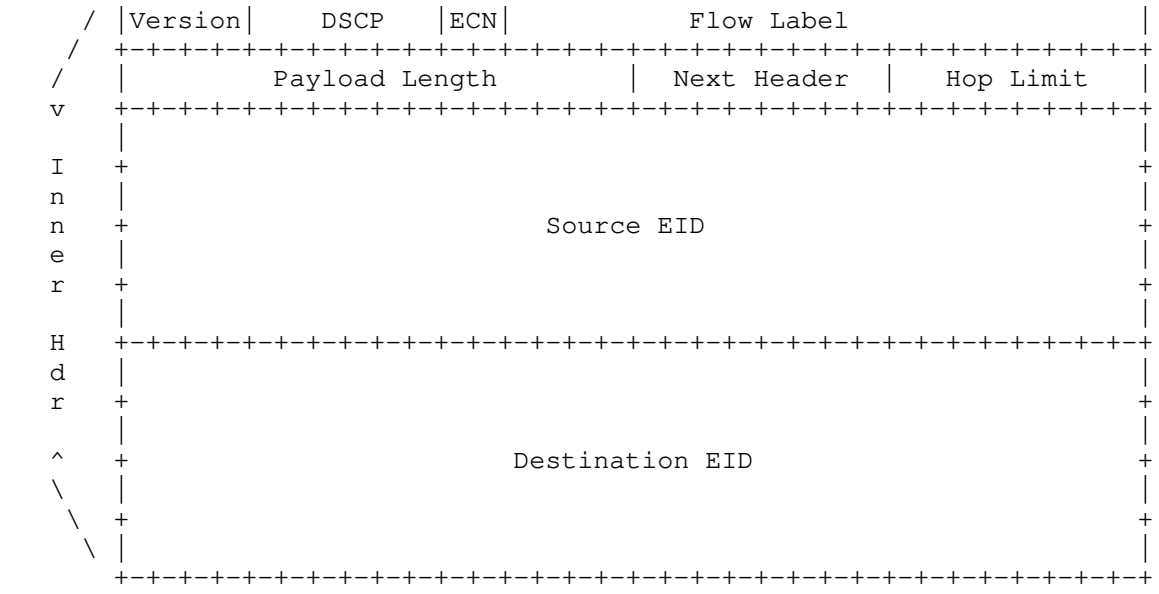
5.1. LISP IPv4-in-IPv4 Header Format



IHL = IP-Header-Length

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 Map-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.

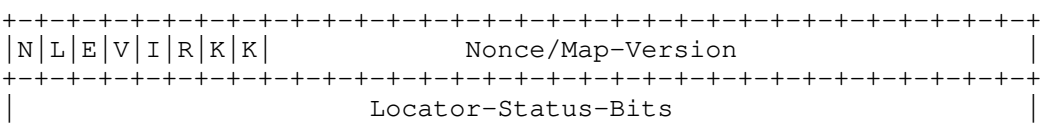
UDP Checksum: The 'UDP Checksum' field SHOULD be transmitted as zero by an ITR for either IPv4 [RFC0768] or IPv6 encapsulation [RFC6935] [RFC6936]. When a packet with a zero UDP checksum is received by an ETR, the ETR MUST accept the packet for decapsulation. When an ITR transmits a non-zero value for 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 either 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.

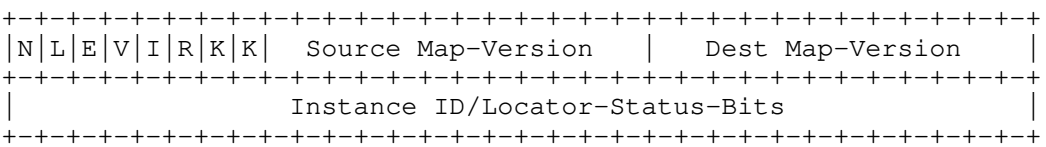
L: The L-bit is the 'Locator-Status-Bits' field enabled bit. When this bit is set to 1, the Locator-Status-Bits in the second

32 bits of the LISP header are in use.

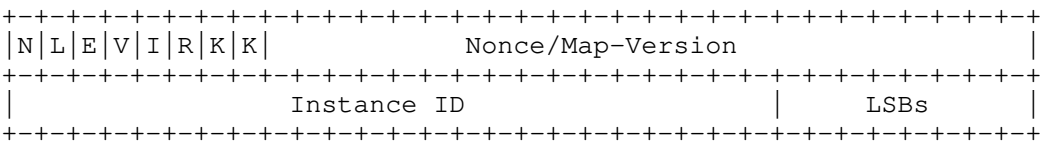


E: The E-bit is the Echo-Nonce-request bit. This bit MUST be ignored and has no meaning when the N-bit is set to 0. When the N-bit is set to 1 and this bit is set to 1, an ITR is requesting that the nonce value in the 'Nonce' field be echoed back in LISP-encapsulated packets when the ITR is also an ETR. 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 [RFC9302] for more details on Database Map-Versioning. 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 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-bits 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 '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 in [RFC2983].

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

When doing ETR/PETR decapsulation:

* 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.

* The outer-header IPv4 'Differentiated Services Code Point (DSCP)' field (or '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 in [RFC2983].

* The IPv4 'Explicit Congestion Notification (ECN)' field and bits 6 and 7 of the IPv6 'Traffic Class' field require 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 discussed 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 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, PETRs, and PITRs perform re-encapsulation operations and need to treat ECN functions 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 data plane 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 [RFC9301] allows an ETR to register data plane capabilities by means of new LISP Canonical Address Format (LCAF) types [RFC8060]. In this way, an ITR can be made aware of the data plane capabilities of an ETR and encapsulate accordingly. The specification of the new LCAF types, the new LCAF mechanisms, and their use are out of the scope of this document.

6. LISP EID-to-RLOC Map-Cache

ITRs and PITRs maintain an on-demand cache, referred to as the 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 [RFC9301]. 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 behavior. 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 (PMTU) 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 the suitable value of L is, so as 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 PMTU 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 implemented 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 Unreachable / Fragmentation Needed or ICMPv6 Packet Too Big (PTB) 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 PMTU issues between the sending ITR and the receiving ETR.

It is RECOMMENDED that L be defined as 1500. Additional information about in-network MTU and fragmentation issues can be found in [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 ICMPv4 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 ICMPv4 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 using ICMP packets for PMTU discovery, as described

in [RFC1191] and [RFC8201], can result in suboptimal behavior in the presence of ICMP packet losses or off-path attackers that spoof ICMP. Possible mitigations include ITRs and ETRs cooperating on MTU probe packets [RFC4821] [RFC8899] or ITRs storing the beginning of large packets to verify that they match the echoed packet in an ICMP Fragmentation Needed / PTB message.

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 [LISP-VPN] for details regarding LISP VPN use cases. Please note that the Instance ID is not protected; an on-path attacker can modify the tags and, for instance, allow communications 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.

The 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 it 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:

* 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. 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.

* 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 server-side ETR 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, as an alternative, use 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 Map-Cache on Tunnel Routers can grow 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 in the 'Time to Live' 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 [RFC9301] 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 RLOC 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 reachability mechanisms based on the control plane are defined in [RFC9301].

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 a 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 an 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 Provider Edge 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 also acting 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 [RFC9302] 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 apply 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) and 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 that 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 request packets are sent is an implementation-specific setting. In this case, an xTR receiving the Echo-Nonce request packets will suspend the Echo-Nonce state and set up an 'Echo-Nonce-request-state' timer. After the 'Echo-Nonce-request-state' timer expires, it will resume the Echo-Nonce 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 that 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. At the time of this writing, this problem exists 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 Routing 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 commonly used 5-tuple hash can be used. The commonly used 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 and 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 their mappings. For scalability reasons, it is desirable to maintain this approach, but implementors 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 [RFC9301].

13.1. Locator-Status-Bits

Locator-Status-Bits (LSBs) can also be used to keep track of the Locator status (up or down) when EID-to-RLOC mappings are changing. When LSBs 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:

1. When a Locator record is added or removed from the Locator-Set, the source xTR will signal this by sending an SMR control plane message [RFC9301] to the destination xTR. At this point, the source xTR MUST NOT use the LSB field, when the L-bit is 0, since the destination xTR site has outdated information. The source xTR will set up a 'use-LSB' timer.

2. As defined in [RFC9301], upon reception of the SMR message, the destination xTR will retrieve the updated EID-to-RLOC mappings by sending a Map-Request.

3. When the 'use-LSB' timer expires, the source xTR can use the LSB again 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 retrieve 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 can send Map-Reply messages carrying a Map-Version Number [RFC9302] in an EID-Record. This is known as the Destination Map-Version Number. ITRs include the Destination Map-Version Number in packets they encapsulate to the site.

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 presented in EID-Records of Map-Register messages [RFC9301], a Map-Version Number is a good way for the Map-Server [RFC9301] to assure that all ETRs for a site registering to it are synchronized according to the Map-Version Number.

See [RFC9302] 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 an RLOC. 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 RLOC address), prepended by a LISP router, can use the same group address as the destination RLOC, use a multicast or unicast RLOC obtained from a Mapping System lookup, or use other means to determine the group address mapping.

With respect to the source RLOC 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 RLOC address, like any other RLOC address, MUST be routable on the underlay.

There are two approaches for LISP-Multicast [RFC6831]: one that uses native multicast routing in the underlay with no support from the Mapping System and another 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], respectively.

15. Router Performance Considerations

LISP is designed to be very "hardware based and 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 Message Authentication Code (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 the interface on which the packet was received can be used to select Virtual Routing and Forwarding (VRF). The VRF system'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 gleaning mechanism 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 bits, 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.

An example of such attacks is when 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 that the attacker only has a small window of time within which to guess a valid nonce that the ITR is requesting to be echoed. 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 tens of seconds. This specific attack can be mitigated by preventing RLOC spoofing in the network by deploying Unicast Reverse Path Forwarding (uRPF) per BCP 84 [RFC8704]. In order to exploit this vulnerability, the off-path attacker must also send Echo-Nonce packets at a 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 to a peering xTR that a local EID-to-RLOC mapping has been updated so that the peering xTR uses a LISP control plane signaling message to retrieve a fresh mapping. This can be used by an attacker to forge the 'Map-Version' field of a LISP-encapsulated header and force an excessive amount of signaling between xTRs that may overload them. Further security considerations on Map-Versioning can be found in [RFC9302].

Locator-Status-Bits, the Echo-Nonce mechanism, 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 that 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 [RFC6830] was published:

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

* Data plane gleaning for creating Map-Cache entries has been made optional. Any ITR implementations that depend on or assume that 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 populating the Map-Cache.

* Most of the changes to this document, which reduce its length, are due to moving the LISP control plane messaging and procedures to [RFC9301].

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

IANA has allocated UDP port number 4341 for the LISP data plane. IANA has updated the description for UDP port 4341 as follows:

Service Name	Port Number	Transport Protocol	Description	Reference
lisp-data	4341	udp	LISP Data Packets	RFC 9300

Table 1

20. References

20.1. Normative References

[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>.

[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>.

[RFC8704] Sriram, K., Montgomery, D., and J. Haas, "Enhanced Feasible-Path Unicast Reverse Path Forwarding", BCP 84, RFC 8704, DOI 10.17487/RFC8704, February 2020, <https://www.rfc-editor.org/info/rfc8704>.

[RFC9301] Farinacci, D., Maino, F., Fuller, V., and A. Cabellos, Ed., "Locator/ID Separation Protocol (LISP) Control Plane", RFC 9301, DOI 10.17487/RFC9301, October 2022, <https://www.rfc-editor.org/info/rfc9301>.

[RFC9302] Iannone, L., Saucez, D., and O. Bonaventure, "Locator/ID Separation Protocol (LISP) Map-Versioning", RFC 9302, DOI 10.17487/RFC9302, October 2022, <https://www.rfc-editor.org/info/rfc9302>.

20.2. Informative References

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

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

[LISP-VPN] Moreno, V. and D. Farinacci, "LISP Virtual Private Networks (VPNs)", Work in Progress, Internet-Draft, draft-ietf-lisp-vpn-10, 3 October 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-lisp-vpn-10>.

[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>.

[RFC2453] Malkin, G., "RIP Version 2", STD 56, RFC 2453, DOI 10.17487/RFC2453, November 1998, <https://www.rfc-editor.org/info/rfc2453>.

[RFC2677] Greene, M., Cucchiara, J., and J. Luciani, "Definitions of Managed Objects for the NBMA Next Hop Resolution Protocol (NHRP)", RFC 2677, DOI 10.17487/RFC2677, August 1999, <https://www.rfc-editor.org/info/rfc2677>.

[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>.

[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>.

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <https://www.rfc-editor.org/info/rfc4760>.

[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>.

[RFC8201] McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed., "Path MTU Discovery for IP version 6", STD 87, RFC 8201, DOI 10.17487/RFC8201, July 2017, <https://www.rfc-editor.org/info/rfc8201>.

[RFC8899] Fairhurst, G., Jones, T., TÃ¼xen, M., RÃ¼ngeler, I., and T. VÃ¶lker, "Packetization Layer Path MTU Discovery for Datagram Transports", RFC 8899, DOI 10.17487/RFC8899, September 2020, <https://www.rfc-editor.org/info/rfc8899>.

[RFC9299] Cabellos, A. and D. Saucez, Ed., "An Architectural Introduction to the Locator/ID Separation Protocol (LISP)", RFC 9299, DOI 10.17487/RFC9299, October 2022, <https://www.rfc-editor.org/info/rfc9299>.

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 was being prepared for IESG Last Call, 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 helped put LISP on the 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 KÃ¼hlewind, Francis Dupont, Benjamin Kaduk, Eric Rescorla, Alvaro Retana, Alexey Melnikov, Alissa Cooper,

Suresh Krishnan, Alberto Rodriguez-Natal, Vina Ermagan, 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.

Authors' Addresses

Dino Farinacci
lispers.net
San Jose, CA
United States of America
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
San Jose, CA
United States of America
Email: darlewis@cisco.com

Albert Cabellos (editor)
Universitat Politecnica de Catalunya
c/ Jordi Girona s/n
08034 Barcelona
Spain
Email: acabello@ac.upc.edu