Network Working Group
Internet-Draft
Intended status: Experimental
Expires: May 17, 2013

D. Farinacci
V. Fuller
D. Meyer
D. Lewis
cisco Systems

November 13, 2012

Locator/ID Separation Protocol (LISP)
draft-ietf-lisp-24

Abstract

This draft describes a network layer based protocol that enables separation of IP addresses into two new numbering spaces: Endpoint Identifiers (EIDs) and Routing Locators (RLOCs). No changes are required to either host protocol stacks or to the "core" of the Internet infrastructure. LISP can be incrementally deployed, without a "flag day", and offers traffic engineering, multi-homing, and mobility benefits to early adopters, even when there are relatively few LISP-capable sites.

Design and development of LISP was largely motivated by the problem statement produced by the October 2006 IAB Routing and Addressing Workshop.

Status of this Memo

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

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

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

This Internet-Draft will expire on May 17, 2013.

Copyright Notice

Table of Contents

1. 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].

2. Introduction

This document describes the Locator/Identifier Separation Protocol (LISP), which provides a set of functions for routers to exchange information used to map from non globally routeable Endpoint Identifiers (EIDs) to routeable Routing Locators (RLOCs). It also defines a mechanism for these LISP routers to encapsulate IP packets addressed with EIDs for transmission across the< Internet that uses RLOCs for routing and forwarding.

Creation of LISP was initially motivated by discussions during the IAB-sponsored Routing and Addressing Workshop held in Amsterdam in October, 2006 (see [RFC4984]). A key conclusion of the workshop was that the Internet routing and addressing system was not scaling well in the face of the explosive growth of new sites; one reason for this poor scaling is the increasing number of multi-homed and other sites that cannot be addressed as part of topologically- or provider-based aggregated prefixes. Additional work that more completely described the problem statement may be found in [RADIR].

A basic observation, made many years ago in early networking research such as that documented in [CHIAPPA] and [RFC4984], is that using a single address field for both identifying a device and for determining where it is topologically located in the network requires optimization along two conflicting axes: for routing to be efficient, the address must be assigned topologically; for collections of devices to be easily and effectively managed, without the need for renumbering in response to topological change (such as that caused by adding or removing attachment points to the network or by mobility events), the address must explicitly not be tied to the topology.

The approach that LISP takes to solving the routing scalability problem is to replace IP addresses with two new types of numbers: Routing Locators (RLOCs), which are topologically assigned to network attachment points (and are therefore amenable to aggregation) and used for routing and forwarding of packets through the network; and Endpoint Identifiers (EIDs), which are assigned independently from the network topology, are used for numbering devices, and are aggregated along administrative boundaries. LISP then defines functions for mapping between the two numbering spaces and for encapsulating traffic originated by devices using non-routeable EIDs for transport across a network infrastructure that routes and forwards using RLOCs. Both RLOCs and EIDs are syntactically-identical to IP addresses; it is the semantics of how they are used that differs.

This document describes the protocol that implements these functions. The database which stores the mappings between EIDs and RLOCs is

explicitly a separate "module" to facilitate experimentation with a variety of approaches. One database design that is being developed for experimentation as part of the LISP working group work is [ALT]. Others that have been described include [CONS], [EMACS], [NERD]. Finally, [LISP-MS], documents a general-purpose service interface for accessing a mapping database; this interface is intended to make the mapping database modular so that different approaches can be tried without the need to modify installed LISP capable devices in LISP sites.

This experimental specification has areas that require additional experience and measurement. It is NOT RECOMMENDED for deployment beyond experimental situations. Results of experimentation may lead to modifications and enhancements of protocol mechanisms defined in this document. See Section 15 for specific, known issues that are in need of further work during development, implementation, and experimentation.

An examination of the implications of LISP on Internet traffic, applications, routers, and security is for future study. This analysis will explain what role LISP can play in scalable routing and will also look at scalability and levels of state required for encapsulation, decapsulation, liveness, and so on.

3. Definition of Terms

Provider Independent (PI) Addresses: PI addresses are an address block assigned from a pool where blocks are not associated with any particular location in the network (e.g. from a particular service provider), and is therefore not topologically aggregatable in the routing system.

Provider Assigned (PA) Addresses: PA addresses are an address block assigned to a site by each service provider to which a site connects. Typically, each block is sub-block of a service provider Classless Inter-Domain Routing (CIDR) [RFC4632] block and is aggregated into the larger block before being advertised into the global Internet. Traditionally, IP multihoming has been implemented by each multi-homed site acquiring its own, globally-visible prefix. LISP uses only topologically-assigned and aggregatable address blocks for RLOCs, eliminating this demonstrably non-scalable practice.

Routing Locator (RLOC): A RLOC is an IPv4 [RFC0791] or IPv6 [RFC2460] address of an egress tunnel router (ETR). A RLOC is the output of an EID-to-RLOC mapping lookup. An EID maps to one or more RLOCs. Typically, RLOCs are numbered from topologically-aggregatable blocks that are assigned to a site at each point to which it attaches to the global Internet; where the topology is defined by the connectivity of provider networks, RLOCs can be thought of as PA addresses. Multiple RLOCs can be assigned to the same ETR device or to multiple ETR devices at a site.

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 an destination address today, for example through a Domain Name System (DNS) [RFC1034] lookup or Session Invitation 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. EIDs MUST NOT be used as LISP RLOCs. 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. As the architecture is realized, if a given bit string is both an RLOC and an EID, it must refer to the same entity in both cases. When used in discussions with other Locator/ID separation proposals, a LISP EID will be called a "LEID". Throughout this document, any references to "EID" refers to an LEID.

EID-prefix: An EID-prefix is a power-of-two block of EIDs which are allocated to a site by an address allocation authority. EID-prefixes are associated with a set of RLOC addresses which make up a "database mapping". EID-prefix allocations can be broken up into smaller blocks when an RLOC set is to be associated with the larger EID-prefix block. A globally routed address block (whether PI or PA) is not inherently an EID-prefix. A globally routed address block MAY be used by its assignee as an EID block. The converse is not supported. That is, a site which receives an explicitly allocated EID-prefix may not use that EID-prefix as a globally routed prefix. This would require coordination and cooperation with the entities managing the mapping infrastructure. Once this has been done, that block could be removed from the globally routed IP system, if other suitable transition and access mechanisms are in place. Discussion of such transition and access mechanisms can be found in [INTERWORK] and [LISP-DEPLOY].

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.

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 globally-routable RLOCs 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 hosts supplied EID).

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.

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.

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.

xTR: A 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. Used synonymously with the term "Tunnel Router". For example, "An xTR can be located at the Customer Edge (CE) router", meaning both ITR and ETR functionality is at the CE router.

LISP Router: A LISP router is a router that performs the functions of any or all of ITR, ETR, PITR, or PETR.

EID-to-RLOC Cache: The EID-to-RLOC cache is a 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-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. The same database mapping entries MUST be configured on all ETRs for a given site. In a steady state the EID-prefixes for the site and the locator-set for each EID-prefix MUST be the same on all ETRs. Procedures to enforce and/or verify this are outside the scope of this document. 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.

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. Any references to tunnels in this specification refers to dynamic encapsulating tunnels and they are never statically configured.

Reencapsulating Tunnels: Reencapsulating tunneling occurs when an ETR removes a LISP header, then acts as an ITR to prepend another LISP header. Doing this allows a packet to be re-routed by the re-encapsulating router without adding the overhead of additional tunnel headers. Any references to tunnels in this specification refers to dynamic encapsulating tunnels and they are never statically configured. When using multiple mapping database systems, care must be taken to not create reencapsulation loops through misconfiguration.

LISP Header: 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 follows the UDP header, an ITR prepends or an ETR strips.

Address Family Identifier (AFI): a term used to describe an address encoding in a packet. An address family currently pertains to an IPv4 or IPv6 address. See [AFI]/[AFI-REGISTRY] 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.

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 (Section 6.1.5).

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. However, this is discouraged if the core is to scale by having less EID-prefixes stored in the core router's routing tables.

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

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

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.

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

Client-side: a term used in this document to indicate a connection initiation attempt by an EID. The ITR(s) at the LISP site are the first to get involved in obtaining database map cache entries by sending Map-Request messages.

Server-side: a term used in this document to indicate a connection initiation attempt is being accepted for a destination EID. The ETR(s) at the destination LISP site are the first to send Map-Replies to the source site initiating the connection. The ETR(s) at this destination site can obtain mappings by gleaning

information from Map-Requests, Data-Probes, or encapsulated packets.

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 6.3.

Anycast Address: 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.

4. Basic Overview

One key concept of LISP is that end-systems (hosts) operate the same way they do today. The IP addresses that hosts use for tracking sockets, 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 egress tunnel router 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 (hosts) only send to addresses which are EIDs. They don't know addresses are EIDs versus RLOCs but assume 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.

- EIDs are always IP addresses assigned to hosts.

- LISP routers mostly deal with Routing Locator addresses. See details later 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 SSH, TELNET, or 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.

- EID prefixes are likely to be hierarchically assigned in a manner which is optimized for administrative convenience and to facilitate scaling of the EID-to-RLOC mapping database. The hierarchy is based on a address allocation hierarchy which is independent of the network topology.

- EIDs may also be structured (subnetted) in a manner suitable for local routing within an autonomous system.

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 ingress tunnel router and the RLOC it uses for the new prepended header would be either a TE-ETR within the ISP (along 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 mandates that a maximum of two LISP headers can be prepended to a packet. For initial LISP deployments, it is assumed two headers is sufficient, where the first prepended header is used at a site for Location/Identity separation and 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 egress tunnel router might be the last-hop router directly-connected to the destination host. Another example, perhaps for a VPN service out-sourced 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. See Section 8 for more details.

4.1. Packet Flow Sequence

This section provides an example of the unicast packet flow with 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 multi-homed, 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 LISP site.
- Map-Requests can be sent on the underlying routing system topology, to a mapping database system, or directly over an alternative topology [ALT]. 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-Replies are sent on the underlying routing system topology.

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 [ALT] or [CONS] for possible solutions.
3. The ITR will send a LISP Map-Request. Map-Requests SHOULD be rate-limited.

4. When an alternate mapping system is not in use, the Map-Request packet is routed through the underlying routing system. Otherwise, the Map-Request packet is routed on an alternate logical topology, for example the [ALT] database mapping system. In either case, 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 mapping 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 the packet MAY be sent to a different ETR than the one which 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.

In order to defer the need for a mapping lookup in the reverse direction, an ETR MAY 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 "gleaned" 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 ITR and the ETR may also influence the decision the other makes in selecting an RLOC. See Section 6 for more details.

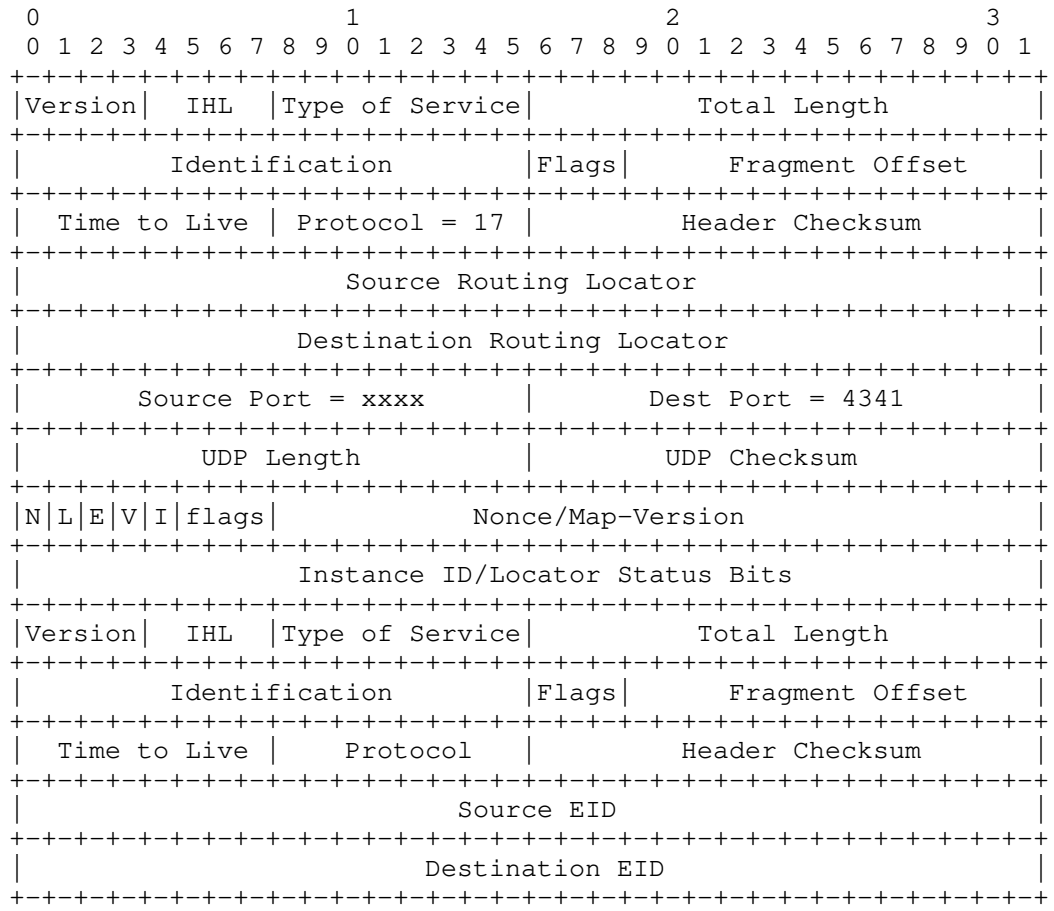
5. LISP Encapsulation Details

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 Too Big message is returned to the source.

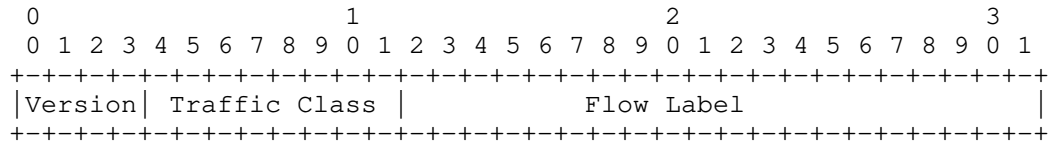
This specification RECOMMENDS that implementations provide support for one of the proposed fragmentation and reassembly schemes. Two existing schemes are detailed in Section 5.4.

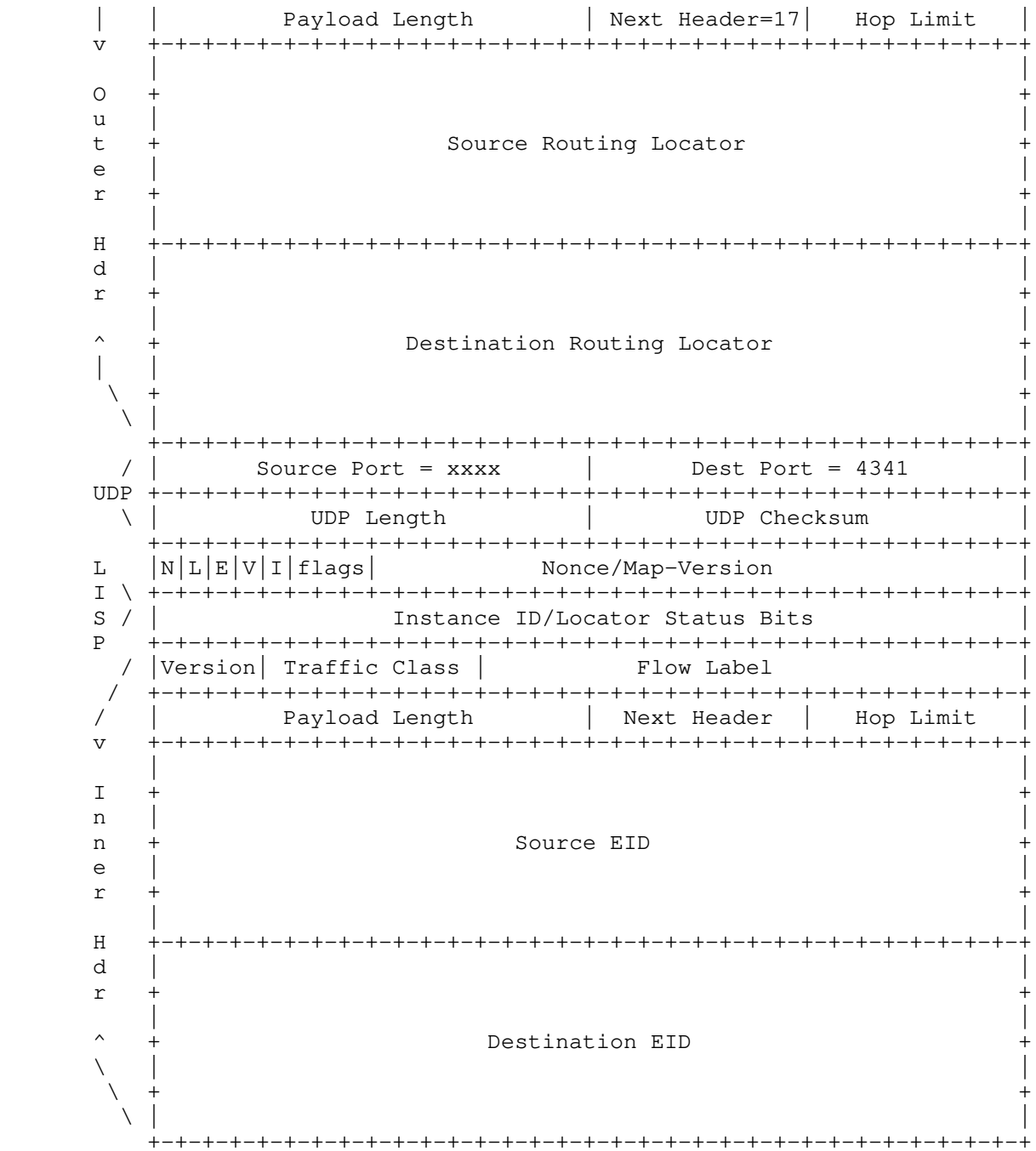
Since IPv4 or IPv6 addresses can be either EIDs or RLOCs, the LISP architecture supports IPv4 EIDs with IPv6 RLOCs (where the inner header is in IPv4 packet format and the other header is in IPv6 packet format) or IPv6 EIDs with IPv4 RLOCs (where the inner header is in IPv6 packet format and the other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

5.1. LISP IPv4-in-IPv4 Header Format



5.2. LISP IPv6-in-IPv6 Header Format




Payload Length
Next Header=17
Hop Limit
Outer Hdr
Source Routing Locator
Destination Routing Locator
UDP
Source Port = xxxx
Dest Port = 4341
UDP Length
UDP Checksum
LISP
N L E V I flags
Nonce/Map-Version
Instance ID/Locator Status Bits
Version
Traffic Class
Flow Label
Payload Length
Next Header
Hop Limit
Inner Hdr
Source EID
Destination EID

5.3. Tunnel Header Field Descriptions

Inner Header (IH): The inner header is the header on the datagram received from the originating host. The source and destination IP addresses are EIDs, [RFC0791], [RFC2460].

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 DF bit Flags field is according to rules in Section 5.4.1 and Section 5.4.2.

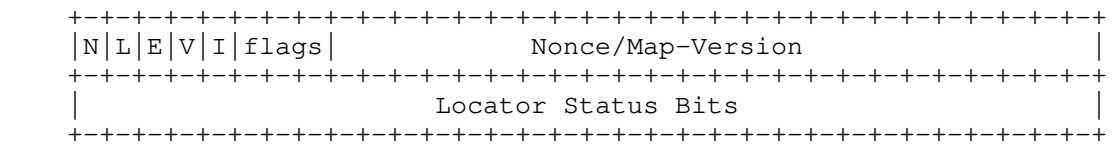
UDP Header: The UDP header contains an ITR selected source port when encapsulating a packet. See Section 6.5 for details on the hash algorithm used to select a source port based on the 5-tuple of the inner header. The destination port MUST be set to the well-known IANA assigned port value 4341.

UDP Checksum: The UDP checksum field SHOULD be transmitted as zero by an ITR for either IPv4 [RFC0768] or IPv6 encapsulation [UDP-TUNNELS] [UDP-ZERO]. 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 checksums for all tunneling protocols, including LISP, is under active discussion within the IETF. When that discussion concludes, any necessary changes will be made to align LISP with the outcome of the broader discussion.

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.

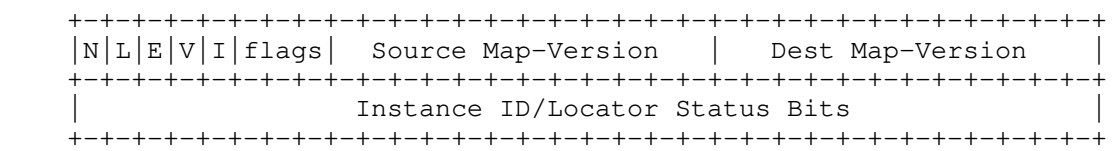
N: The N bit is the nonce-present bit. When this bit is set to 1, the low-order 24-bits of the first 32-bits of the LISP header contains a Nonce. See Section 6.3.1 for details. Both N and V bits MUST NOT be set in the same packet. If they are, a decapsulating ETR MUST treat the "Nonce/Map-Version" field 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.

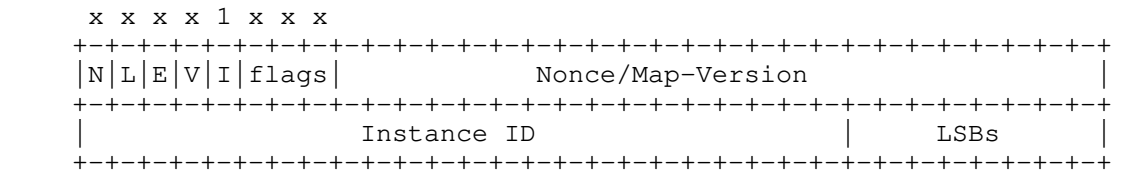


E: The E bit is the echo-nonce-request bit. This bit MUST be ignored and has no meaning when the N bit is set to 0. When the N bit is set to 1 and this bit is set to 1, means an ITR is requesting for the nonce value in the Nonce field to be echoed back in LISP encapsulated packets when the ITR is also an ETR. See Section 6.3.1 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 6.6.3 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 5.5 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 in this case:



flags: The flags field is a 3-bit field is reserved for future flag use. It MUST be set to 0 on transmit and MUST be ignored on receipt.

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 to the same destination. 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 6.3.1 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 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 the RLOC associated with the bit ordinal has up status. See Section 6.3 for details on how an ITR can determine the status of the ETRs at the same site. When a site has multiple EID-prefixes which 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 the ETR is considered 'up'.

When doing ITR/PITR encapsulation:

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

- The outer header Type of Service field (or the Traffic Class field, in the case of IPv6) SHOULD be copied from the inner header Type of Service field (with one exception, see below).

When doing ETR/PETR decapsulation:

- The inner header Time to Live field (or Hop Limit field, in case of IPv6) SHOULD be copied from the outer header Time to Live field, when the Time to Live field of the outer header is less than the Time to Live of the inner header. Failing to perform this check can cause the Time to Live of the inner header to increment across encapsulation/decapsulation cycle. 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 Type of Service field (or the Traffic Class field, in the case of IPv6) SHOULD be copied from the outer header Type of Service field (with one exception, see below).

Note if an ETR/PETR is also an ITR/PITR and choose to reencapsulate 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 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. See Section 9.3 for TTL exception handling for traceroute packets.

The 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]. 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. 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 Congestion Experienced (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.

5.4. 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 Reencapsulating and Recursive Tunneling. So any actions below referring to an ITR also apply to an TE-ITR.

5.4.1. 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 so the effective MTU can be met. That is, S = L - 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, 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 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 is 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 sends an ICMP Too Big message to the source with a value of S, where S is (L - H).

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

This specification RECOMMENDS that L be defined as 1500.

5.4.2. 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 mapping cache entry. The effective MTU is what the core network can deliver along the path between ITR and ETR.

2. When an IPv6 encapsulated packet or an IPv4 encapsulated packet with DF bit set to 1, exceeds what the core network can deliver, one of the intermediate routers on the path will send an ICMP Too Big 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 mapping 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 mapping cache entry associated with the destination EID the packet is for, the ITR will send an ICMP Too Big message back to the source. The packet size advertised by the ITR in the ICMP Too Big 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.

5.5. Using Virtualization and Segmentation with LISP

When multiple organizations inside of a LISP site are using private addresses [RFC1918] as EID-prefixes, their address spaces MUST remain segregated due to possible address duplication. An Instance ID in

the address encoding can aid in making the entire AFI based address unique. See IANA Considerations Section 14.2 for details for possible address encodings.

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.

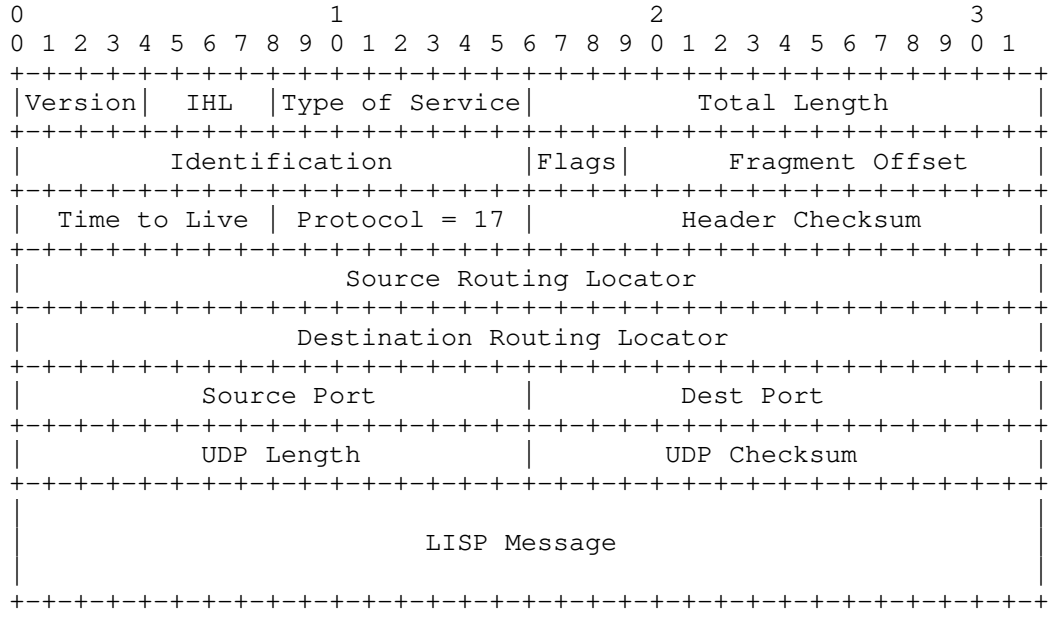
When an ETR decapsulates a packet, the Instance ID from the LISP header is used as a table identifier to locate the forwarding table to use for the inner destination EID lookup.

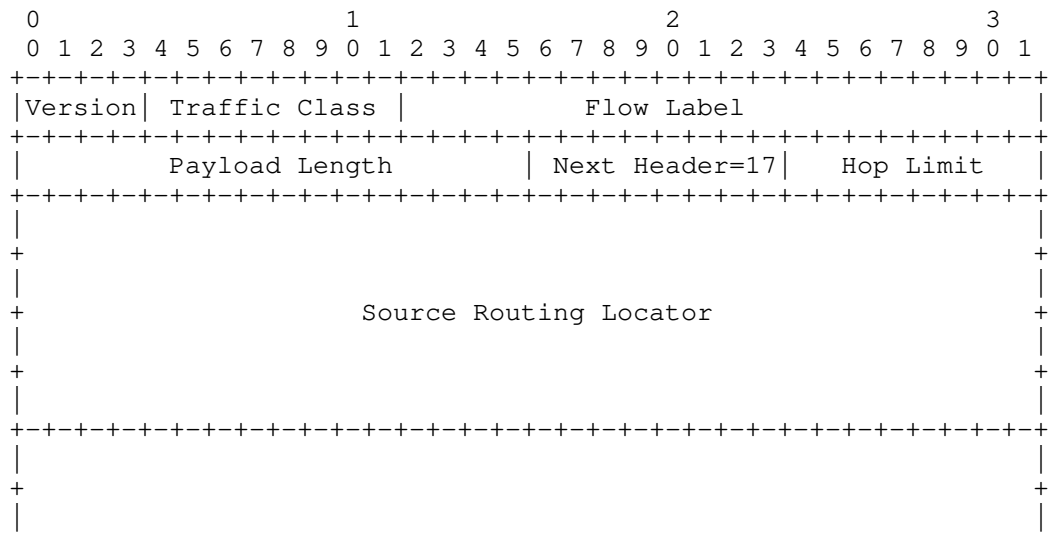
For example, a 802.1Q VLAN tag or VPN identifier could be used as a 24-bit Instance ID.

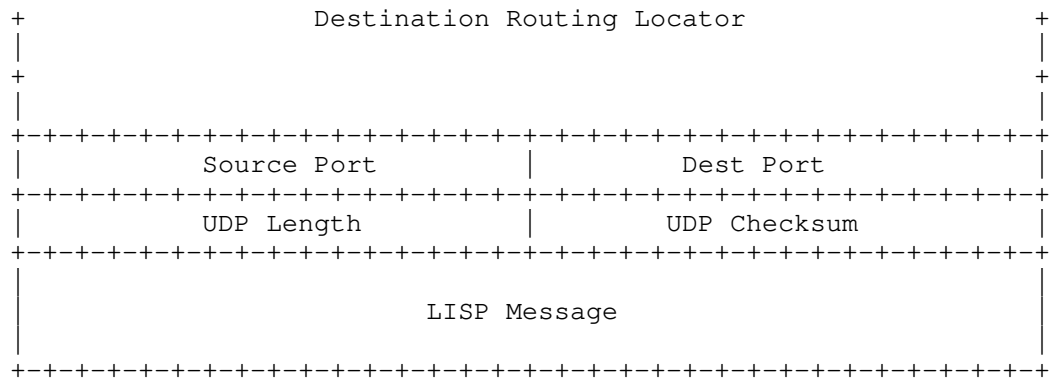
6. EID-to-RLOC Mapping

6.1. LISP IPv4 and IPv6 Control Plane Packet Formats

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







The LISP UDP-based messages are the Map-Request and Map-Reply messages. When a UDP Map-Request is sent, the UDP source port is chosen by the sender and the destination UDP port number is set to 4342. When a UDP Map-Reply is sent, the source UDP port number is set to 4342 and the destination UDP port number is copied from the source port of either the Map-Request or the invoking data packet. Implementations MUST be prepared to accept packets when either the source port or destination UDP port is set to 4342 due to NATs changing port number values.

The UDP Length field will reflect the length of the UDP header and the LISP Message payload.

The UDP Checksum is computed and set to non-zero for Map-Request, Map-Reply, Map-Register and ECM control messages. It MUST be checked on receipt and if the checksum fails, the packet MUST be dropped.

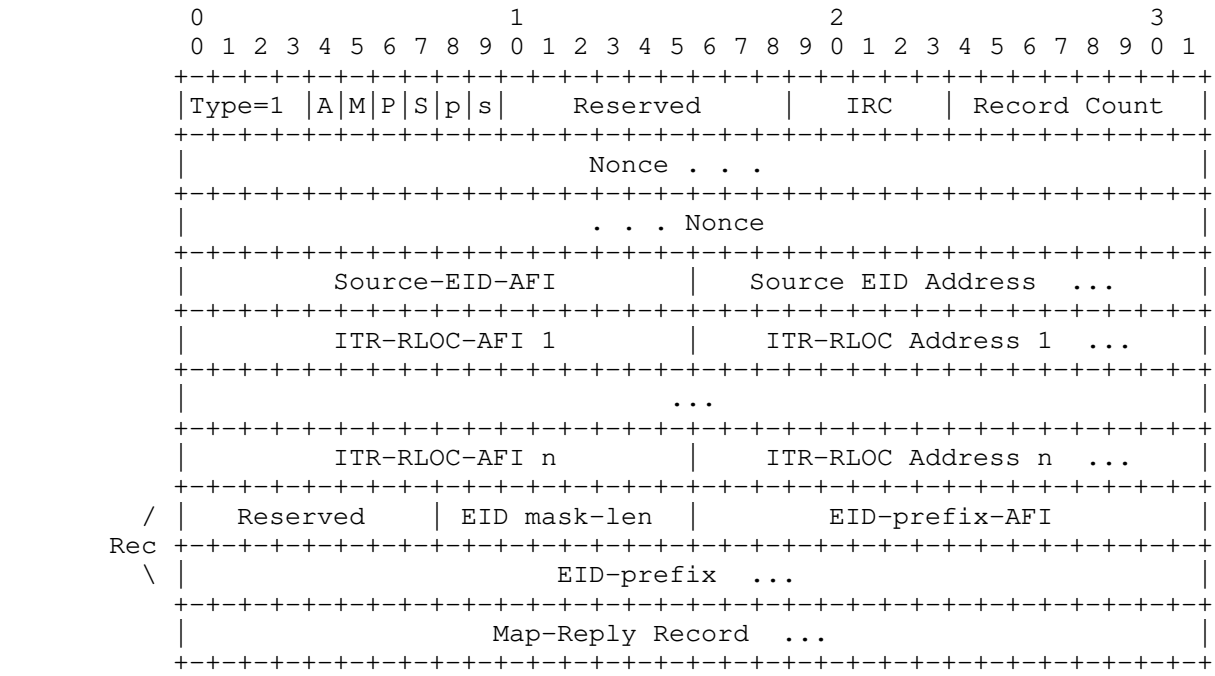
The format of control messages includes the UDP header so the checksum and length fields can be used to protect and delimit message boundaries.

6.1.1. LISP Packet Type Allocations

This section will be the authoritative source for allocating LISP Type values and for defining LISP control message formats. Current allocations are:

Reserved:	0	b'0000'
LISP Map-Request:	1	b'0001'
LISP Map-Reply:	2	b'0010'
LISP Map-Register:	3	b'0011'
LISP Map-Notify:	4	b'0100'
LISP Encapsulated Control Message:	8	b'1000'

6.1.2. Map-Request Message Format



Packet field descriptions:

Type: 1 (Map-Request)

A: This is an authoritative bit, which is set to 0 for UDP-based Map-Requests sent by an ITR. Set to 1 when an ITR wants the destination site to return the Map-Reply rather than the mapping database system.

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

P: This is the probe-bit which indicates that a Map-Request SHOULD be treated as a locator reachability probe. The receiver SHOULD respond with a Map-Reply with the probe-bit set, indicating the Map-Reply is a locator reachability probe reply, with the nonce copied from the Map-Request. See Section 6.3.2 for more details.

S: This is the Solicit-Map-Request (SMR) bit. See Section 6.6.2 for details.

p: This is the PITR bit. This bit is set to 1 when a PITR sends a Map-Request.

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

Reserved: It MUST be set to 0 on transmit and MUST be ignored on receipt.

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

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

Nonce: An 8-octet random value created by the sender of the Map-Request. This nonce will be returned in the Map-Reply. The security of the LISP mapping protocol depends critically on the strength of the nonce in the Map-Request message. The nonce SHOULD be generated by a properly seeded pseudo-random (or strong random) source. See [RFC4086] for advice on generating security-sensitive random data.

Source-EID-AFI: Address family of the "Source EID Address" field.

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

ITR-RLOC-AFI: Address family of the "ITR-RLOC Address" field that follows this field.

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

EID mask-len: Mask length for EID prefix.

EID-prefix-AFI: Address family of EID-prefix according to [AFI]

EID-prefix: 4 octets if an IPv4 address-family, 16 octets if an IPv6 address-family. When a Map-Request is sent by an ITR because a data packet is received for a destination where there is no mapping entry, the EID-prefix is set to the destination IP address of the data packet. And the 'EID mask-len' is set to 32 or 128 for IPv4 or IPv6, respectively. When an xTR wants to query a site about the status of a mapping it already has cached, the EID-prefix used in the Map-Request has the same mask-length as the EID-prefix returned from the site when it sent a Map-Reply message.

Map-Reply Record: When the M bit is set, this field is the size of a single "Record" in the Map-Reply format. This Map-Reply record contains the EID-to-RLOC mapping entry associated with the Source EID. This allows the ETR which will receive this Map-Request to cache the data if it chooses to do so.

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

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

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

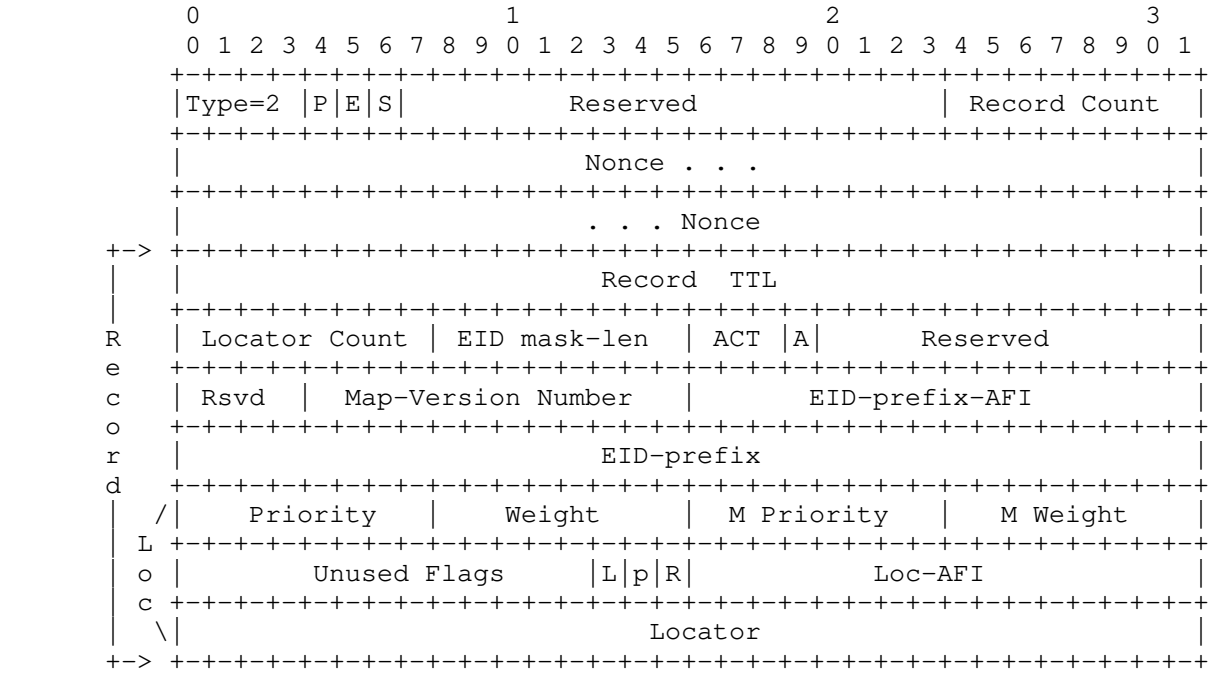
Map-Requests can also be LISP encapsulated using UDP destination port 4342 with a LISP type value set to "Encapsulated Control Message", when sent from an ITR to a Map-Resolver. Likewise, Map-Requests are LISP encapsulated the same way from a Map-Server to an ETR. Details on encapsulated Map-Requests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

Map-Requests MUST be rate-limited. It is RECOMMENDED that a Map-Request for the same EID-prefix be sent no more than once per second.

An ITR that is configured with mapping database information (i.e. it is also an ETR) MAY optionally include those mappings in a Map-Request. When an ETR configured to accept and verify such "piggybacked" mapping data receives such a Map-Request and it does not have this mapping in the map-cache, it MAY originate a "verifying Map-Request", addressed to the map-requesting ITR and the ETR MAY add a map-cache entry. If the ETR has a map-cache entry that matches the "piggybacked" EID and the RLOC is in the locator-set for the entry, then it may send the "verifying Map-Request" directly to the originating Map-Request source. If the RLOC is not in the locator-set, then the ETR MUST send the "verifying Map-Request" to the "piggybacked" EID. Doing this forces the "verifying Map-Request" to

go through the mapping database system to reach the authoritative source of information about that EID, guarding against RLOC-spoofing in in the "piggybacked" mapping data.

6.1.4. Map-Reply Message Format



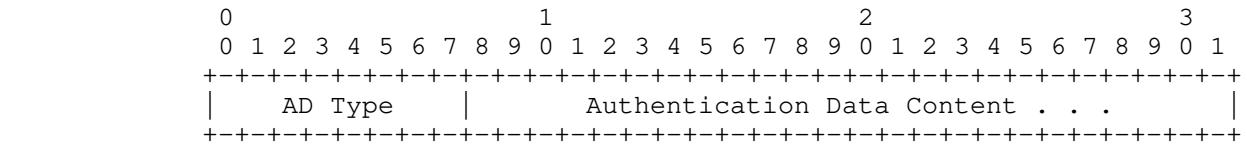
Packet field descriptions:

Type: 2 (Map-Reply)

P: This is the probe-bit which indicates that the Map-Reply is in response to a locator reachability probe Map-Request. The nonce field MUST contain a copy of the nonce value from the original Map-Request. See Section 6.3.2 for more details.

E: Indicates that the ETR which sends this Map-Reply message is advertising that the site is enabled for the Echo-Nonce locator reachability algorithm. See Section 6.3.1 for more details.

S: This is the Security bit. When set to 1 the following authentication information will be appended to the end of the Map-Reply. The detailed format of the Authentication Data Content is for further study.



Reserved: It MUST be set to 0 on transmit and MUST be ignored on receipt.

Record Count: The number of records in this reply message. A record is comprised of that portion of the packet labeled 'Record' above and occurs the number of times equal to Record count.

Nonce: A 24-bit value set in a Data-Probe packet or a 64-bit value from the Map-Request is echoed in this Nonce field of the Map-Reply. When a 24-bit value is supplied, it resides in the low-order 64 bits of the nonce field.

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

Locator Count: The number of Locator entries. A locator entry comprises what is labeled above as 'Loc'. The locator count can be 0 indicating there are no locators for the EID-prefix.

EID mask-len: Mask length for EID prefix.

ACT: This 3-bit field describes negative Map-Reply actions. In any other message type, these bits are set to 0 and ignored on receipt. These bits are used only when the 'Locator Count' field is set to 0. The action bits are encoded only in Map-Reply messages. The actions defined are used by an ITR or PITR when a destination EID matches a negative mapping cache entry. Unassigned values should cause a map-cache entry to be created and, when packets match this negative cache entry, they will be dropped. The current assigned values are:

(0) No-Action: The map-cache is kept alive and no packet encapsulation occurs.

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

(2) Send-Map-Request: The packet invokes sending a Map-Request.

(3) Drop: A packet that matches this map-cache entry is dropped. An ICMP Unreachable message SHOULD be sent.

A: The Authoritative bit, when sent is always set to 1 by an ETR. When a Map-Server is proxy Map-Replying [LISP-MS] for a LISP site, the Authoritative bit is set to 0. This indicates to requesting ITRs that the Map-Reply was not originated by a LISP node managed at the site that owns the EID-prefix.

Map-Version Number: When this 12-bit value is non-zero the Map-Reply sender is informing the ITR what the version number is for the EID-record contained in the Map-Reply. The ETR can allocate this number internally but MUST coordinate this value with other ETRs for the site. When this value is 0, there is no versioning information conveyed. The Map-Version Number can be included in Map-Request and Map-Register messages. See Section 6.6.3 for more details.

EID-prefix-AFI: Address family of EID-prefix according to [AFI].

EID-prefix: 4 octets if an IPv4 address-family, 16 octets if an IPv6 address-family.

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

Weight: when priorities are the same for multiple RLOCs, the weight indicates how to balance unicast traffic between them. Weight is encoded as a relative weight of total unicast packets that match the mapping entry. For example if there are 4 locators in a locator set, where the weights assigned are 30, 20, 20, and 10, the first locator will get 37.5% of the traffic, the 2nd and 3rd locators will get 25% of traffic and the 4th locator will get 12.5% of the traffic. If all weights for a locator-set are equal, receiver of the Map-Reply will decide how to load-split traffic. See Section 6.5 for a suggested hash algorithm to distribute load

across locators with same priority and equal weight values.

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

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

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

L: when this bit is set, the locator is flagged as a local locator to the ETR that is sending the Map-Reply. When a Map-Server is doing proxy Map-Replying [LISP-MS] for a LISP site, the L bit is set to 0 for all locators in this locator-set.

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

R: set when the sender of a Map-Reply has a route to the locator in the locator data record. This receiver may find this useful to know if the locator is up but not necessarily reachable from the receiver's point of view. See also Section 6.4 for another way the R-bit may be used.

Locator: an IPv4 or IPv6 address (as encoded by the 'Loc-AFI' field) assigned to an ETR. Note that the destination RLOC address MAY be an anycast address. A source RLOC can be an anycast address as well. The source or destination RLOC MUST NOT be the broadcast address (255.255.255.255 or any subnet broadcast address known to the router), and MUST NOT be a link-local multicast address. The source RLOC MUST NOT be a multicast address. The destination RLOC

SHOULD be a multicast address if it is being mapped from a multicast destination EID.

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

A Map-Reply returns an EID-prefix with a prefix length that is less than or equal to the EID being requested. The EID being requested is either from the destination field of an IP header of a Data-Probe or the EID record of a Map-Request. The RLOCs in the Map-Reply are globally-routable IP addresses of all ETRs for the LISP site. Each RLOC conveys status reachability but does not convey path reachability from a requesters perspective. Separate testing of path reachability is required, See Section 6.3 for details.

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

When an ETR is configured with overlapping EID-prefixes, a Map-Request with an EID that longest matches any EID-prefix MUST be returned in a single Map-Reply message. For instance, if an ETR had database mapping entries for EID-prefixes:

10.0.0.0/8
10.1.0.0/16
10.1.1.0/24
10.1.2.0/24

A Map-Request for EID 10.1.1.1 would cause a Map-Reply with a record count of 1 to be returned with a mapping record EID-prefix of 10.1.1.0/24.

A Map-Request for EID 10.1.5.5, would cause a Map-Reply with a record count of 3 to be returned with mapping records for EID-prefixes 10.1.0.0/16, 10.1.1.0/24, and 10.1.2.0/24.

Note that not all overlapping EID-prefixes need to be returned, only the more specifics (note in the second example above 10.0.0.0/8 was not returned for requesting EID 10.1.5.5) entries for the matching

EID-prefix of the requesting EID. When more than one EID-prefix is returned, all SHOULD use the same Time-to-Live value so they can all time out at the same time. When a more specific EID-prefix is received later, its Time-to-Live value in the Map-Reply record can be stored even when other less specifics exist. When a less specific EID-prefix is received later, its map-cache expiration time SHOULD be set to the minimum expiration time of any more specific EID-prefix in the map-cache. This is done so the integrity of the EID-prefix set is wholly maintained so no more-specific entries are removed from the map-cache while keeping less-specific entries.

Map-Replies SHOULD be sent for an EID-prefix no more often than once per second to the same requesting router. For scalability, it is expected that aggregation of EID addresses into EID-prefixes will allow one Map-Reply to satisfy a mapping for the EID addresses in the prefix range thereby reducing the number of Map-Request messages.

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

For each Map-Reply record, the list of locators in a locator-set MUST appear in the same order for each ETR that originates a Map-Reply message. The locator-set MUST be sorted in order of ascending IP address where an IPv4 locator address is considered numerically 'less than' an IPv6 locator address.

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

6.1.5.1. Traffic Redirection with Coarse EID-Prefixes

When an ETR is misconfigured or compromised, it could return coarse EID-prefixes in Map-Reply messages it sends. The EID-prefix could

cover EID-prefixes which are allocated to other sites redirecting their traffic to the locators of the compromised site.

To solve this problem, there are two basic solutions that could be used. The first is to have Map-Servers proxy-map-reply on behalf of ETRs so their registered EID-prefixes are the ones returned in Map-Replies. Since the interaction between an ETR and Map-Server is secured with shared-keys, it is easier for an ETR to detect misbehavior. The second solution is to have ITRs and PITRs cache EID-prefixes with mask-lengths that are greater than or equal to a configured prefix length. This limits the damage to a specific width of any EID-prefix advertised, but needs to be coordinated with the allocation of site prefixes. These solutions can be used independently or at the same time.

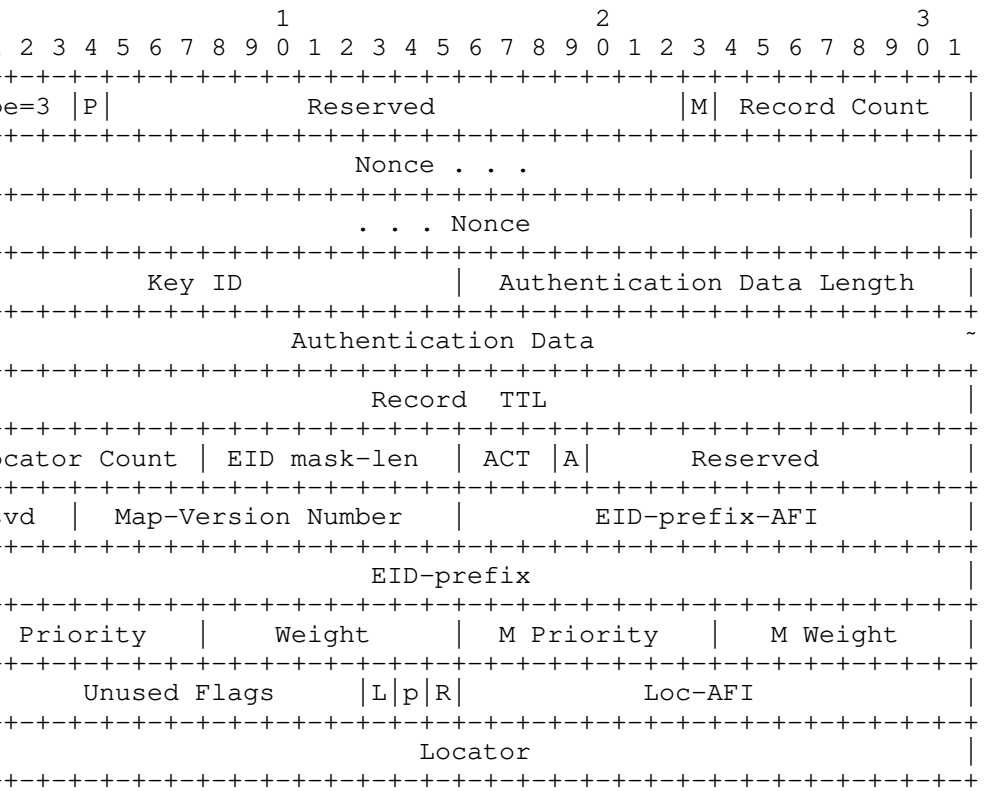
At the time of this writing, other approaches are being considered and researched.

6.1.6. Map-Register Message Format

The usage details of the Map-Register message can be found in specification [LISP-MS]. This section solely defines the message format.

The message is sent in UDP with a destination UDP port of 4342 and a randomly selected UDP source port number.

The Map-Register message format is:



Packet field descriptions:

Type: 3 (Map-Register)

P: This is the proxy-map-reply bit, when set to 1 an ETR sends a Map-Register message requesting for the Map-Server to proxy Map-Reply. The Map-Server will send non-authoritative Map-Replies on behalf of the ETR. Details on this usage can be found in [LISP-MS].

Reserved: It MUST be set to 0 on transmit and MUST be ignored on receipt.

M: This is the want-map-notify bit, when set to 1 an ETR is requesting for a Map-Notify message to be returned in response to sending a Map-Register message. The Map-Notify message sent by a Map-Server is used to an acknowledge receipt of a Map-Register message.

Record Count: The number of records in this Map-Register message. A record is comprised of that portion of the packet labeled 'Record' above and occurs the number of times equal to Record count.

Nonce: This 8-octet Nonce field is set to 0 in Map-Register messages. Since the Map-Register message is authenticated, the nonce field is not currently used for any security function but may be in the future as part of an anti-replay solution.

Key ID: A configured ID to find the configured Message Authentication Code (MAC) algorithm and key value used for the authentication function. See Section 14.4 for codepoint assignments.

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

Authentication Data: The message digest used from the output of the Message Authentication Code (MAC) algorithm. The entire Map-Register payload is authenticated with this field preset to 0. After the MAC is computed, it is placed in this field. Implementations of this specification MUST include support for HMAC-SHA-1-96 [RFC2404] and support for HMAC-SHA-256-128 [RFC6234] is RECOMMENDED.

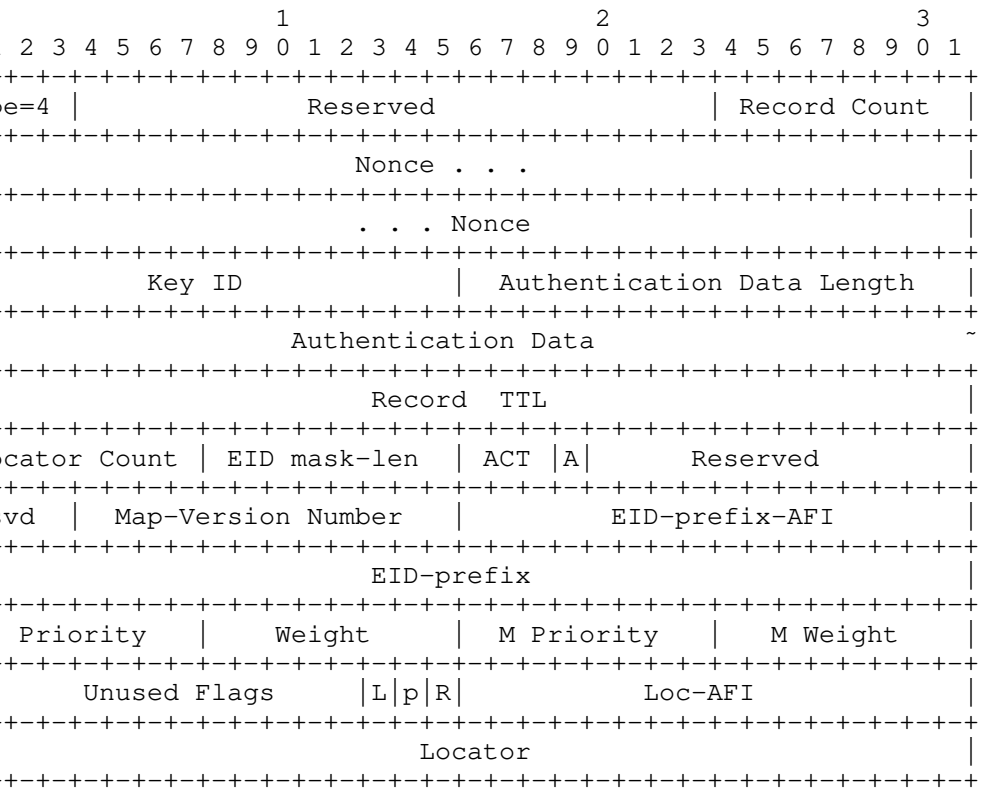
The definition of the rest of the Map-Register can be found in the Map-Reply section.

6.1.7. Map-Notify Message Format

The usage details of the Map-Notify message can be found in specification [LISP-MS]. This section solely defines the message format.

The message is sent inside a UDP packet with source and destination UDP ports equal to 4342.

The Map-Notify message format is:



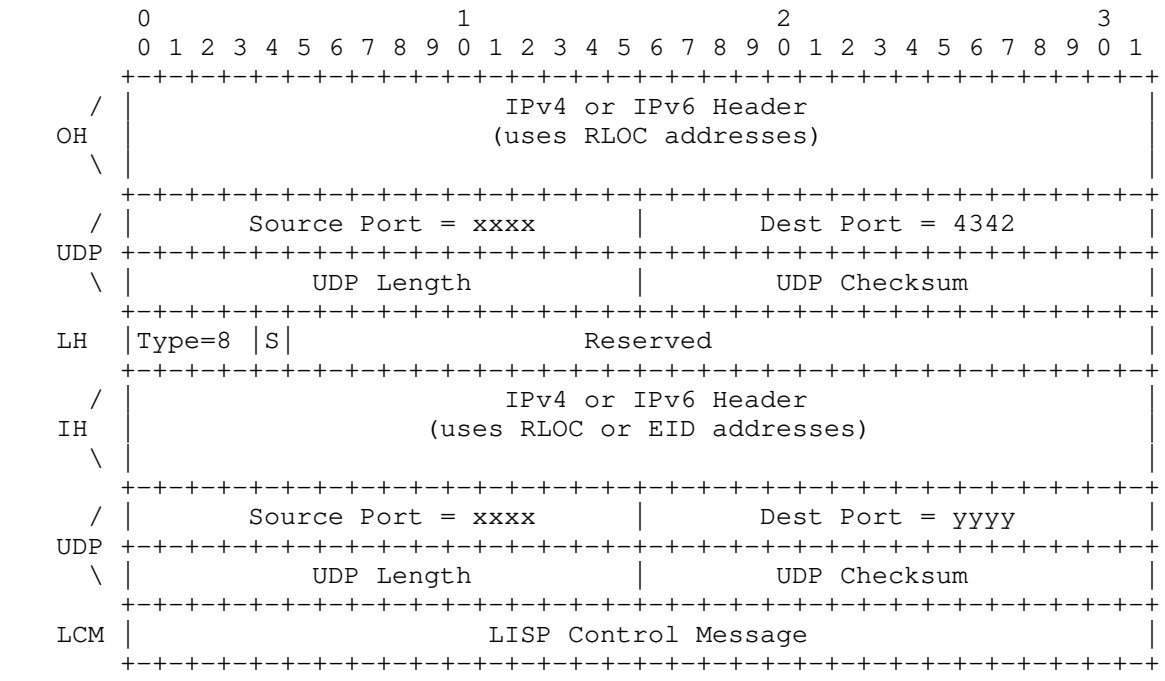
Packet field descriptions:

Type: 4 (Map-Notify)

The Map-Notify message has the same contents as a Map-Register message. See Map-Register section for field descriptions.

6.1.8. Encapsulated Control Message Format

An Encapsulated Control Message (ECM) is used to encapsulate control packets sent between xTRs and the mapping database system described in [LISP-MS].



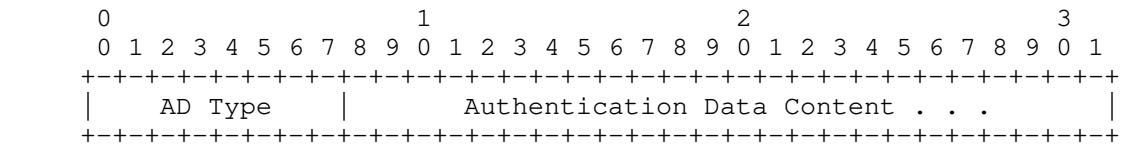
Packet header descriptions:

OH: The outer IPv4 or IPv6 header which uses RLOC addresses in the source and destination header address fields.

UDP: The outer UDP header with destination port 4342. The source port is randomly allocated. The checksum field MUST be non-zero.

LH: Type 8 is defined to be a "LISP Encapsulated Control Message" and what follows is either an IPv4 or IPv6 header as encoded by the first 4 bits after the reserved field.

S: This is the Security bit. When set to 1 the field following the Reserved field will have the following format. The detailed format of the Authentication Data Content is for further study.



IH: The inner IPv4 or IPv6 header which can use either RLOC or EID addresses in the header address fields. When a Map-Request is encapsulated in this packet format the destination address in this header is an EID.

UDP: The inner UDP header where the port assignments depends on the control packet being encapsulated. When the control packet is a Map-Request or Map-Register, the source port is ITR/PITR selected and the destination port is 4342. When the control packet is a Map-Reply, the source port is 4342 and the destination port is assigned from the source port of the invoking Map-Request. Port number 4341 MUST NOT be assigned to either port. The checksum field MUST be non-zero.

LCM: The format is one of the control message formats described in this section. At this time, only Map-Request messages are allowed to be encapsulated. And in the future, PIM Join-Prune messages [MLISP] might be allowed. Encapsulating other types of LISP control messages are for further study. When Map-Requests are sent for RLOC-probing purposes (i.e the probe-bit is set), they MUST NOT be sent inside Encapsulated Control Messages.

6.2. Routing Locator Selection

Both client-side and server-side may need control over the selection of RLOCs for conversations between them. This control is achieved by manipulating the Priority and Weight fields in EID-to-RLOC Map-Reply messages. Alternatively, RLOC information MAY be gleaned from received tunneled packets or EID-to-RLOC Map-Request messages.

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

- Server-side returns one RLOC. Client-side can only use one RLOC. Server-side has complete control of the selection.

- Server-side returns a list of RLOC where a subset of the list has the same best priority. 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.

- Server-side sets weight of 0 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 determining load distribution. Again, the client can use alternative RLOCs if the server-provided list of RLOCs are unreachable.

- Either side (more likely on 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 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.

- 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 which have a source EID that matches the EID-prefix of the verified entry.

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 or 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.

6.3. Routing Locator Reachability

Several mechanisms for determining RLOC reachability are currently defined:

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. An ITR may receive an ICMP Network or ICMP Host Unreachable message for an RLOC it is using. This indicates that the RLOC is likely down. Note, trusting ICMP messages may not be desirable but neither is ignoring them completely. Implementations are encouraged to follow current best practices in treating these conditions.

3. An ITR which participates in the global routing system can determine that an RLOC is down if no BGP RIB route exists that matches the RLOC IP address.

4. An ITR may receive an ICMP Port Unreachable message from a destination host. This occurs if an ITR attempts to use interworking [INTERWORK] and LISP-encapsulated data is sent to a non-LISP-capable site.

5. An ITR may receive a Map-Reply from an ETR in response to a previously sent Map-Request. The RLOC source of the Map-Reply is likely up since the ETR was able to send the Map-Reply to the ITR.

6. When an ETR receives an encapsulated packet from an ITR, the source RLOC from the outer header of the packet is likely up.

7. An ITR/ETR pair can use the Locator Reachability Algorithms described in this section, namely Echo-Noncing or RLOC-Probing.

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.

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.

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.

When ITRs receive ICMP Network or Host Unreachable messages as a method to determine unreachability, they will refrain from using Locators which are described in Locator lists of Map-Replies. However, using this approach is unreliable because many network operators turn off generation of ICMP Unreachable messages.

If an ITR does receive an ICMP Network or Host Unreachable message, it MAY originate its own ICMP Unreachable message destined for the host that originated the data packet the ITR encapsulated.

Also, BGP-enabled ITRs can unilaterally examine the RIB to see if a locator address from a locator-set in a mapping entry matches a prefix. If it does not find one and BGP is running in the Default Free Zone (DFZ), it can decide to not use the locator even though the

Locator Status Bits indicate the locator is up. In this case, the path from the ITR to the ETR that is assigned the locator is not available. More details are in [LOC-ID-ARCH].

Optionally, an ITR can send a Map-Request to a Locator and if a Map-Reply is returned, reachability of the Locator has been determined. Obviously, sending such probes increases the number of control messages originated by tunnel routers for active flows, so Locators are assumed to be reachable when they are advertised.

This assumption does create a dependency: Locator unreachability is detected by the receipt of ICMP Host Unreachable messages. When an Locator has been determined to be unreachable, it is not used for active traffic; this is the same as if it were listed in a Map-Reply with priority 255.

The ITR can test the reachability of the unreachable Locator by sending periodic Requests. Both Requests and Replies MUST be rate-limited. Locator reachability testing is never done with data packets since that increases the risk of packet loss for end-to-end sessions.

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 mechanisms to determine reachability.

6.3.1. 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 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 echo-nonce-request state. The time the ITR waits to process the

echoed nonce before it determines the path is unreachable is variable and 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 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 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 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 echo-nonce-request state, it can echo the ETR's nonce in the next set of packets that it encapsulates and then 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 which 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 regard the locator unreachable erroneously. An ITR SHOULD only set the E-bit in a 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.

Note that other locator reachability mechanisms are being researched and can be used to compliment or even override the Echo Nonce Algorithm. See next section for an example of control-plane probing.

6.3.2. RLOC Probing Algorithm

RLOC Probing is a method that an ITR or PITR can use to determine the reachability status of one or more locators that it has cached in a map-cache entry. The probe-bit of the Map-Request and Map-Reply messages are used for RLOC Probing.

RLOC probing is done in the control-plane on a timer basis where an ITR or PITR will originate a Map-Request destined to a locator address from one of its own locator addresses. A Map-Request used as

an RLOC-probe is NOT encapsulated and NOT sent to a Map-Server or on the ALT like one would when soliciting mapping data. The EID record encoded in the Map-Request is the EID-prefix of the map-cache entry cached by the ITR or PITR. The ITR may include a mapping data record for its own database mapping information which contains the local EID-prefixes and RLOCs for its site. RLOC-probes are sent periodically using a jittered timer interval.

When an ETR receives a Map-Request message with the probe-bit set, it returns a Map-Reply with the probe-bit set. The source address of the Map-Reply is set according to the procedure described in Section 6.1.5. The Map-Reply SHOULD contain mapping data for the EID-prefix contained in the Map-Request. This provides the opportunity for the ITR or PITR, which sent the RLOC-probe to get mapping updates if there were changes to the ETR's database mapping entries.

There are advantages and disadvantages of RLOC Probing. The greatest benefit of RLOC Probing is that it can handle many failure scenarios allowing the ITR to determine when the path to a specific locator is reachable or has become unreachable, thus providing a robust mechanism for switching to using another locator from the cached locator. RLOC Probing can also provide rough RTT estimates between a pair of locators which can be useful for network management purposes as well as for selecting low delay paths. The major disadvantage of RLOC Probing is in the number of control messages required and the amount of bandwidth used to obtain those benefits, especially if the requirement for failure detection times are very small.

Continued research and testing will attempt to characterize the tradeoffs of failure detection times versus message overhead.

6.4. 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 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. This problem is under investigation

with the expectation that experiments will tell us more. Note, this is not a new problem introduced by the LISP architecture. The problem exists today when a multi-homed site uses BGP to advertise its reachability upstream.

6.5. Routing Locator Hashing

When an ETR provides an EID-to-RLOC mapping in a Map-Reply message to a requesting ITR, the locator-set for the EID-prefix may contain different priority 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 can be used or the traditional 5-tuple hash which includes the source and destination addresses, source and destination TCP, UDP, or SCTP port numbers and the IP protocol number field or IPv6 next-protocol fields of a packet 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 which 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 which 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.

6.6. 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, we want to maintain this approach but need to provide a way for ETRs change their mappings and inform the sites that are currently communicating with the ETR site using such mappings.

When adding a new locator record in lexicographic order to the end of a locator-set, it is easy to update mappings. We assume new mappings will maintain the same locator ordering as the old mapping but 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 loc-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, the SMR procedure in Section 6.6.2 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 loc-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 loc-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 loc-status-bit settings can be efficiently packed.

We propose here three approaches for locator-set compaction, one operational and two protocol mechanisms. The operational approach uses a clock sweep method. The protocol approaches use the concept of Solicit-Map-Requests and Map-Versioning.

6.6.1. 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 time to live equal to the TTL in the Map-Reply.

6.6.2. Solicit-Map-Request (SMR)

Soliciting a Map-Request is a selective way for ETRs, at the site where mappings change, to control the rate they receive requests for Map-Reply messages. SMRs are also used to tell remote ITRs to update the mappings they have cached.

Since the ETRs don't keep track of remote ITRs that have cached their mappings, they do not know which ITRs need to have their mappings updated. As a result, an ETR will solicit Map-Requests (called an SMR message) from those sites to which it has been sending encapsulated data to for the last minute. In particular, an ETR will send an SMR an ITR to which it has recently sent encapsulated data.

An SMR message is simply a bit set in a Map-Request message. An ITR or PITR will send a Map-Request when they receive an SMR message. Both the SMR sender and the Map-Request responder MUST rate-limited these messages. Rate-limiting can be implemented as a global rate-limiter or one rate-limiter per SMR destination.

The following procedure shows how a SMR exchange occurs when a site is doing locator-set compaction for an EID-to-RLOC mapping:

1. When the database mappings in an ETR change, the ETRs at the site begin to send Map-Requests with the SMR bit set for each locator in each map-cache entry the ETR caches.

2. A remote ITR which receives the SMR message will schedule sending a Map-Request message to the source locator address of the SMR message or to the mapping database system. A newly allocated random nonce is selected and the EID-prefix used is the one copied from the SMR message. If the source locator is the only locator in the cached locator-set, the remote ITR SHOULD send a Map-Request to the database mapping system just in case the single locator has changed and may no longer be reachable to accept the Map-Request.

3. The remote ITR MUST rate-limit the Map-Request until it gets a Map-Reply while continuing to use the cached mapping. When Map Versioning is used, described in Section 6.6.3, an SMR sender can detect if an ITR is using the most up to date database mapping.

4. The ETRs at the site with the changed mapping will reply to the Map-Request with a Map-Reply message that has a nonce from the SMR-invoked Map-Request. The Map-Reply messages SHOULD be rate limited. This is important to avoid Map-Reply implosion.

5. The ETRs, at the site with the changed mapping, record the fact that the site that sent the Map-Request has received the new mapping data in the mapping cache entry for the remote site so the loc-status-bits are reflective of the new mapping for packets going to the remote site. The ETR then stops sending SMR messages.

Experimentation is in progress to determine the appropriate rate-limit parameters.

For security reasons an ITR MUST NOT process unsolicited Map-Replies. To avoid map-cache entry corruption by a third-party, a sender of an SMR-based Map-Request MUST be verified. If an ITR receives an SMR-based Map-Request and the source is not in the locator-set for the stored map-cache entry, then the responding Map-Request MUST be sent with an EID destination to the mapping database system. Since the mapping database system is more secure to reach an authoritative ETR, it will deliver the Map-Request to the authoritative source of the mapping data.

When an ITR receives an SMR-based Map-Request for which it does not

have a cached mapping for the EID in the SMR message, it MAY not send a SMR-invoked Map-Request. This scenario can occur when an ETR sends SMR messages to all locators in the locator-set it has stored in its map-cache but the remote ITRs that receive the SMR may not be sending packets to the site. There is no point in updating the ITRs until they need to send, in which case, they will send Map-Requests to obtain a map-cache entry.

6.6.3. 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 can stop to a removed locator and start 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 the Destination Map-Version Number is less than the current version for its mapping, the SMR procedure described in Section 6.6.2 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 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 can assure that all ETRs for a site registering to it will be Map-Version number synchronized.

See [VERSIONING] for a more detailed analysis and description of Database Map Versioning.

7. 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 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 value 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 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 a 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 Section 12.

8. Deployment Scenarios

This section will explore how and where ITRs and ETRs can be deployed and will discuss the pros and cons of each deployment scenario. For a more detailed deployment recommendation, refer to [LISP-DEPLOY].

There are two basic deployment trade-offs to consider: centralized versus distributed caches and flat, recursive, or re-encapsulating tunneling. When deciding on centralized versus distributed caching, the following issues should be considered:

- Are the tunnel routers spread out so that the caches are spread across all the memories of each router? A centralized cache is when an ITR keeps a cache for all the EIDs it is encapsulating to. The packet takes a direct path to the destination locator. A distributed cache is when an ITR needs help from other re-encapsulating routers because it does not store all the cache entries for the EIDs is it encapsulating to. So the packet takes a path through re-encapsulating routers that have a different set of cache entries.

- Should management "touch points" be minimized by choosing few tunnel routers, just enough for redundancy?

- In general, using more ITRs doesn't increase management load, since caches are built and stored dynamically. On the other hand, more ETRs does require more management since EID-prefix-to-RLOC mappings need to be explicitly configured.

When deciding on flat, recursive, or re-encapsulation tunneling, the following issues should be considered:

- Flat tunneling implements a single tunnel between source site and destination site. This generally offers better paths between sources and destinations with a single tunnel path.

- Recursive tunneling is when tunneled traffic is again further encapsulated in another tunnel, either to implement VPNs or to perform Traffic Engineering. When doing VPN-based tunneling, the site has some control since the site is prepending a new tunnel header. In the case of TE-based tunneling, the site may have control if it is prepending a new tunnel header, but if the site's ISP is doing the TE, then the site has no control. Recursive tunneling generally will result in suboptimal paths but at the benefit of steering traffic to resource available parts of the network.

- The technique of re-encapsulation ensures that packets only require one tunnel header. So if a packet needs to be rerouted, it is first decapsulated by the ETR and then re-encapsulated with a new tunnel header using a new RLOC.

The next sub-sections will survey where tunnel routers can reside in the network.

8.1. First-hop/Last-hop Tunnel Routers

By locating tunnel routers close to hosts, the EID-prefix set is at the granularity of an IP subnet. So at the expense of more EID-prefix-to-RLOC sets for the site, the caches in each tunnel router can remain relatively small. But caches always depend on the number of non-aggregated EID destination flows active through these tunnel routers.

With more tunnel routers doing encapsulation, the increase in control traffic grows as well: since the EID-granularity is greater, more Map-Requests and Map-Replies are traveling between more routers.

The advantage of placing the caches and databases at these stub routers is that the products deployed in this part of the network have better price-memory ratios then their core router counterparts. Memory is typically less expensive in these devices and fewer routes are stored (only IGP routes). These devices tend to have excess capacity, both for forwarding and routing state.

LISP functionality can also be deployed in edge switches. These devices generally have layer-2 ports facing hosts and layer-3 ports facing the Internet. Spare capacity is also often available in these devices as well.

8.2. Border/Edge Tunnel Routers

Using customer-edge (CE) routers for tunnel endpoints allows the EID space associated with a site to be reachable via a small set of RLOCs assigned to the CE routers for that site. This is the default behavior envisioned in the rest of this specification.

This offers the opposite benefit of the first-hop/last-hop tunnel router scenario: the number of mapping entries and network management touch points are reduced, allowing better scaling.

One disadvantage is that less of the network's resources are used to reach host endpoints thereby centralizing the point-of-failure domain and creating network choke points at the CE router.

Note that more than one CE router at a site can be configured with the same IP address. In this case an RLOC is an anycast address. This allows resilience between the CE routers. That is, if a CE router fails, traffic is automatically routed to the other routers using the same anycast address. However, this comes with the disadvantage where the site cannot control the entrance point when the anycast route is advertised out from all border routers. Another disadvantage of using anycast locators is the limited advertisement scope of /32 (or /128 for IPv6) routes.

8.3. ISP Provider-Edge (PE) Tunnel Routers

Use of ISP PE routers as tunnel endpoint routers is not the typical deployment scenario envisioned in the specification. This section attempts to capture some of reasoning behind this preference of implementing LISP on CE routers.

Use of ISP PE routers as tunnel endpoint routers gives an ISP, rather than a site, control over the location of the egress tunnel endpoints. That is, the ISP can decide if the tunnel endpoints are in the destination site (in either CE routers or last-hop routers within a site) or at other PE edges. The advantage of this case is that two tunnel headers can be avoided. By having the PE be the first router on the path to encapsulate, it can choose a TE path first, and the ETR can decapsulate and re-encapsulate for a tunnel to the destination end site.

An obvious disadvantage is that the end site has no control over where its packets flow or the RLOCs used. Other disadvantages include the difficulty in synchronizing path liveness updates between CE and PE routers.

As mentioned in earlier sections a combination of these scenarios is possible at the expense of extra packet header overhead, if both site and provider want control, then recursive or re-encapsulating tunnels are used.

8.4. LISP Functionality with Conventional NATs

LISP routers can be deployed behind Network Address Translator (NAT) devices to provide the same set of packet services hosts have today when they are addressed out of private address space.

It is important to note that a locator address in any LISP control message MUST be a globally routable address and therefore SHOULD NOT contain [RFC1918] addresses. If a LISP router is configured with private addresses, they MUST be used only in the outer IP header so the NAT device can translate properly. Otherwise, EID addresses MUST

be translated before encapsulation is performed. Both NAT translation and LISP encapsulation functions could be co-located in the same device.

More details on LISP address translation can be found in [INTERWORK].

8.5. Packets Egressing a LISP Site

When a LISP site is using two ITRs for redundancy, the failure of one ITR will likely shift outbound traffic to the second. This second ITR's cache may not not be populated with the same EID-to-RLOC mapping entries as the first. If this second ITR does not have these mappings, traffic will be dropped while the mappings are retrieved from the mapping system. The retrieval of these messages may increase the load of requests being sent into the mapping system. Deployment and experimentation will determine whether this issue requires more attention.

9. Traceroute Considerations

When a source host in a LISP site initiates a traceroute to a destination host in another LISP site, it is highly desirable for it to see the entire path. Since packets are encapsulated from ITR to ETR, the hop across the tunnel could be viewed as a single hop. However, LISP traceroute will provide the entire path so the user can see 3 distinct segments of the path from a source LISP host to a destination LISP host:

Segment 1 (in source LISP site based on EIDs):

```
source-host ---> first-hop ... next-hop ---> ITR
```

Segment 2 (in the core network based on RLOCs):

```
ITR ---> next-hop ... next-hop ---> ETR
```

Segment 3 (in the destination LISP site based on EIDs):

```
ETR ---> next-hop ... last-hop ---> destination-host
```

For segment 1 of the path, ICMP Time Exceeded messages are returned in the normal manner as they are today. The ITR performs a TTL decrement and test for 0 before encapsulating. So the ITR hop is seen by the traceroute source has an EID address (the address of site-facing interface).

For segment 2 of the path, ICMP Time Exceeded messages are returned to the ITR because the TTL decrement to 0 is done on the outer header, so the destination of the ICMP messages are to the ITR RLOC address, the source RLOC address of the encapsulated traceroute packet. The ITR looks inside of the ICMP payload to inspect the traceroute source so it can return the ICMP message to the address of the traceroute client as well as retaining the core router IP address in the ICMP message. This is so the traceroute client can display the core router address (the RLOC address) in the traceroute output. The ETR returns its RLOC address and responds to the TTL decrement to 0 like the previous core routers did.

For segment 3, the next-hop router downstream from the ETR will be decrementing the TTL for the packet that was encapsulated, sent into the core, decapsulated by the ETR, and forwarded because it isn't the final destination. If the TTL is decremented to 0, any router on the path to the destination of the traceroute, including the next-hop router or destination, will send an ICMP Time Exceeded message to the source EID of the traceroute client. The ICMP message will be

encapsulated by the local ITR and sent back to the ETR in the originated traceroute source site, where the packet will be delivered to the host.

9.1. IPv6 Traceroute

IPv6 traceroute follows the procedure described above since the entire traceroute data packet is included in ICMP Time Exceeded message payload. Therefore, only the ITR needs to pay special attention for forwarding ICMP messages back to the traceroute source.

9.2. IPv4 Traceroute

For IPv4 traceroute, we cannot follow the above procedure since IPv4 ICMP Time Exceeded messages only include the invoking IP header and 8 octets that follow the IP header. Therefore, when a core router sends an IPv4 Time Exceeded message to an ITR, all the ITR has in the ICMP payload is the encapsulated header it prepended followed by a UDP header. The original invoking IP header, and therefore the identity of the traceroute source is lost.

The solution we propose to solve this problem is to cache traceroute IPv4 headers in the ITR and to match them up with corresponding IPv4 Time Exceeded messages received from core routers and the ETR. The ITR will use a circular buffer for caching the IPv4 and UDP headers of traceroute packets. It will select a 16-bit number as a key to find them later when the IPv4 Time Exceeded messages are received. When an ITR encapsulates an IPv4 traceroute packet, it will use the 16-bit number as the UDP source port in the encapsulating header. When the ICMP Time Exceeded message is returned to the ITR, the UDP header of the encapsulating header is present in the ICMP payload thereby allowing the ITR to find the cached headers for the traceroute source. The ITR puts the cached headers in the payload and sends the ICMP Time Exceeded message to the traceroute source retaining the source address of the original ICMP Time Exceeded message (a core router or the ETR of the site of the traceroute destination).

The signature of a traceroute packet comes in two forms. The first form is encoded as a UDP message where the destination port is inspected for a range of values. The second form is encoded as an ICMP message where the IP identification field is inspected for a well-known value.

9.3. Traceroute using Mixed Locators

When either an IPv4 traceroute or IPv6 traceroute is originated and the ITR encapsulates it in the other address family header, you

cannot get all 3 segments of the traceroute. Segment 2 of the traceroute can not be conveyed to the traceroute source since it is expecting addresses from intermediate hops in the same address format for the type of traceroute it originated. Therefore, in this case, segment 2 will make the tunnel look like one hop. All the ITR has to do to make this work is to not copy the inner TTL to the outer, encapsulating header's TTL when a traceroute packet is encapsulated using an RLOC from a different address family. This will cause no TTL decrement to 0 to occur in core routers between the ITR and ETR.

10. Mobility Considerations

There are several kinds of mobility of which only some might be of concern to LISP. Essentially they are as follows.

10.1. Site Mobility

A site wishes to change its attachment points to the Internet, and its LISP Tunnel Routers will have new RLOCs when it changes upstream providers. Changes in EID-RLOC mappings for sites are expected to be handled by configuration, outside of the LISP protocol.

10.2. Slow Endpoint Mobility

An individual endpoint wishes to move, but is not concerned about maintaining session continuity. Renumbering is involved. LISP can help with the issues surrounding renumbering [RFC4192] [LISA96] by decoupling the address space used by a site from the address spaces used by its ISPs. [RFC4984]

10.3. Fast Endpoint Mobility

Fast endpoint mobility occurs when an endpoint moves relatively rapidly, changing its IP layer network attachment point. Maintenance of session continuity is a goal. This is where the Mobile IPv4 [RFC5944] and Mobile IPv6 [RFC6275] [RFC4866] mechanisms are used, and primarily where interactions with LISP need to be explored.

The problem is that as an endpoint moves, it may require changes to the mapping between its EID and a set of RLOCs for its new network location. When this is added to the overhead of mobile IP binding updates, some packets might be delayed or dropped.

In IPv4 mobility, when an endpoint is away from home, packets to it are encapsulated and forwarded via a home agent which resides in the home area the endpoint’s address belongs to. The home agent will encapsulate and forward packets either directly to the endpoint or to a foreign agent which resides where the endpoint has moved to. Packets from the endpoint may be sent directly to the correspondent node, may be sent via the foreign agent, or may be reverse-tunneled back to the home agent for delivery to the mobile node. As the mobile node’s EID or available RLOC changes, LISP EID-to-RLOC mappings are required for communication between the mobile node and the home agent, whether via foreign agent or not. As a mobile endpoint changes networks, up to three LISP mapping changes may be required:

- The mobile node moves from an old location to a new visited network location and notifies its home agent that it has done so. The Mobile IPv4 control packets the mobile node sends pass through one of the new visited network's ITRs, which needs an EID-RLOC mapping for the home agent.

- The home agent might not have the EID-RLOC mappings for the mobile node's "care-of" address or its foreign agent in the new visited network, in which case it will need to acquire them.

- When packets are sent directly to the correspondent node, it may be that no traffic has been sent from the new visited network to the correspondent node's network, and the new visited network's ITR will need to obtain an EID-RLOC mapping for the correspondent node's site.

In addition, if the IPv4 endpoint is sending packets from the new visited network using its original EID, then LISP will need to perform a route-returnability check on the new EID-RLOC mapping for that EID.

In IPv6 mobility, packets can flow directly between the mobile node and the correspondent node in either direction. The mobile node uses its "care-of" address (EID). In this case, the route-returnability check would not be needed but one more LISP mapping lookup may be required instead:

- As above, three mapping changes may be needed for the mobile node to communicate with its home agent and to send packets to the correspondent node.

- In addition, another mapping will be needed in the correspondent node's ITR, in order for the correspondent node to send packets to the mobile node's "care-of" address (EID) at the new network location.

When both endpoints are mobile the number of potential mapping lookups increases accordingly.

As a mobile node moves there are not only mobility state changes in the mobile node, correspondent node, and home agent, but also state changes in the ITRs and ETRs for at least some EID-prefixes.

The goal is to support rapid adaptation, with little delay or packet loss for the entire system. Also IP mobility can be modified to require fewer mapping changes. In order to increase overall system performance, there may be a need to reduce the optimization of one area in order to place fewer demands on another.

In LISP, one possibility is to "glean" information. When a packet arrives, the ETR could examine the EID-RLOC mapping and use that mapping for all outgoing traffic to that EID. It can do this after performing a route-returnability check, to ensure that the new network location does have a internal route to that endpoint. However, this does not cover the case where an ITR (the node assigned the RLOC) at the mobile-node location has been compromised.

Mobile IP packet exchange is designed for an environment in which all routing information is disseminated before packets can be forwarded. In order to allow the Internet to grow to support expected future use, we are moving to an environment where some information may have to be obtained after packets are in flight. Modifications to IP mobility should be considered in order to optimize the behavior of the overall system. Anything which decreases the number of new EID-RLOC mappings needed when a node moves, or maintains the validity of an EID-RLOC mapping for a longer time, is useful.

10.4. Fast Network Mobility

In addition to endpoints, a network can be mobile, possibly changing xTRs. A "network" can be as small as a single router and as large as a whole site. This is different from site mobility in that it is fast and possibly short-lived, but different from endpoint mobility in that a whole prefix is changing RLOCs. However, the mechanisms are the same and there is no new overhead in LISP. A map request for any endpoint will return a binding for the entire mobile prefix.

If mobile networks become a more common occurrence, it may be useful to revisit the design of the mapping service and allow for dynamic updates of the database.

The issue of interactions between mobility and LISP needs to be explored further. Specific improvements to the entire system will depend on the details of mapping mechanisms. Mapping mechanisms should be evaluated on how well they support session continuity for mobile nodes.

10.5. LISP Mobile Node Mobility

A mobile device can use the LISP infrastructure to achieve mobility by implementing the LISP encapsulation and decapsulation functions and acting as a simple ITR/ETR. By doing this, such a "LISP mobile node" can use topologically-independent EID IP addresses that are not advertised into and do not impose a cost on the global routing system. These EIDs are maintained at the edges of the mapping system (in LISP Map-Servers and Map-Resolvers) and are provided on demand to only the correspondents of the LISP mobile node.

Refer to the LISP Mobility Architecture specification [LISP-MN] for more details.

11. 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, determines 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, will use the same group address as the destination Routing Locator.

Having said that, only the source EID and source Routing Locator needs to be dealt with. Therefore, an ITR merely needs to put its own IP address in the source Routing Locator field when prepending the outer IP header. This source Routing Locator address, like any other Routing Locator address MUST be globally routable.

Therefore, an EID-to-RLOC mapping does not need to be performed by an ITR when a received data packet is a multicast data packet or when processing a source-specific Join (either by IGMPv3 or PIM). But the source Routing Locator is decided by the multicast routing protocol in a receiver site. That is, an EID to Routing Locator translation is done at control-time.

Another approach is to have the ITR not encapsulate a multicast packet and allow the host built packet to flow into the core even if the source address is allocated out of the EID namespace. If the RPF-Vector TLV [RFC5496] is used by PIM in the core, then core routers can RPF to the ITR (the Locator address which is injected into core routing) rather than the host source address (the EID address which is not injected into core routing).

To avoid any EID-based multicast state in the network core, the first approach is chosen for LISP-Multicast. Details for LISP-Multicast and Interworking with non-LISP sites is described in specification [MLISP].

12. Security Considerations

It is believed that most of the security mechanisms will be part of the mapping database service when using control plane procedures for obtaining EID-to-RLOC mappings. For data plane triggered mappings, as described in this specification, protection is provided against ETR spoofing by using Return-Routability (see Section 3) mechanisms evidenced by the use of a 24-bit Nonce field in the LISP encapsulation header and a 64-bit Nonce field in the LISP control message.

The nonce, coupled with the ITR accepting only solicited Map-Replies provides a basic level of security, in many ways similar to the security experienced in the current Internet routing system. It is hard for off-path attackers to launch attacks against these LISP mechanisms, as they do not have the nonce values. Sending a large number of packets to accidentally find the right nonce value is possible, but would already by itself be a denial-of-service attack. On-path attackers can perform far more serious attacks, but on-path attackers can launch serious attacks in the current Internet as well, including eavesdropping, blocking or redirecting traffic. See more discussion on this topic in Section 6.1.5.1.

LISP does not rely on a PKI or a more heavy weight authentication system. These systems challenge the scalability of LISP which was a primary design goal.

DoS attack prevention will depend on implementations rate-limiting Map-Requests and Map-Replies to the control plane as well as rate-limiting the number of data-triggered Map-Replies.

An incorrectly implemented or malicious ITR might choose to ignore the priority and weights provided by the ETR in its Map-Reply. This traffic steering would be limited to the traffic that is sent by this ITR's site, and no more severe than if the site initiated a bandwidth DoS attack on (one of) the ETR's ingress links. The ITR's site would typically gain no benefit from not respecting the weights, and would likely to receive better service by abiding by them.

To deal with map-cache exhaustion attempts in an ITR/PITR, the implementation should consider putting a maximum cap on the number of entries stored with a reserve list for special or frequently accessed sites. This should be a configuration policy control set by the network administrator who manages ITRs and PITRs. When overlapping EID-prefixes occur across multiple map-cache entries, the integrity of the set must be wholly maintained. So if a more-specific entry cannot be added due to reaching the maximum cap, then none of the less specifics should be stored in the map-cache.

Given that the ITR/PITR maintains a cache of EID-to-RLOC mappings, cache sizing and maintenance is an issue to be kept in mind during implementation. It is a good idea to have instrumentation in place to detect thrashing of the cache. Implementation experimentation will be used to determine which cache management strategies work best. In general, it is difficult to defend against cache trashing attacks. It should be noted that an undersized cache in an ITR/PITR not only causes adverse affect on the site or region they support, but may also cause increased Map-Request load on the mapping system.

"Piggybacked" mapping data discussed in Section 6.1.3 specifies how to handle such mappings and includes the possibility for an ETR to temporarily accept such a mapping before verification when running in "trusted" environments. In such cases, there is a potential threat that a fake mapping could be inserted (even if only for a short period) into a map-cache. As noted in Section 6.1.3, an ETR MUST be specifically configured to run in such a mode and might usefully only consider some specific ITRs as also running in that same trusted environment.

There is a security risk implicit in the fact that ETRs generate the EID prefix to which they are responding. An ETR can claim a shorter prefix than it is actually responsible for. Various mechanisms to ameliorate or resolve this issue will be examined in the future, [LISP-SEC].

Spoofing of inner header addresses of LISP encapsulated packets is possible like with any tunneling mechanism. ITRs MUST verify the source address of a packet to be an EID that belongs to the site's EID-prefix range prior to encapsulation. An ETR must only decapsulate and forward datagrams with an inner header destination that matches one of its EID-prefix ranges. If, upon receipt and decapsulation, the destination EID of a datagram does not match one of the ETR's configured EID-prefixes, the ETR MUST drop the datagram. If a LISP encapsulated packet arrives at an ETR, it SHOULD compare the inner header source EID address and the outer header source RLOC address with the mapping that exists in the mapping database. Then when spoofing attacks occur, the outer header source RLOC address can be used to trace back the attack to the source site, using existing operational tools.

This experimental specification does not address automated key management (AKM). BCP 107 provides guidance in this area. In addition, at the time of this writing, substantial work is being undertaken to improve security of the routing system [KARP], [RPKI], [BGP-SEC], [LISP-SEC]. Future work on LISP should address BCP-107 as well as other open security considerations, which may require changes to this specification.

13. Network Management Considerations

Considerations for Network Management tools exist so the LISP protocol suite can be operationally managed. The mechanisms can be found in [LISP-MIB] and [LISP-LIG].

14. IANA Considerations

This section provides guidance to the Internet Assigned Numbers Authority (IANA) regarding registration of values related to the LISP specification, in accordance with BCP 26 and RFC 5226 [RFC5226].

There are four name spaces in LISP that require registration:

- LISP IANA registry allocations should not be made for purposes unrelated to LISP routing or transport protocols.
- The following policies are used here with the meanings defined in BCP 26: "Specification Required", "IETF Review", "Experimental Use", "First Come First Served".

14.1. LISP ACT and Flag Fields

New ACT values (Section 6.1.4) can be allocated through IETF review or IESG approval. Four values have already been allocated by this specification (Section 6.1.4).

In addition, the LISP protocol has a number of flag and reserved fields, such as the LISP header flags field (Section 5.3). New bits for flags can be taken into use from these fields through IETF review or IESG approval, but these need not be managed by IANA.

14.2. LISP Address Type Codes

LISP Address [LCAF] type codes have a range from 0 to 255. New type codes MUST be allocated consecutively starting at 0. Type Codes 0 - 127 are to be assigned by IETF review or IESG approval.

Type Codes 128 - 255 are available on a First Come First Served policy.

This registry, initially empty, is constructed for future-use experimental work of LCAF values. See [LCAF] for details for other possible unapproved address encodings. The unapproved LCAF encodings are an area for further study and experimentation.

14.3. LISP UDP Port Numbers

The IANA registry has allocated UDP port numbers 4341 and 4342 for lisp-data and lisp-control operation, respectively. IANA is requested to update the description for udp ports 4341 and 4342 as follows:

```
lisp-data      4341 udp    LISP Data Packets
lisp-control   4342 udp    LISP Control Packets
```

14.4. LISP Key ID Numbers

The following Key ID values are defined by this specification as used in any packet type that references a Key ID field:

Name	Number	Defined in
None	0	n/a
HMAC-SHA-1-96	1	[RFC2404]
HMAC-SHA-256-128	2	[RFC6234]

Number values are in the range of 0 to 65355. The allocation of values is on a first come first serve basis.

15. Known Open Issues and Areas of Future Work

As an experimental specification, this work is, by definition, incomplete. Specific areas where additional experience and work are needed include:

- At present, only [ALT] is defined for implementing a database of EID-to-RLOC mapping information. Additional research on other mapping database systems is strongly encouraged.

- Failure and recovery of LISP site partitioning (see Section 6.4), in the presence of redundant configuration (see Section 8.5) needs further research and experimentation.

- The characteristics of map-cache management under exceptional conditions, such as denial-of-service attacks are not fully understood. Further experience is needed to determine whether current caching methods are practical or in need of further development. In particular, the performance, scaling and security characteristics of the map-cache will be discovered as part of this experiment. Performance metrics to be observed are packet reordering associated with the LISP data probe and loss of the first packet in a flow associated with map-caching. The impact of these upon TCP will be observed. See Section 12 for additional thoughts and considerations.

- Preliminary work has been done to ensure that sites employing LISP can interconnect with the rest of the Internet. This work is documented in [INTERWORK], but further experimentation and experience is needed.

- At present, no mechanism for automated key management for message authentication is defined. Addressing automated key management is necessary before this specification could be developed into a standards track RFC. See Section 12 for further details regarding security considerations.

- In order to maintain security and stability, Internet Protocols typically isolate the control and data planes. Therefore, user activity cannot cause control plane state to be created or destroyed. LISP does not maintain this separation. The degree to which the loss of separation impacts security and stability is a topic for experimental observation.

- LISP allows for different mapping database systems to be used. While only one [ALT] is currently well-defined, each mapping database will likely have some impact on the security of the EID-to-RLOC mappings. How each mapping database system's security

properties impact on LISP overall is for further study.

- An examination of the implications of LISP on Internet traffic, applications, routers, and security is needed. This will help to understand the consequences for network stability, routing protocol function, routing scalability, migration and backward compatibility, and implementation scalability (as influenced by additional protocol components, additional state, and additional processing for encapsulation, decapsulation, liveness).

- Experiments need to verify that LISP produces no significant change in the behavior of protocols run between end-systems over a LISP infrastructure versus being run directly between those same end-systems.

- Experiments need to verify that the issues raised in the Critique section of [RFC6115] are either insignificant or have been addressed by updates to the LISP protocol.

Other LISP documents may also include open issues and areas for future work.

16. References

16.1. Normative References

[ALT] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "LISP Alternative Topology (LISP-ALT)", draft-ietf-lisp-alt-10.txt (work in progress).

[LISP-MS] Farinacci, D. and V. Fuller, "LISP Map Server", draft-ietf-lisp-ms-16.txt (work in progress).

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2404] Madson, C. and R. Glenn, "The Use of HMAC-SHA-1-96 within ESP and AH", RFC 2404, November 1998.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.

[RFC3232] Reynolds, J., "Assigned Numbers: RFC 1700 is Replaced by an On-line Database", RFC 3232, January 2002.

[RFC4086] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

[RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, August 2006.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5496] Wijnands, IJ., Boers, A., and E. Rosen, "The Reverse Path Forwarding (RPF) Vector TLV", RFC 5496, March 2009.

[RFC5944] Perkins, C., "IP Mobility Support for IPv4, Revised", RFC 5944, November 2010.

[RFC6115] Li, T., "Recommendation for a Routing Architecture", RFC 6115, February 2011.

[RFC6234] Eastlake, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, May 2011.

[RFC6275] Perkins, C., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, July 2011.

[VERSIONING] Iannone, L., Saucez, D., and O. Bonaventure, "LISP Mapping Versioning", draft-ietf-lisp-map-versioning-09.txt (work in progress).

16.2. Informative References

[AFI] IANA, "Address Family Indicators (AFIs)", ADDRESS FAMILY NUMBERS http://www.iana.org/assignments/address-family-numbers.

[AFI-REGISTRY] IANA, "Address Family Indicators (AFIs)", ADDRESS FAMILY NUMBER registry http://www.iana.org/assignments/address-family-numbers/address-family-numbers.xml#address-family-numbers-1.

[BGP-SEC] Lepinski, M., "An Overview of BGPSEC", draft-lepinski-bgpsec-overview-00.txt (work in progress), March 2011.

[CHIAPPA] Chiappa, J., "Endpoints and Endpoint names: A Proposed Enhancement to the Internet Architecture", Internet-Draft http://www.chiappa.net/~jnc/tech/endpoints.txt.

[CONS] Farinacci, D., Fuller, V., and D. Meyer, "LISP-CONS: A Content distribution Overlay Network Service for LISP", draft-meyer-lisp-cons-04.txt (work in progress).

[EMACS] Brim, S., Farinacci, D., Meyer, D., and J. Curran, "EID Mappings Multicast Across Cooperating Systems for LISP", draft-curran-lisp-emacs-00.txt (work in progress).

[INTERWORK]
Lewis, D., Meyer, D., Farinacci, D., and V. Fuller, "Interworking LISP with IPv4 and IPv6", draft-ietf-lisp-interworking-06.txt (work in progress).

[KARP] Lebovitz, G. and M. Bhatia, "Keying and Authentication for Routing Protocols (KARP)Design Guidelines", draft-ietf-karp-design-guide-06.txt (work in progress), October 2011.

[LCAF] Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format", draft-ietf-lisp-lcaf-00.txt (work in progress).

[LISA96] Lear, E., Katinsky, J., Coffin, J., and D. Tharp, "Renumbering: Threat or Menace?", Usenix .

[LISP-DEPLOY]
Jakab, L., Coras, F., Domingo-Pascual, J., and D. Lewis, "LISP Network Element Deployment Considerations", draft-ietf-lisp-deployment-05.txt (work in progress).

[LISP-LIG]
Farinacci, D. and D. Meyer, "LISP Internet Groper (LIG)", draft-ietf-lisp-lig-06.txt (work in progress).

[LISP-MAIN]
Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol (LISP)", draft-farinacci-lisp-12.txt (work in progress).

[LISP-MIB]
Schudel, G., Jain, A., and V. Moreno, "LISP MIB", draft-ietf-lisp-mib-07.txt (work in progress).

[LISP-MN] Farinacci, D., Fuller, V., Lewis, D., and D. Meyer, "LISP Mobility Architecture", draft-meyer-lisp-mn-08.txt (work in progress).

[LISP-SEC]
Maino, F., Ermagon, V., Cabellos, A., Sausez, D., and O. Bonaventure, "LISP-Security (LISP-SEC)", draft-ietf-lisp-sec-04.txt (work in progress).

[LOC-ID-ARCH]
Meyer, D. and D. Lewis, "Architectural Implications of Locator/ID Separation", draft-meyer-loc-id-implications-02.txt (work in progress).

[MLISP] Farinacci, D., Meyer, D., Zwiebel, J., and S. Venaas, "LISP for Multicast Environments", draft-ietf-lisp-multicast-14.txt (work in progress).

[NERD] Lear, E., "NERD: A Not-so-novel EID to RLOC Database", draft-lear-lisp-nerd-08.txt (work in progress).

[OPENLISP] Iannone, L. and O. Bonaventure, "OpenLISP Implementation Report", draft-iannone-openlisp-implementation-01.txt (work in progress).

[RADIR] Narten, T., "Routing and Addressing Problem Statement", draft-narten-radir-problem-statement-05.txt (work in progress).

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.

[RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC4192] Baker, F., Lear, E., and R. Droms, "Procedures for Renumbering an IPv6 Network without a Flag Day", RFC 4192, September 2005.

[RFC4866] Arkko, J., Vogt, C., and W. Haddad, "Enhanced Route Optimization for Mobile IPv6", RFC 4866, May 2007.

[RFC4984] Meyer, D., Zhang, L., and K. Fall, "Report from the IAB Workshop on Routing and Addressing", RFC 4984, September 2007.

[RPKI] Lepinski, M., "An Infrastructure to Support Secure Internet Routing", draft-ietf-sidr-arch-13.txt (work in progress), February 2011.

[UDP-TUNNELS] Eubanks, M. and P. Chimento, "UDP Checksums for Tunneled

Packets", draft-ietf-6man-udpchecksums-05.txt (work in progress), October 2012.

[UDP-ZERO]
Fairhurst, G. and M. Westerland, "IPv6 UDP Checksum Considerations", draft-ietf-6man-udpzero-07.txt (work in progress), October 2012.

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 review 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 discussion 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, Albert Cabellos, Fabio Maino, Victor Moreno, Chris White, Clarence Filsfils, and Alia Atlas.

This work originated in the Routing Research Group (RRG) of the IRTF. The individual submission [LISP-MAIN] was converted into this IETF LISP working group draft.

The LISP working group would like to give a special thanks to Jari Arkko, the Internet Area AD at the time the set of LISP documents were being prepared for IESG last call, for his meticulous review and detail commentary on the 7 working group last call drafts progressing toward experimental RFCs.

Appendix B. Document Change Log

B.1. Changes to draft-ietf-lisp-24.txt

- Posted November 2012 for final pre-RFC version.
- Move draft-ietf-6man-udpchecksums reference back to Informative References section.

B.2. Changes to draft-ietf-lisp-23.txt

- Posted May 2012 for final pre-RFC version.
- Move only the reference draft-ietf-6man-udpzero to the Informative References section. Leave the draft-ietf-6man-udpchecksums reference in the Normative References section. After talking to many people involved with this issue at Paris IETF, all thought this would be an acceptable change.
- Added text to IANA Considerations section 14.4 to reflect IANA comments about allocating Key-ID numbers.

B.3. Changes to draft-ietf-lisp-22.txt

- Posted February 2012 to reflect final DISCUSS comments from Adrian Farrel.

B.4. Changes to draft-ietf-lisp-21.txt

- Posted February 2012 to reflect DISCUSS comments from Adrian Farrel, Stewart Bryant, and Wesley Eddy.

B.5. Changes to draft-ietf-lisp-20.txt

- Posted January 2012 for resolution to Adrian Farrel's security comments as well as additions to the end of section 2, Elwyn Davies Gen-Art comments, and Ralph Droms' IANA and EID definition comments.

B.6. Changes to draft-ietf-lisp-19.txt

- Posted January 2012 for Stephen Farrell's comment resolution.

B.7. Changes to draft-ietf-lisp-18.txt

- Posted December 2011 after reflecting comments from IANA.

- Create reference to sections 5.4.1 and 5.4.2 about DF bit setting from section 5.3.

- Inserted two references for Route-Returnability and on-path attacks in Security Considerations section.

B.8. Changes to draft-ietf-lisp-17.txt

- Posted December 2011 after IETF last call comments.

- Make Map-Notify port assignment be 4342 in both source and destination ports. This change was agreed on and put in [LISP-MS] but was not updated in this spec.

B.9. Changes to draft-ietf-lisp-16.txt

- Posted October 2011 after AD review by Jari.

B.10. Changes to draft-ietf-lisp-15.txt

- Posted July 2011. Fixing IDnits errors.

- Change description on how to select a source address for RLOC-probe Map-Replies to refer to the "EID-to-RLOC Map-Reply Message" section.

B.11. Changes to draft-ietf-lisp-14.txt

- Post working group last call and pre-IESG last call review.

- Indicate that an ICMP Unreachable message should be sent when a packet matches a drop-based negative map-cache entry.

- Indicate how a map-cache set of overlapping EID-prefixes must maintain integrity when the map-cache maximum cap is reached.

- Add Joel's description for the definition of an EID, that the bit string value can be an RLOC for another device in abstract but the architecture allows it to be an EID of one device and the same value as an RLOC for another device.

- In the "Tunnel Encapsulation Details" section, indicate that 4 combinations of encapsulation are supported.

- Add what ETR should do for a Data-Probe when received for a destination EID outside of its EID-prefix range. This was added in the Data Probe definition section.

- Added text indicating that more-specific EID-prefixes must not be removed when less-specific entries stay in the map-cache. This is to preserve the integrity of the EID-prefix set.

- Add clarifying text in the Security Considerations section about how an ETR must not decapsulate and forward a packet that is not for its configured EID-prefix range.

B.12. Changes to draft-ietf-lisp-13.txt

- Posted June 2011 to complete working group last call.

- Tracker item 87. Put Yakov suggested wording in the EID-prefix definition section to reference [INTERWORK] and [LISP-DEPLOY] about discussion on transition and access mechanisms.

- Change "ITRs" to "ETRs" in the Locator Status Bit definition section and data packet description section per Damien's comment.

- Remove the normative reference to [LISP-SEC] when describing the S-bit in the ECM and Map-Reply headers.

- Tracker item 54. Added text from John Scudder in the "Packets Egressing a LISP Site" section.

- Add sentence to the "Reencapsulating Tunnel" definition about how reencapsulation loops can occur when not coordinating among multiple mapping database systems.

- Remove "In theory" from a sentence in the Security Considerations section.

- Remove Security Area Statement title and reword section with Eliot's provided text. The text was agreed upon by LISP-WG chairs and Security ADs.

- Remove word "potential" from the over-claiming paragraph of the Security Considerations section per Stephen's request.

- Wordsmithing and other editorial comments from Alia.

B.13. Changes to draft-ietf-lisp-12.txt

- Posted April 2011.

- Tracker item 87. Provided rewording how an EID-prefix can be reused in the definition section of "EID-prefix".

- Tracker item 95. Change "eliminate" to "defer" in section 4.1.
- Tracker item 110. Added that the Mapping Protocol Data field in the Map-Reply message is only used when needed by the particular Mapping Database System.
- Tracker item 111. Indicate that if an LSB that is associated with an anycast address, that there is at least one RLOC that is up.
- Tracker item 108. Make clear the R-bit does not define RLOC path reachability.
- Tracker item 107. Indicate that weights are relative to each other versus requiring an addition of up to 100%.
- Tracker item 46. Add a sentence how LISP products should be sized for the appropriate demand so cache thrashing is avoided.
- Change some references of RFC 5226 to [AFI] per Luigi.
- Per Luigi, make reference to "EID-AFI" consistent to "EID-prefix-AFI".
- Tracker item 66. Indicate that appending locators to a locator-set is done when the added locators are lexicographically greater than the previous ones in the set.
- Tracker item 87. Once again reword the definition of the EID-prefix to reflect recent comments.
- Tracker item 70. Added text to security section on what the implications could be if an ITR does not obey priority and weights from a Map-Reply message.
- Tracker item 54. Added text to the new section titled "Packets Egressing a LISP Site" to describe the implications when two or more ITRs exist at a site where only one ITR is used for egress traffic and when there is a shift of traffic to the others, how the map-cache will need to be populated in those new egress ITRs.
- Tracker item 33. Make more clear in the Routing Locator Selection section what an ITR should do when it sees an R-bit of 0 in a locator-record of a Map-Reply.
- Tracker item 33. Add paragraph to the EID Reachability section indicating that site partitioning is under investigation.

- Tracker item 58. Added last paragraph of Security Considerations section about how to protect inner header EID address spoofing attacks.

- Add suggested Sam text to indicate that all security concerns need not be addressed for moving document to Experimental RFC status. Put this in a subsection of the Security Considerations section.

B.14. Changes to draft-ietf-lisp-11.txt

- Posted March 30, 2011.

- Change IANA URL. The URL we had pointed to a general protocol numbers page.

- Added the "s" bit to the Map-Request to allow SMR-invoked Map-Requests to be sent to a MN ETR via the map-server.

- Generalize text for the definition of Reencapsuatling tunnels.

- Add paragraph suggested by Joel to explain how implementation experimentation will be used to determine the proper cache management techniques.

- Add Yakov provided text for the definition of "EID-to-RLOC "Database".

- Add reference in Section 8, Deployment Scenarios, to the draft-jakab-lisp-deploy-02.txt draft.

- Clarify sentence about no hardware changes needed to support LISP encapsulation.

- Add paragraph about what is the procedure when a locator is inserted in the middle of a locator-set.

- Add a definition for Locator Status Bits so we can emphasize they are used as a hint for router up/down status and not path reachability.

- Change "BGP RIB" to "RIB" per Clarence's comment.

- Fixed complaints by IDnits.

- Add subsection to Security Considerations section indicating how EID-prefix overclaiming in Map-Replies is for further study and add a reference to LISP-SEC.

B.15. Changes to draft-ietf-lisp-10.txt

- Posted March 2011.
- Add p-bit to Map-Request so there is documentary reasons to know when a PITR has sent a Map-Request to an ETR.
- Add Map-Notify message which is used to acknowledge a Map-Register message sent to a Map-Server.
- Add M-bit to the Map-Register message so an ETR that wants an acknowledgment for the Map-Register can request one.
- Add S-bit to the ECM and Map-Reply messages to describe security data that can be present in each message. Then refer to [LISP-SEC] for expansive details.
- Add Network Management Considerations section and point to the MIB and LIG drafts.
- Remove the word "simple" per Yakov's comments.

B.16. Changes to draft-ietf-lisp-09.txt

- Posted October 2010.
- Add to IANA Consideration section about the use of LCAF Type values that accepted and maintained by the IANA registry and not the LCAF specification.
- Indicate that implementations should be able to receive LISP control messages when either UDP port is 4342, so they can be robust in the face of intervening NAT boxes.
- Add paragraph to SMR section to indicate that an ITR does not need to respond to an SMR-based Map-Request when it has no map-cache entry for the SMR source's EID-prefix.

B.17. Changes to draft-ietf-lisp-08.txt

- Posted August 2010.
- In section 6.1.6, remove statement about setting TTL to 0 in Map-Register messages.
- Clarify language in section 6.1.5 about Map-Replying to Data-Probes or Map-Requests.

- Indicate that outer TTL should only be copied to inner TTL when it is less than inner TTL.
- Indicate a source-EID for RLOC-probes are encoded with an AFI value of 0.
- Indicate that SMRs can have a global or per SMR destination rate-limiter.
- Add clarifications to the SMR procedures.
- Add definitions for "client-side" and 'server-side" terms used in this specification.
- Clear up language in section 6.4, last paragraph.
- Change ACT of value 0 to "no-action". This is so we can RLOC-probe a PETR and have it return a Map-Reply with a locator-set of size 0. The way it is spec'ed the map-cache entry has action "dropped". Drop-action is set to 3.
- Add statement about normalizing locator weights.
- Clarify R-bit definition in the Map-Reply locator record.
- Add section on EID Reachability within a LISP site.
- Clarify another disadvantage of using anycast locators.
- Reworded Abstract.
- Change section 2.0 Introduction to remove obsolete information such as the LISP variant definitions.
- Change section 5 title from "Tunneling Details" to "LISP Encapsulation Details".
- Changes to section 5 to include results of network deployment experience with MTU. Recommend that implementations use either the stateful or stateless handling.
- Make clarification wordsmithing to Section 7 and 8.
- Identify that if there is one locator in the locator-set of a map-cache entry, that an SMR from that locator should be responded to by sending the the SMR-invoked Map-Request to the database mapping system rather than to the RLOC itself (which may be unreachable).

- When describing Unicast and Multicast Weights indicate the the values are relative weights rather than percentages. So it doesn't imply the sum of all locator weights in the locator-set need to be 100.

- Do some wordsmithing on copying TTL and TOS fields.

- Numerous wordsmithing changes from Dave Meyer. He fine toothed combed the spec.

- Removed Section 14 "Prototype Plans and Status". We felt this type of section is no longer appropriate for a protocol specification.

- Add clarification text for the IRC description per Damien's commentary.

- Remove text on copying nonce from SMR to SMR-invoked Map- Request per Vina's comment about a possible DoS vector.

- Clarify (S/2 + H) in the stateless MTU section.

- Add text to reflect Damien's comment about the description of the "ITR-RLOC Address" field in the Map-Request. that the list of RLOC addresses are local addresses of the Map-Requester.

B.18. Changes to draft-ietf-lisp-07.txt

- Posted April 2010.

- Added I-bit to data header so LSB field can also be used as an Instance ID field. When this occurs, the LSB field is reduced to 8-bits (from 32-bits).

- Added V-bit to the data header so the 24-bit nonce field can also be used for source and destination version numbers.

- Added Map-Version 12-bit value to the EID-record to be used in all of Map-Request, Map-Reply, and Map-Register messages.

- Added multiple ITR-RLOC fields to the Map-Request packet so an ETR can decide what address to select for the destination of a Map-Reply.

- Added L-bit (Local RLOC bit) and p-bit (Probe-Reply RLOC bit) to the Locator-Set record of an EID-record for a Map-Reply message. The L-bit indicates which RLOCs in the locator-set are local to the sender of the message. The P-bit indicates which RLOC is the

source of a RLOC-probe Reply (Map-Reply) message.

- Add reference to the LISP Canonical Address Format [LCAF] draft.
- Made editorial and clarification changes based on comments from Dhirendra Trivedi.
- Added wordsmithing comments from Joel Halpern on DF=1 setting.
- Add John Zwiebel clarification to Echo Nonce Algorithm section 6.3.1.
- Add John Zwiebel comment about expanding on proxy-map-reply bit for Map-Register messages.
- Add NAT section per Ron Bonica comments.
- Fix IDnits issues per Ron Bonica.
- Added section on Virtualization and Segmentation to explain the use if the Instance ID field in the data header.
- There are too many P-bits, keep their scope to the packet format description and refer to them by name every where else in the spec.
- Scanned all occurrences of "should", "should not", "must" and "must not" and uppercased them.
- John Zwiebel offered text for section 4.1 to modernize the example. Thanks Z!
- Make it more clear in the definition of "EID-to-RLOC Database" that all ETRs need to have the same database mapping. This reflects a comment from John Scudder.
- Add a definition "Route-returnability" to the Definition of Terms section.
- In section 9.2, add text to describe what the signature of traceroute packets can look like.
- Removed references to Data Probe for introductory example. Data-probes are still part of the LISP design but not encouraged.
- Added the definition for "LISP site" to the Definition of Terms" section.

B.19. Changes to draft-ietf-lisp-06.txt

Editorial based changes:

- Posted December 2009.
- Fix typo for flags in LISP data header. Changed from "4" to "5".
- Add text to indicate that Map-Register messages must contain a computed UDP checksum.
- Add definitions for PITR and PETR.
- Indicate an AFI value of 0 is an unspecified address.
- Indicate that the TTL field of a Map-Register is not used and set to 0 by the sender. This change makes this spec consistent with [LISP-MS].
- Change "... yield a packet size of L octets" to "... yield a packet size greater than L octets".
- Clarify section 6.1.5 on what addresses and ports are used in Map-Reply messages.
- Clarify that LSBs that go beyond the number of locators do not to be SMRed when the locator addresses are greater lexicographically than the locator in the existing locator-set.
- Add Gregg, Srini, and Amit to acknowledgment section.
- Clarify in the definition of a LISP header what is following the UDP header.
- Clarify "verifying Map-Request" text in section 6.1.3.
- Add Xu Xiaohu to the acknowledgment section for introducing the problem of overlapping EID-prefixes among multiple sites in an RRG email message.

Design based changes:

- Use stronger language to have the outer IPv4 header set DF=1 so we can avoid fragment reassembly in an ETR or PETR. This will also make IPv4 and IPv6 encapsulation have consistent behavior.
- Map-Requests should not be sent in ECM with the Probe bit is set. These type of Map-Requests are used as RLOC-probes and are sent

directly to locator addresses in the underlying network.

- Add text in section 6.1.5 about returning all EID-prefixes in a Map-Reply sent by an ETR when there are overlapping EID-prefixes configure.

- Add text in a new subsection of section 6.1.5 about dealing with Map-Replies with coarse EID-prefixes.

B.20. Changes to draft-ietf-lisp-05.txt

- Posted September 2009.

- Added this Document Change Log appendix.

- Added section indicating that encapsulated Map-Requests must use destination UDP port 4342.

- Don't use AH in Map-Registers. Put key-id, auth-length, and auth-data in Map-Register payload.

- Added Jari to acknowledgment section.

- State the source-EID is set to 0 when using Map-Requests to refresh or RLOC-probe.

- Make more clear what source-RLOC should be for a Map-Request.

- The LISP-CONS authors thought that the Type definitions for CONS should be removed from this specification.

- Removed nonce from Map-Register message, it wasn't used so no need for it.

- Clarify what to do for unspecified Action bits for negative Map-Replies. Since No Action is a drop, make value 0 Drop.

B.21. Changes to draft-ietf-lisp-04.txt

- Posted September 2009.

- How do deal with record count greater than 1 for a Map-Request. Damien and Joel comment. Joel suggests: 1) Specify that senders compliant with the current document will always set the count to 1, and note that the count is included for future extensibility. 2) Specify what a receiver compliant with the draft should do if it receives a request with a count greater than 1. Presumably, it should send some error back?

- Add Fred Templin in acknowledgment section.

- Add Margaret and Sam to the acknowledgment section for their great comments.

- Say more about LAGs in the UDP section per Sam Hartman's comment.

- Sam wants to use MAY instead of SHOULD for ignoring checksums on ETR. From the mailing list: "You'd need to word it as an ITR MAY send a zero checksum, an ETR MUST accept a 0 checksum and MAY ignore the checksum completely. And of course we'd need to confirm that can actually be implemented. In particular, hardware that verifies UDP checksums on receive needs to be checked to make sure it permits 0 checksums."

- Margaret wants a reference to http://www.ietf.org/id/draft-eubanks-chimento-6man-00.txt.

- Fix description in Map-Request section. Where we describe Map-Reply Record, change "R-bit" to "M-bit".

- Add the mobility bit to Map-Replies. So PITRs don't probe so often for MNs but often enough to get mapping updates.

- Indicate SHA1 can be used as well for Map-Registers.

- More Fred comments on MTU handling.

- Isidor comment about spec'ing better periodic Map-Registers. Will be fixed in draft-ietf-lisp-ms-02.txt.

- Margaret's comment on gleaning: "The current specification does not make it clear how long gleaned map entries should be retained in the cache, nor does it make it clear how/ when they will be validated. The LISP spec should, at the very least, include a (short) default lifetime for gleaned entries, require that they be validated within a short period of time, and state that a new gleaned entry should never overwrite an entry that was obtained from the mapping system. The security implications of storing "gleaned" entries should also be explored in detail."

- Add section on RLOC-probing per working group feedback.

- Change "loc-reach-bits" to "loc-status-bits" per comment from Noel.

- Remove SMR-bit from data-plane. Dino prefers to have it in the control plane only.

- Change LISP header to allow a "Research Bit" so the Nonce and LSB fields can be turned off and used for another future purpose. For Luigi et al versioning convergence.

- Add a N-bit to the data header suggested by Noel. Then the nonce field could be used when N is not 1.

- Clarify that when E-bit is 0, the nonce field can be an echoed nonce or a random nonce. Comment from Jesper.

- Indicate when doing data-gleaning that a verifying Map-Request is sent to the source-EID of the gleaned data packet so we can avoid map-cache corruption by a 3rd party. Comment from Pedro.

- Indicate that a verifying Map-Request, for accepting mapping data, should be sent over the ALT (or to the EID).

- Reference IPsec RFC 4302. Comment from Sam and Brian Weis.

- Put E-bit in Map-Reply to tell ITRs that the ETR supports echo-noncing. Comment by Pedro and Dino.

- Jesper made a comment to loosen the language about requiring the copy of inner TTL to outer TTL since the text to get mixed-AF traceroute to work would violate the "MUST" clause. Changed from MUST to SHOULD in section 5.3.

B.22. Changes to draft-ietf-lisp-03.txt

- Posted July 2009.

- Removed loc-reach-bits longword from control packets per Damien comment.

- Clarifications in MTU text from Roque.

- Added text to indicate that the locator-set be sorted by locator address from Isidor.

- Clarification text from John Zwiebel in Echo-Nonce section.

B.23. Changes to draft-ietf-lisp-02.txt

- Posted July 2009.

- Encapsulation packet format change to add E-bit and make loc-reach-bits 32-bits in length.

- Added Echo-Nonce Algorithm section.
- Clarification how ECN bits are copied.
- Moved S-bit in Map-Request.
- Added P-bit in Map-Request and Map-Reply messages to anticipate RLOC-Probe Algorithm.
- Added to Mobility section to reference [LISP-MN].

B.24. Changes to draft-ietf-lisp-01.txt

- Posted 2 days after draft-ietf-lisp-00.txt in May 2009.
- Defined LEID to be a "LISP EID".
- Indicate encapsulation use IPv4 DF=0.
- Added negative Map-Reply messages with drop, native-forward, and send-map-request actions.
- Added Proxy-Map-Reply bit to Map-Register.

B.25. Changes to draft-ietf-lisp-00.txt

- Posted May 2009.
- Rename of draft-farinacci-lisp-12.txt.
- Acknowledgment to RRG.

Authors' Addresses

Dino Farinacci
cisco Systems
Tasman Drive
San Jose, CA 95134
USA

Email: dino@cisco.com

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

Email: vaf@cisco.com

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

Email: dmm@cisco.com

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

Email: darlewis@cisco.com

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: September 2, 2012

L. Iannone
Telekom Innovation Laboratories
D. Saucez
INRIA Sophia Antipolis
O. Bonaventure
Universite catholique de Louvain

March 1, 2012

LISP Map-Versioning
draft-ietf-lisp-map-versioning-09.txt

Abstract

This document describes the LISP (Locator/ID Separation Protocol) Map-Versioning mechanism, which provides in-packet information about Endpoint-ID to Routing Locator (EID-to-RLOC) mappings used to encapsulate LISP data packets. The proposed approach is based on associating a version number to EID-to-RLOC mappings and transport such a version number in the LISP specific header of LISP-encapsulated packets. LISP Map-Versioning is particularly useful to inform communicating Ingress Tunnel Routers (ITRs) and Egress Tunnel Routers (ETRs) about modifications of the mappings used to encapsulate packets. The mechanism is transparent to implementations not supporting this feature, since in the LISP-specific header and in the Map Records, bits used for Map-Versioning can be safely ignored by ITRs and ETRs that do not support the mechanism.

Table of Contents

1. Introduction

This document describes the Map-Versioning mechanism used to provide information on changes in the EID-to-RLOC (Endsystem ID to Routing LOCator) mappings used in the LISP (Locator/Id Separation Protocol [I-D.ietf-lisp]) context to perform packet encapsulation. The mechanism is totally transparent to xTRs (Ingress and Egress Tunnel Routers) not supporting such functionality. It is not meant to replace any existing LISP mechanism, but rather to extend them providing new functionalities. If for any unforseen reason a normative conflict between the present document and the LISP main specifications is found, the latter ([I-D.ietf-lisp]) has precedence on the present document.

The basic mechanism is to associate a Map-Version number to each LISP EID-to-RLOC mapping and transport such a version number in the LISP-specific header. When a mapping changes, a new version number is assigned to the updated mapping. A change in an EID-to-RLOC mapping can be a change in the RLOCs set, by adding or removing one or more RLOCs, but it can also be a change in the priority or weight of one or more RLOCs.

When Map-Versioning is used, LISP-encapsulated data packets contain the version number of the two mappings used to select the RLOCs in the outer header (i.e., both source and destination). These version numbers are encoded in the 24 low-order bits of the first longword of the LISP header and indicated by a specific bit in the flags (first 8 high-order bits of the first longword of the LISP header). Note that not all packets need to carry version numbers.

When an ITR (Ingress Tunnel Router) encapsulates a data packet, with a LISP header containing the Map-Version numbers, it puts in the LISP-specific header two version numbers:

1. The version number assigned to the mapping (contained in the EID-to-RLOC Database) used to select the source RLOC.

2. The version number assigned to the mapping (contained in the EID-to-RLOC Cache) used to select the destination RLOC.

This operation is two-fold. On the one hand, it enables the ETR (Egress Tunnel Router) receiving the packet to know if the ITR has the latest version number that any ETR at the destination EID site has provided to the ITR in a Map-Reply. If it is not the case the ETR can send to the ITR a Map-Request containing the updated mapping or soliciting a Map-Request from the ITR (both cases are already defined in [I-D.ietf-lisp]). In this way the ITR can update its EID-to-RLOC Cache. On the other hand, it enables an ETR receiving such a

packet to know if it has in its EID-to-RLOC Cache the latest mapping for the source EID (in case of bidirectional traffic). If it is not the case a Map-Request can be sent.

Issues and concerns about the deployment of LISP for Internet traffic are discussed in [I-D.ietf-lisp]. Section 12 provides additional issues and concerns raised by this document. In particular, Section 12.1 provides details about the ETRs' synchronization issue in the context of Map-Versioning.

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. Definitions of Terms

The present document uses terms already defined in main LISP specification [I-D.ietf-lisp]. Hereafter are defined only the terms that are specific to the Map-Versioning mechanism. Throughout the whole document Big Endian bit ordering is used.

Map-Version number: An unsigned 12-bits assigned to an EID-to-RLOC mapping, not including the value 0 (0x000).

Null Map-Version: The 12-bits null value of 0 (0x000) is not used as Map-Version number. It is used to signal that no Map-Version number is assigned to the EID-to-RLOC mapping.

Source Map-Version number: Map-Version number of the EID-to-RLOC mapping used to select the source address (RLOC) of the outer IP header of LISP-encapsulated packets.

Destination Map-Version number: Map-Version number of the EID-to-RLOC mapping used to select the destination address (RLOC) of the outer IP header of LISP-encapsulated packets.

4. EID-to-RLOC Map-Version number

The EID-to-RLOC Map-Version number consists in an unsigned 12-bits integer. The version number is assigned on a per-mapping basis, meaning that different mappings have a different version number, which is also updated independently. An update in the version number (i.e., a newer version) consists in incrementing by one the older

version number. Appendix A contains a rough estimation of the wrap-around time for the Map-Version number.

The space of version numbers has a circular order where half of the version numbers is greater(i.e., newer) than the current Map-Version number and the other half is smaller (i.e., older) than current Map-Version number. In a more formal way, assuming we have two version numbers V1 and V2 and that the numbers are expressed on N bits, the following steps MUST be performed (in the same order as hereafter) to strictly define their order:

1. V1 = V2 : The map-version number are the same.

2. V2 > V1 : if and only if

   V2 > V1 AND (V2 - V1) <= 2**(N-1)

   OR

   V1 > V2 AND (V1 - V2) > 2**(N-1)

3. V1 > V2 : otherwise.

Using 12 bits, as defined in this document, and assuming a Map-Version value of 69, Map-Version numbers in the range [70; 69 + 2048] are greater than 69, while Map-Version numbers in the range [69 + 2049; (69 + 4096) mod 4096] are smaller than 69.

Map-version number are assigned to mappings by configuration. The initial Map-Version number of a new EID-to-RLOC mapping SHOULD be assigned randomly, but it MUST NOT be set to the Null Map-Version value (0x000), because it has a special meaning (see Section 4.1).

Upon reboot, an ETR will use mappings configured in its EID-to-RLOC Database. If those mappings have a Map-Version number, it will be used according to the mechnisms described in this document. ETRs MUST NOT automatically generate and assign Map-Version numbers to mappings in the EID-to-RLOC Database.

4.1. The Null Map-Version

The value 0x000 (zero) is not a valid Map-Version number indicating the version of the EID-to-RLOC mapping. Such a value is used for special purposes and is named the Null Map-Version number.

The Null Map-Version MAY appear in the LISP specific header as either Source Map-Version number (cf. Section 5.2) or Destination Map-Version number (cf. Section 5.1). When the Source Map-Version number

is set to the Null Map-version value it means that no map version information is conveyed for the source site. This means that if a mapping exists for the source EID in the EID-to-RLOC Cache, then the ETR MUST NOT compare the received Null Map-Version with the content of the EID-to-RLOC Cache. When the Destination Map-version number is set to the Null Map-version value it means that no map version information is conveyed for the destination site. This means that the ETR MUST NOT compare the value with the Map-Version number of the mapping for the destination EID present in the EID-to-RLOC Database.

The other use of the Null Map-Version number is in the Map Records, which are part of the Map-Request, Map-Reply and Map-Register messages (defined in [I-D.ietf-lisp]). Map Records that have a Null Map-Version number indicate that there is no Map-Version number associated with the mapping. This means that LISP encapsulated packets, destined to the EID-Prefix the Map Record refers to, MUST either not contain any Map-Version numbers (V bit set to 0), or if it contains Map-Version numbers (V bit set to 1) then the destination Map-Version number MUST be set to the Null Map-Version number. Any value different from zero means that Map-Versioning is supported and MAY be used.

The fact that the 0 value has a special meaning for the Map-Version number implies that, when updating a Map-Version number because of a change in the mapping, if the next value is 0 then Map-Version number MUST be incremented by 2 (i.e., set to 1, which is the next valid value).

5. Dealing with Map-Version numbers

The main idea of using Map-Version numbers is that whenever there is a change in the mapping (e.g., adding/removing RLOCs, a change in the weights due to TE policies, or a change in the priorities) or a LISP site realizes that one or more of its own RLOCs are not reachable anymore from a local perspective (e.g., through IGP, or policy changes) the LISP site updates the mapping also assigning a new Map-Version number.

To each mapping, a version number is associated and changes each time the mapping is changed. Note that map-versioning does not introduce new problems concerning the coordination of different ETRs of a domain. Indeed, ETRs belonging to the same LISP site must return for a specific EID-prefix the same mapping, including the same Map-Version number. In principle this is orthogonal to whether or not map-versioning is used. The synchronization problem and its implication on the traffic is out of the scope of this document (see Section 12).

In order to announce in a data-driven fashion that the mapping has been updated, Map-Version numbers used to create the outer IP header of the LISP-encapsulated packet are embedded in the LISP-specific header. This means that the header needs to contain two Map-Version numbers:

- The Source Map-Version number of the EID-to-RLOC mapping in the EID-to-RLOC Database used to select the source RLOC.

- The Destination Map-Version number of the EID-to-RLOC mapping in the EID-to-RLOC Cache used to select the destination RLOC.

By embedding both Source Map-Version number and Destination Map-Version number an ETR receiving a LISP packet with Map-Version numbers, can perform the following checks:

1. The ITR that has sent the packet has an up-to-date mapping in its EID-to-RLOC Cache for the destination EID and is performing encapsulation correctly.

2. In case of bidirectional traffic, the mapping in the local ETR EID-to-RLOC Cache for the source EID is up-to-date.

If one or both of the above conditions do not hold, the ETR can send a Map-Request either to make the ITR aware that a new mapping is available (see Section 5.1) or to update the mapping in the local EID-to-RLOC Cache (see Section 5.2).

5.1. Handling Destination Map-Version number

When an ETR receives a packet, the Destination Map-Version number relates to the mapping for the destination EID for which the ETR is a RLOC. This mapping is part of the ETR EID-to-RLOC Database. Since the ETR is authoritative for the mapping, it has the correct and up-to-date Destination Map-Version number. A check on this version number can be done, where the following cases can arise:

1. The packets arrive with the same Destination Map-Version number stored in the EID-to-RLOC Database. This is the regular case. The ITR sending the packet has in its EID-to-RLOC Cache an up-to-date mapping. No further actions are needed.

2. The packet arrives with a Destination Map-Version number greater (i.e., newer) than the one stored in the EID-to-RLOC Database. Since the ETR is authoritative on the mapping, meaning that the Map-Version number of its mapping is the correct one, this implies that someone is not behaving correctly with respect to the specifications. In this case the packet carries a version

number that is not valid, otherwise the ETR would have the same, and SHOULD be silently dropped.

3. The packets arrive with a Destination Map-Version number smaller (i.e., older) than the one stored in the EID-to-RLOC Database. This means that the ITR sending the packet has an old mapping in its EID-to-RLOC Cache containing stale information. The ETR MAY choose to normally process the encapsulated datagram according to [I-D.ietf-lisp], however, the ITR sending the packet has to be informed that a newer mapping is available. This is done with a Map-Request message sent back to the ITR. The Map-Request will either trigger a Map-Request back using the Solicit-Map-Request (SMR) bit or it will piggyback the newer mapping. These are not new mechanisms; how to SMR or piggyback mappings in Map-Request messages is already described in [I-D.ietf-lisp], while their security is discussed in [I-D.ietf-lisp-threats]. These Map-Request messages should be rate limited (rate limitation policies are also described in [I-D.ietf-lisp]). The feature introduced by Map-Version numbers is the possibility of blocking traffic not using the latest mapping. Indeed, after a certain number of retries, if the Destination Map-Version number in the packets is not updated, the ETR MAY drop packets with a stale Map-Version number while strongly reducing the rate of Map-Request messages. This because either the ITR is refusing to use the mapping for which the ETR is authoritative or (worse) it might be some form of attack. Another case might be that the control-plane is experiencing transient failures so the Map-Requests cannot reach that ITR. By keeping sending Map-Requests at very low rate it is possible to recover from this situation.

The rule in the third case MAY be more restrictive. If the mapping has been the same for a period of time as long as the TTL (defined in [I-D.ietf-lisp]) of the previous version of the mapping, all packets arriving with an old Map-Version SHOULD be silently dropped right away without issuing any Map-Request. The reason that allows such action is the fact that if the new mapping with the updated version number has been unchanged for at least the same time as the TTL of the older mapping, all the entries in the EID-to-RLOC Caches of ITRs must have expired. Hence, all ITRs sending traffic should have refreshed the mapping according to [I-D.ietf-lisp]. If packets with old Map-Version number are still received, then either someone has not respected the TTL, or it is a form of spoof/attack. In both cases this is not valid behavior with respect to the specifications and the packet SHOULD be silently dropped.

LISP-encapsulated packets with the V-bit set, when the original mapping in the EID-to-RLOC Database has version number set to the Null Map-Version value, MAY be silently dropped. As explained in

Section 4.1, if an EID-to-RLOC mapping has a Null Map-Version, it means that ITRs, using the mapping for encapsulation, MUST NOT use Map-Version number in the LISP-specific header.

For LISP-encapsulated packets with the V-bit set, when the original mapping in the EID-to-RLOC Database has version number set to a value different from the Null Map-Version value, a Destination Map-Version number equal to the Null Map-Version value means that the Destination Map-Version number MUST be ignored.

5.2. Handling Source Map-Version number

When an ETR receives a packet, the Source Map-Version number relates to the mapping for the source EID for which the ITR that sent the packet is authoritative. If the ETR has an entry in its EID-to-RLOC Cache for the source EID, then a check can be performed and the following cases can arise:

1. The packet arrives with the same Source Map-Version number stored in the EID-to-RLOC Cache. This is the correct regular case. The ITR has in its EID-to-RLOC Cache an up-to-date copy of the mapping. No further actions are needed.

2. The packet arrives with a Source Map-Version number greater (i.e., newer) than the one stored in the local EID-to-RLOC Cache. This means that ETR has in its EID-to-RLOC Cache a mapping that is stale and needs to be updated. A Map-Request SHOULD be sent to get the new mapping for the source EID. This is a normal Map-Request message sent through the mapping system and MUST respect the specifications in [I-D.ietf-lisp], including rate limitation policies.

3. The packet arrives with a Source Map-Version number smaller (i.e., older) than the one stored in the local EID-to-RLOC Cache. Such a case is not valid with respect to the specifications. Indeed, if the mapping is already present in the EID-to-RLOC Cache, this means that an explicit Map-Request has been sent and a Map-Reply has been received from an authoritative source. Assuming that the mapping system is not corrupted anyhow, the Map-Version in the EID-to-RLOC Cache is the correct one, while the one carried by the packet is stale. In this situation the packet MAY be silently dropped.

If the ETR does not have an entry in the EID-to-RLOC Cache for the source EID (e.g., in case of unidirectional traffic) then the Source Map-Version number can be safely ignored.

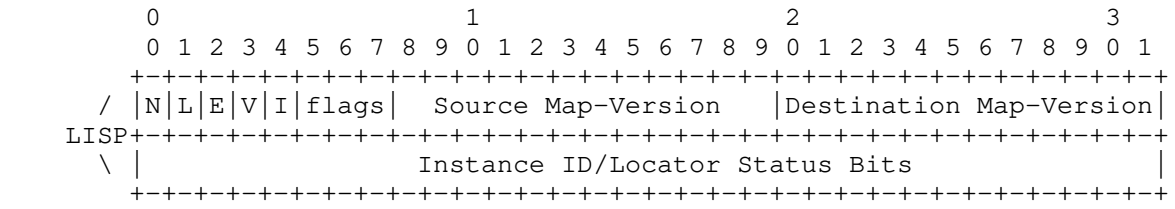
For LISP-encapsulated packets with the V-bit set, if the Source Map-

Version number is the Null Map-Version value, it means that the Source Map-Version number MUST be ignored.

6. LISP header and Map-Version numbers

In order for the versioning approach to work, the LISP specific header has to carry both Source Map-Version number and Destination Map-Version number. This is done by setting the V-bit in the LISP specific header as defined in [I-D.ietf-lisp] Section 5.3. When the V-bit is set the low-order 24-bits of the first longword are used to transport both source and destination Map-Version numbers. In particular the first 12 bits are used for Source Map-Version number and the second 12 bits for the Destination Map-Version number.

Hereafter is the example of LISP header carrying version numbers in the case of IPv4-in-IPv4 encapsulation. The same setting can be used for any other case (IPv4-in-IPv6, IPv6-in-IPv4, and IPv6-in-IPv6).



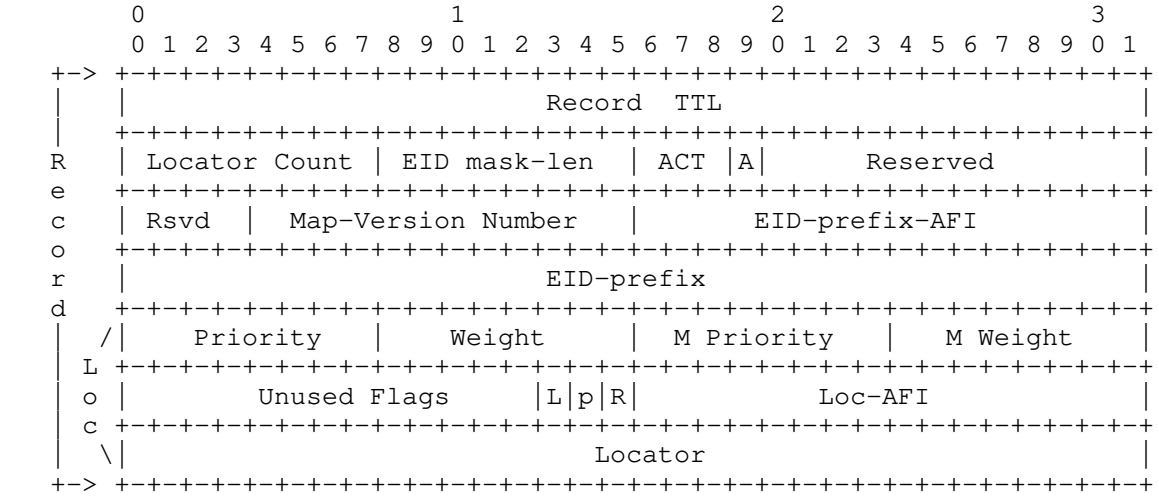
Source Map-Version number (12 bits): Map-Version of the mapping used by the ITR to select the RLOC present in the "Source Routing Locator" field. How to set on transmission and handle on reception this value is described in Section 5.2.

Destination Map-Version number (12 bits): Map-Version of the mapping used by the ITR to select the RLOC present in the "Destination Routing Locator" field. How to set on transmission and handle on reception this value is described in Section 5.1.

The present document just specifies how to use the low-order 24-bits of the first longword of the LISP-specific header when the V-bit is set to 1. All other cases, including the bit fields of the rest of the LISP-specific header and the whole LISP packet format are specified in [I-D.ietf-lisp]. Not all of the LISP encapsulated packets need to carry version numbers. When Map-Version numbers are carried the V-bit MUST be set to 1. All legal combinations of the flags, when the V-bit is set to 1, are described in [I-D.ietf-lisp].

7. Map Record and Map-Version

To accommodate the proposed mechanism, the Map Records that are transported on Map-Request/Map-Reply/Map-Register messages need to carry the Map-Version number as well. For this purpose the 12-bits before the EID-AFI field in the Record that describe a mapping is used. This is defined in Section 6.1.4 of [I-D.ietf-lisp] and reported here as example.



Map-Version Number: Map-Version of the mapping contained in the Record. As explained in Section 4.1 this field can be zero (0), meaning that no Map-Version is associated to the mapping, hence packets that are LISP-encapsulated using this mapping MUST NOT contain Map-Version numbers in the LISP specific header and the V-bit MUST be set to 0.

This packet format works perfectly with xTRs that do not support Map-Versioning, since they can simply ignore those bits.

8. Benefits and case studies for Map-Versioning

In the following sections we provide more discussion on various aspects and use of the Map-Versioning. Security observations are instead grouped in Section 10.

8.1. Map-Versioning and unidirectional traffic

When using Map-Versioning the LISP specific header carries two Map-Version numbers, for both source and destination mappings. This can raise the question on what will happen in the case of unidirectional flows, like for instance in the case presented in Figure 1, since LISP specification do not mandate for ETR to have a mapping for the source EID.

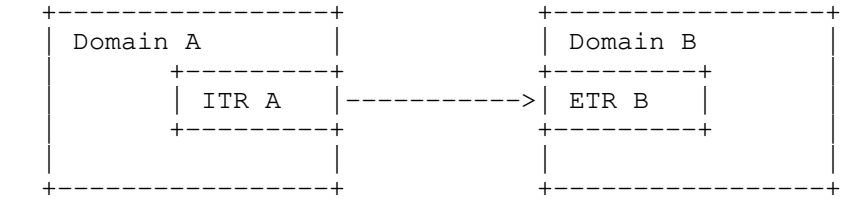


Figure 1

For what concerns the ITR, it is able to put both source and destination version number in the LISP header since the Source Map-Version number is in ITR's database, while the Destination Map-Version number is in ITR's cache.

For what concerns the ETR, it simply checks only the Destination Map-Version number in the same way as described in Section 5, ignoring the Source Map-Version number.

8.2. Map-Versioning and interworking

Map-Versioning is compatible with the LISP interworking between LISP and non-LISP sites as defined in [I-D.ietf-lisp-interworking]. LISP interworking defines three techniques to make LISP sites and non-LISP sites, namely Proxy-ITR, LISP-NAT, and Proxy-ETR. Hereafter it is described how Map-Versioning relates to these three mechanisms.

8.2.1. Map-Versioning and Proxy-ITRs

The purpose of the Proxy-ITR (PITR) is to encapsulate traffic originating in a non-LISP site in order to deliver the packet to one of the ETRs of the LISP site (cf. Figure 2). This case is very similar to the unidirectional traffic case described in Section 8.1, hence similar rules apply.

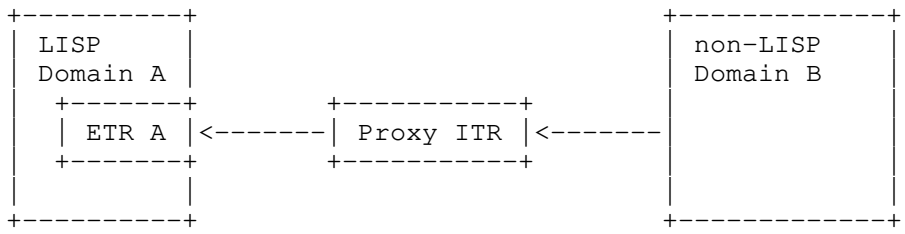


Figure 2

The main difference is that a Proxy-ITR does not have any mapping, since it just encapsulate packets arriving from non-LISP site, thus cannot provide a Source Map-Version. In this case, the proxy-ITR will just put the Null Map-Version value as Source Map-Version number, while the receiving ETR will ignore the field.

With this setup the LISP Domain A is able to check whether or not the PITR is using the latest mapping. If this is not the case the mapping for LISP Domain A on the PITR can be updated using one of the mechanisms defined in [I-D.ietf-lisp] and [I-D.ietf-lisp-interworking].

8.2.2. Map-Versioning and LISP-NAT

The LISP-NAT mechanism is based on address translation from non-routable EIDs to routable EIDs and does not involve any form of encapsulation. As such Map-Versioning does not apply in this case.

8.2.3. Map-Versioning and Proxy-ETRs

The purpose of the Proxy-ETR (PETR) is to decapsulate traffic originating in a LISP site in order to deliver the packet to the non-LISP site (cf. Figure 3). One of the main reasons of deploy PETRs is to bypass uRPF (Unicast Reverse Path Forwarding) checks on the provider edge.

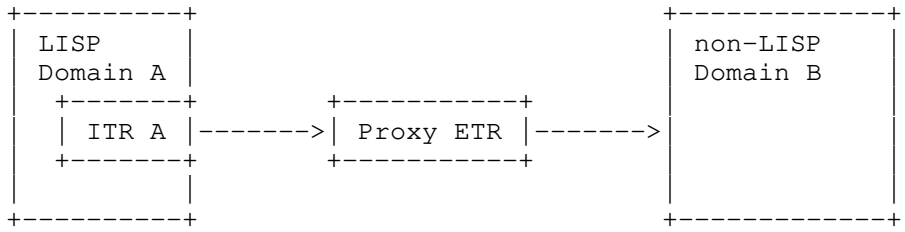


Figure 3

A Proxy-ETR does not have any mapping, since it just decapsulates packets arriving from LISP site. In this case, the ITR will just put the Null Map-Version value as Destination Map-Version number, while the receiving Proxy-ETR will ignore the field.

With this setup the Proxy-ETR is able to check whether or not the mapping has changed. If this is the case the mapping for LISP Domain A on the PETR can be updated using one of the mechanisms defined in [I-D.ietf-lisp] and [I-D.ietf-lisp-interworking].

8.3. RLOC shutdown/withdraw

Map-Versioning can be even used to perform a graceful shutdown or withdraw of a specific RLOC. This is achieved by simply issuing a new mapping, with an updated Map-Version number, where the specific RLOC to be shut down is withdrawn or announced as unreachable (R bit in the Map Record, see [I-D.ietf-lisp]), but without actually turning it off.

Once no more traffic is received by the RLOC, it can be shut down gracefully, because at least all sites actively using the mapping have updated it.

It should be pointed out that for frequent up/down changes such a mechanism should not be used since this can generate excessive load on the Mapping System.

8.4. Map-Version for lightweight LISP implementation

The use of Map-Versioning can help in developing a lightweight implementation of LISP. This comes with the price of not supporting Loc-Status-Bit, which are useful in some contexts.

In the current LISP specifications the set of RLOCs must always be maintained ordered and consistent with the content of the Loc Status Bits (see section 6.5 of [I-D.ietf-lisp]). With Map-Versioning such type of mechanisms can be avoided. When a new RLOC is added to a mapping, it is not necessary to "append" new locators to the existing ones as explained in Section 6.5 of [I-D.ietf-lisp]. A new mapping with a new Map-Version number will be issued, and since the old locators are still valid the transition will be with no disruptions. The same applies for the case a RLOC is withdrawn. There is no need to maintain holes in the list of locators, as is the case when using Locator Status Bits, for sites that are not using the RLOC that has been withdrawn the transition will be with no disruptions.

All of these operations, as already stated, do not need to maintain any consistency among Locator Status Bits, and the way RLOC are

stored in the EID-to-RLOC Cache.

Further, Map-Version can be used to substitute the "clock sweep" operation described in Section 6.5.1 of [I-D.ietf-lisp]. Indeed, every LISP site communicating to a specific LISP site that has updated the mapping will be informed of the available new mapping in a data-driven manner.

Note that what is proposed in the present section is just an example and MUST NOT be considered as specifications for a lightweight LISP implementation. In case the IETF decides to undertake such a work, it will be documented elsewhere.

9. Incremental deployment and implementation status

Map-Versioning can be incrementally deployed without any negative impact on existing LISP elements (e.g., xTRs, Map-Servers, Proxy-ITRs, etc). Any LISP element that does not support Map-Versioning can safely ignore them. Further, there is no need of any specific mechanism to discover if an xTR supports or not Map-Versioning. This information is already included in the Map Record.

Map-Versioning is currently implemented in OpenLISP [I-D.iannone-openlisp-implementation].

Note that the reference document for LISP implementation and interoperability tests remains [I-D.ietf-lisp].

10. Security Considerations

Map-Versioning does not introduce any security issue concerning both the data-plane and the control-plane. On the contrary, as described in the following, if Map-Versioning may be used also to update mappings in case of change in the reachability information (i.e., instead of the Locator Status Bits) it is possible to reduce the effects of some DoS or spoofing attacks that can happen in an untrusted environment.

Robustness of the Map-Versioning mechanism leverages on a trusted Mapping Distribution System. A thorough security analysis of LISP is documented in [I-D.ietf-lisp-threats].

10.1. Map-Versioning against traffic disruption

An attacker can try to disrupt ongoing communications by creating LISP encapsulated packets with wrong Locator Status Bits. If the xTR

blindly trusts the Locator Status Bits it will change the encapsulation accordingly, which can result in traffic disruption.

This does not happen in the case of Map-Versioning. As described in Section 5, upon a version number change the xTR first issues a Map-Request. The assumption is that the mapping distribution system is sufficiently secure that Map-Request and Map-Reply messages and their content can be trusted. Security issues concerning specific mapping distribution system are out of the scope of this document. In the case of Map-Versioning the attacker should "guess" a valid version number that triggers a Map-Request, as described in Section 5, otherwise the packet is simply dropped. Nevertheless, guessing a version number that generates a Map-Request is easy, hence it is important to follow the rate limitations policies described in [I-D.ietf-lisp] in order to avoid DoS attacks.

Note that a similar level of security can be obtained with Loc Status Bits, by simply making mandatory to verify any change through a Map-Request. However, in this case Locator Status Bits loose their meaning, because, it does not matter anymore which specific bits has changed, the xTR will query the mapping system and trust the content of the received Map-Reply. Furthermore there is no way to perform filtering as in the Map-Versioning in order to drop packets that do not carry a valid Map-Version number. In the case of Locator Status Bits, any random change can trigger a Map-Request (unless rate limitation is enabled which raise another type of attack discussed in Section 10.2).

10.2. Map-Versioning against reachability information DoS

Attackers can try to trigger a large amount of Map-Request by simply forging packets with random Map-Version or random Locator Status Bits. In both cases the Map-Requests are rate limited as described in [I-D.ietf-lisp]. However, differently from Locator Status Bit where there is no filtering possible, in the case of Map-Versioning is possible to filter not valid version numbers before triggering a Map-Request, thus helping in reducing the effects of DoS attacks. In other words the use of Map-Versioning enables a fine control on when to update a mapping or when to notify that a mapping has been updated.

It is clear, that Map-Versioning does not protect against DoS and DDoS attacks, where an xTR looses processing power doing checks on the LISP header of packets sent by attackers. This is independent from Map-Versioning and is the same for Loc Status Bits.

11. IANA Considerations

This document has no actions for IANA.

12. Open Issues and Considerations

There are a number of implications of the use of Map-Versioning that are not yet completely explored. Among these are:

- Performance of the convergence time when an EID-to-RLOC mapping changes, i.e., how much time is needed to update mappings in the EID-to-RLOC Cache of the ITRs currently sending traffic to ETRs for the EID whose mapping has been changed.

- Support to ETR synchronization. The implications that a temporary lack of synchronization may have on the traffic is yet to be fully explored. Details on how to keep synchronization are presented in Section 6.6 of [I-D.ietf-lisp]. Section 12.1 hereafter discusses the issue in further details with respect to the Map-Versioning mechanism.

The authors expect that experimentation will help assess the performance and the limitations of the Map-Versioning mechanism. Issues and concerns about the deployment of LISP for Internet traffic are discussed in [I-D.ietf-lisp].

12.1. Lack of Synchronization among ETRs

Even without Map-Versioning, LISP ([I-D.ietf-lisp]) requires ETRs to announce the same mapping for the same EID-Prefix to a requester. The implications that a temporary lack of synchronization may have on the traffic is yet to be fully explored.

Map-Versioning does not require additional synchronization mechanism compared to the normal functioning of LISP without Map-Versioning. Clearly all the ETRs have to reply with the same Map-Version number, otherwise there can be an inconsistency that creates additional control traffic, instabilities, traffic disruptions. It is the same without Map-Versioning, with ETRs that have to reply with the same mapping, otherwise the same problems can arise.

There are two ways Map-Versioning is helpful with respect to the synchronization problem. On the one hand, assigning version numbers to mappings helps in debugging, since quick checks on the consistency of the mappings on different ETRs can be done by looking at the Map-Version number. On the other hand, Map-Versioning can be used to control the traffic toward ETRs that announce the latest mapping.

As an example, let's consider the topology of Figure 4 where ITR A.1 of domain A is sending unidirectional traffic to the domain B, while A.2 of domain A exchanges bidirectional traffic with domain B. In particular, ITR A.2 sends traffic to ETR B and ETR A.2 receives traffic from ITR B.

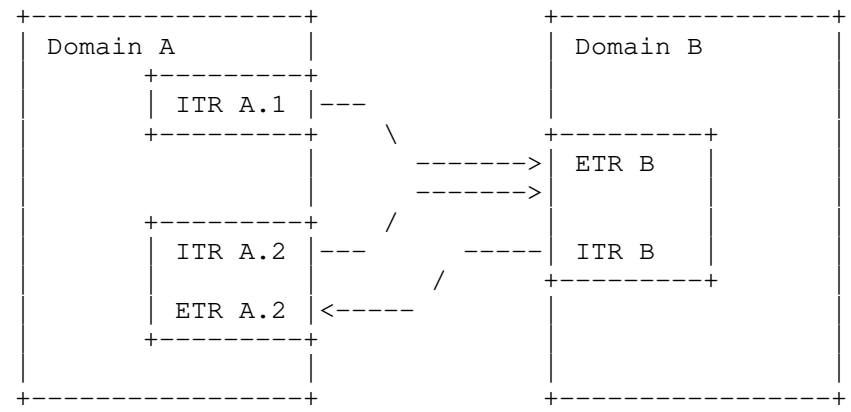


Figure 4

Obviously in the case of Map-Versioning both ITR A.1 and ITR A.2 of domain A must use the same value otherwise the ETR of domain B will start to send Map-Requests.

The same problem can, however, arise without Map-Versioning. For instance, if the two ITRs of domain A send different Locator Status Bits. In this case either the traffic is disrupted, if the ETR B trusts the Locator Status Bits, or if ETR B does not trust the Locator Status Bits it will start sending Map-Requests to confirm the each change in the reachability.

So far, LISP does not provide any specific synchronization mechanism, but assumes that synchronization is provided by configuring the different xTRs consistently (see Section 6.6 in [I-D.ietf-lisp]). The same applies for Map-Versioning. If in the future any synchronization mechanism is provided, Map-Versioning will take advantage of it automatically since it is included in the Record format, as described in Section 7.

13. Acknowledgements

The authors would like to thank Alia Atlas, Jesper Skriver, Pierre Francois, Noel Chiappa, Dino Farinacci for their comments and review.

This work has been partially supported by the INFSO-ICT-216372 TRILOGY Project (www.trilogy-project.org).

14. References

14.1. Normative References

[I-D.ietf-lisp]
Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol (LISP)", draft-ietf-lisp-22 (work in progress), February 2012.

[I-D.ietf-lisp-interworking]
Lewis, D., Meyer, D., Farinacci, D., and V. Fuller, "Interworking LISP with IPv4 and IPv6", draft-ietf-lisp-interworking-05 (work in progress), February 2012.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

14.2. Informative References

[I-D.iannone-openlisp-implementation]
Iannone, L., Saucez, D., and O. Bonaventure, "OpenLISP Implementation Report", draft-iannone-openlisp-implementation-01 (work in progress), July 2008.

[I-D.ietf-lisp-alt]
Fuller, V., Farinacci, D., Meyer, D., and D. Lewis, "LISP Alternative Topology (LISP+ALT)", draft-ietf-lisp-alt-10 (work in progress), December 2011.

[I-D.ietf-lisp-ms]
Fuller, V. and D. Farinacci, "LISP Map Server Interface", draft-ietf-lisp-ms-15 (work in progress), January 2012.

[I-D.ietf-lisp-threats]
Saucez, D., Iannone, L., and O. Bonaventure, "LISP Threats Analysis", draft-ietf-lisp-threats-00 (work in progress), July 2011.

Appendix A. Estimation of time before Map-Version wrap-around

The present section proposes an estimation of the wrap-around time

for the 12 bits size of the Map-Version number.

Using a granularity of seconds and assuming as worst-case that a new version is issued each second, it takes slightly more than 1 hour before the version wraps around. Note that the granularity of seconds is in line with the rate limitation policy for Map-Request messages, as proposed in the LISP main specifications ([I-D.ietf-lisp]).

Alternatively a granularity of minutes can also be used, as for the TTL of the Map-Reply ([I-D.ietf-lisp]). In this case the worst scenario is when a new version is issued every minute, leading to a much longer time before wrap-around. In particular, when using 12 bits, the wrap-around time is almost 3 days.

For general information, hereafter there is a table with a rough estimation of the time before wrap-around in the worst-case scenario, considering different sizes (bits length) of the Map-Version number and different time granularity.

Since even in the case of high mapping change rate (1 per second) the wrap around time using 12 bits is far larger then any reasonable Round-Trip-Time (RTT), there is no risk of race conditions.

Version Number Size (bits)	Time before wrap around	
	Granularity: Minutes (mapping changes every 1 minute)	Granularity: Seconds (mapping changes every 1 second)
32	8171 Years	136 Years
30	2042 Years	34 Years
24	31 Years	194 Days
16	45 Days	18 Hours
15	22 Days	9 Hours
14	11 Days	4 Hours
13	5.6 Days	2.2 Hours
12	2.8 Days	1.1 Hours

Figure 5: Estimation of time before wrap-around

Appendix B. Document Change Log

- Version 09 Posted March 2012.
  - Text in Section 5.1 made more explicit in the case of smaller (i.e., older) Destination Map-Version Number, as pointed out by Ralph E. Droms.
- Version 08 Posted Ferbruary 2012.
  - Clarifications added to Appendix A as requested by S. Bryant.
- Version 07 Posted January 2012.
  - Moved Subsection 8.1 in Section 12 as requested by R. Bonica.
  - Added explicit reference to the discussion about ETR synchronization at the end of the Introduction, as requested by R. Bonica.
  - Added cross-reference to Section 6.6 in [I-D.ietf-lisp] as requested by R. Bonica.
  - Moved [I-D.ietf-lisp-interworking] as normative reference as requested by R. Droms.
  - Added long version of all acronyms in the Introduction as requested by S. Bryant.
- Version 06 Posted October 2011.
  - Added disclaimer in the Introduction about general issues concerning LISP as requested by A. Farrel.
  - Fixed sentence about legacy systems in the abstract as requested by A. Farrel.
  - Added Section 12 as requested by A. Farrel.
- Version 05 Posted October 2011.
  - Added sentence in Section 3 on the use of Big Endian, as for comment of P. Resnick.
  - Extended the end of Section 4 in order to clarify that Map-Version numbers are assigned to mappings by configuration and not automatically generated by ETRs, as for comments of R. Sparks

* Changed formal definition of Map-Version order (greater vs. smaller) in Section 4 as for comments from R. Housley and R. Sparks.

* Added disclaimer in Section 1 stating that in case of unforseen conflict with the main spec the base document has precedence on the present one, as for comment from Sthephen Farrell.

- Version 04 Posted September 2011.

  * Added clarifications in Section 1, Section 4, Section 5.2, and Section 5.1 to address Stephen Farrell's comments.

  * Used the term LISP Site instead of ISP in Section 5 as suggested by Stephen Farrell.

  * Deleted "(usually contains the nonce)" from Section 6 because confusing, as suggested by Stephen Farrell.

  * Fixed several typos pointed out by Stephen Farrell.

- Version 03 Posted September 2011.

  * Added reference in Section 7 toward the main lisp documents specifying the section, as requested by Jari Arkko.

  * Fixed all typos and editorial issues pointed out by Jari Arkko.

  * Added clarification in Section 8.3 as requested by Jari Arkko.

  * Extentend all acronyms in the abstract as requested by Jari Arkko.

  * Clarified silent drop polocy in Section 5.2 as requested by both Richard Barnes and Jari Arkko.

  * Fixed typos pointed out by Richard Barnes.

- Version 02 Posted July 2011.

  * Added text in Section 5 about ETR synchronization, as suggested by Alia Atlas.

  * Modified text in Section 8.4 concerning lightweight LISP implementation, as suggested by Alia Atlas.

  * Deleted text concerning old versions of [I-D.ietf-lisp-ms] and [I-D.ietf-lisp-alt] in Section 7, as pointed out by Alia Atlas.

  * Fixed section 4.1 to be less restrictive, as suggested by Jesper Skriver.

- Version 01 Posted March 2011.
  * Changed the wording from "Map-Version number 0" to "Null Map-Version.
  * Clarification of the use of the Null Map-Version value as Source Map-Version Number and Destination Map-Version Number.
  * Extended the section describing Map-Versioning and LISP Interworking co-existence.
  * Reduce packet format description to avoid double definitions with the main specs.
- Version 00 Posted September 2010.
  * Added Section "Definitions of Terms".
  * Editorial polishing of all sections.
  * Added clarifications in section "Dealing with Map-Version numbers" for the case of the special Map-Version number 0.
  * Rename of draft-iannone-mapping-versioning-02.txt.

Authors' Addresses

Luigi Iannone
Telekom Innovation Laboratories
Ernst-Reuter Platz 7
Berlin
Germany

Email: luigi@net.t-labs.tu-berlin.de

Damien Saucez
INRIA Sophia Antipolis
2004 route des Lucioles - BP 93
Sophia Antipolis
France

Email: damien.saucez@inria.fr

Olivier Bonaventure
Universite catholique de Louvain
Place St. Barbe 2
Louvain-la-Neuve
Belgium

Email: olivier.bonaventure@uclouvain.be

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: March 17, 2014

G. Schudel
cisco Systems
A. Jain
Juniper Networks
V. Moreno
cisco Systems

September 13, 2013

LISP MIB
draft-ietf-lisp-mib-13

Abstract

This document defines the MIB module that contains managed objects to support the monitoring devices that support the Locator/ID Separation Protocol (LISP). These objects provide information useful for monitoring LISP devices, including determining basic LISP configuration information, LISP functional status, and operational counters and other statistics.

Status of This Memo

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

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

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

This Internet-Draft will expire on March 17, 2014.

Copyright Notice

Table of Contents

1. Introduction

This document describes the Management Information Base (MIB) module for use with network management protocols in the Internet community. Specifically, the MIB for managing devices that support the Locator/ID Separation Protocol (LISP) is described.

LISP [RFC6830] specifies a network-based architecture and mechanisms that implement a new semantic for IP addressing using two separate name spaces: Endpoint Identifiers (EIDs), used within sites, and Routing Locators (RLOCs), used on the transit networks that make up the Internet infrastructure. To achieve this separation, LISP defines protocol mechanisms for mapping from EIDs to RLOCs.

From a data plane perspective, LISP traffic is handled exclusively at the network layer by devices performing Ingress Tunnel Router (ITR) and Egress Tunnel Router (ETR) LISP functions. Data plane operations performed by these devices are described in [RFC6830]. Additionally, data plane interworking between legacy (Internet) and LISP sites is implemented by devices performing Proxy ITR (PITR) and Proxy ETR (PETR) functions. The data plane operations of these devices is described in [RFC6832].

From a control plane perspective, LISP employs mechanisms related to creating, maintaining, and resolving mappings from EIDs to RLOCs. LISP ITRs, ETRs, PITRs, and PETRs perform specific control plane functions, and these control plane operations are described in [RFC6830]. Additionally, LISP infrastructure devices supporting LISP control plane functionality include Map-Servers and Map-Resolvers, and the control plane operations of these devices are described in [RFC6833].

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 [RFC2119].

3. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP).

Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIv2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

4. Definition of Terms

This document does not define any new terms. All terms used in this document are listed here for completeness; the authoritative definition of each term can be found in the definition section of the respective, specified reference.

Endpoint ID (EID): [RFC6830]

Routing Locator (RLOC): [RFC6830]

EID-to-RLOC Cache: [RFC6830]

EID-to-RLOC Database: [RFC6830]

Ingress Tunnel Router (ITR): [RFC6830]

Egress Tunnel Router (ETR): [RFC6830]

xTR: [RFC6830]

Proxy ITR (PITR): [RFC6832]

Proxy ETR (PETR): [RFC6832]

LISP Site: [RFC6830]

Map-Server: [RFC6833]

Map-Resolver: [RFC6833]

Map-Request: [RFC6833]

Map-Reply: [RFC6833]

Negative Map-Reply: [RFC6833]

5. LISP MIB Objectives

The objectives for this LISP MIB module are to provide a read-only mechanism to support the following functions:

- Provide a means for obtaining (read-only) a current status of LISP features enabled on a device, and (read-only) a current status of configuration attributes related to those features. As one example, this MIB could determine the ON/OFF status of LISP features such as ITR, ETR, PITR, PETR, MS or MR support, specifically as realated to both IPv4 or IPv6 address families. Other examples could include: obtaining the (read-only) status of whether rloc-probing is enabled, whether the use of a PETR is configured, and obtaining the (read-only) values of other related attributes such as the map-cache limit value, or a mapping time-to-live value.

- Provide a means for obtaining (read-only) the current attributes of various LISP tables, such as the EID-to-RLOC policy data contained in the Map-Cache, or the local EID-to-RLOC policy data contained in the Mapping-Database.

- Provide a means for obtaining (read-only) the current operational statistics of various LISP functions, such as the number of packets encapsulated and decapsulated by the device. Other counters of operational interest, depending on LISP function, include things like the current number of map-cache entries, and the total number and rate of map-requests received and sent by the device.

6. Structure of LISP MIB Module

6.1. Overview of Defined Notifications

No LISP MIB notifications are defined.

6.2. Overview of Defined Tables

The LISP MIB module is composed of the following tables of objects:

lispFeatures - This table provides information representing the various lisp features that can be enabled on LISP devices.

lispIidToVrf - This table provides information representing the mapping of a LISP instance ID to a VRF (Virtual Routing/ Forwarding).

lispGlobalStats - This table provides global statistics for a given Instance ID per address-family on a LISP device.

lispMappingDatabase - This table represents the EID-to-RLOC database that contains the EID-prefix to RLOC mappings configured on an ETR. In general, this table would be representative of all such mappings for a given site that this device belongs to.

lispMappingDatabaseLocator - This table represents the set of routing locators contained in the EID-to-RLOC database configured on an ETR.

lispMapCache - This table represents the short-lived, on-demand table maintained on an ITR that stores, tracks, and times-out EID-to-RLOC mappings.

lispMapCacheLocator - This table represents the set of locators per EID prefix contained in the map-cache table of an ITR.

lispConfiguredLocator - This table represents the set of routing locators configured on a LISP device.

lispEidRegistration - This table provides the properties of each EID prefix that is registered with this device when configured to be a Map-Server.

lispEidRegistrationEtr - This table provides the properties of the different ETRs that send registers, for a given EID prefix, to this device when configured to be a Map-Server.

lispEidRegistrationLocator - This table provides the properties of the different locators per EID prefix that is registered with this device when configured to be a Map-Server.

lispUseMapServer - This table provides the properties of all Map-Servers that this device is configured to use.

lispUseMapResolver - This table provides the properties of all Map-Resolvers that this device is configured to use.

lispUseProxyEtr - This table provides the properties of all Proxy ETRs that this device is configured to use.

7. LISP MIB Definitions

```
LISP-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE,
    mib-2, Unsigned32, Counter64,
```

```
    Integer32, TimeTicks            FROM SNMPv2-SMI        -- [RFC2578]
    TruthValue, TEXTUAL-CONVENTION,
    TimeStamp                       FROM SNMPv2-TC         -- [RFC2579]
    MODULE-COMPLIANCE, OBJECT-GROUP FROM SNMPv2-CONF       -- [RFC2580]
    MplsL3VpnName
                 FROM MPLS-L3VPN-STD-MIB                   -- [RFC4382]
    AddressFamilyNumbers
                 FROM IANA-ADDRESS-FAMILY-NUMBERS-MIB;     --
         http://www.iana.org/assignments/ianaaddressfamilynumbers-mib

lispMIB MODULE-IDENTITY
    LAST-UPDATED "201309130000Z" -- 13 September 2013
    ORGANIZATION
            "IETF Locator/ID Separation Protocol (LISP) Working Group"
    CONTACT-INFO
            "Email: lisp@ietf.org
            WG charter:
            http://www.ietf.org/html.charters/lisp-charter.html"
    DESCRIPTION
            "This MIB module contains managed objects to support
             monitoring devices that support the Locator/ID Separation
             Protocol (LISP).

            Copyright (C) The IETF Trust (2013)."
    REVISION     "201309130000Z" -- 13 September 2013
    DESCRIPTION  "Initial version of the IETF LISP-MIB module. Published
                  as RFC xxxx."
-- RFC Ed.: RFC-editor pls fill in xxxx
     ::= { mib-2 XXX }
-- RFC Ed.: assigned by IANA, see section 10 for details

--
-- Textual Conventions
--

LispAddressType ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "39a"
    STATUS current
    DESCRIPTION
        "LISP architecture can be applied to a wide variety of
         address-families. This textual-convention is a generalization
         for representing addresses belonging to those address-families.
         For convenience, this document refers to any such address as a
         LISP address. LispAddressType textual-convention consists of
         the following four-tuple:
          1. IANA Address Family Number: A field of length 2-octets,
             whose value is of the form following the assigned
             AddressFamilyNumbers textual-convention described in
```

IANA-ADDRESS-FAMILY-NUMBERS-MIB DEFINITIONS [IANA] http://www.iana.org/assignments/ianaaddressfamilynumbers-mib. The enumerations are also listed in [IANA]. Note that this list of address family numbers is maintained by IANA.

2. Length of LISP address: A field of length 1-octet, whose value indicates the octet-length of the next (third) field of this LispAddressType four-tuple.
3. LISP address: A field of variable length as indicated in the previous (second) field, whose value is an address of the IANA Address Family indicated in the first field of this LispAddressType four-tuple. Note that any of the IANA Address Families can be represented. Particularly when the address family is LISP Canonical Address Format (LCAF) [LCAF] http://tools.ietf.org/id/draft-ietf-lisp-lcaf-02.txt with IANA assigned Address Family Number 16387, then the first octet of this field indicates the LCAF type, and the rest of this field is same as the encoding format of the LISP Canonical Address after the length field, as defined in [LCAF].
4. Mask-length of address: A field of length 1-octet, whose value is the mask-length to be applied to the LISP address specified in the previous (third) field.

To illustrate the use of this object, consider the LISP MIB Object below entitled lispMapCacheEntry. This object begins with the following entities:

```
lispMapCacheEntry ::= SEQUENCE {
   lispMapCacheEidLength          INTEGER,
   lispMapCacheEid                LispAddressType,
    ... [skip] ...
```

Example 1: Suppose that the IPv4 EID prefix stored is 192.0.2.0/24. In this case, the values within lispMapCacheEntry would be:

```
   lispMapCacheEidLength  = 8
   lispMapCacheEid = 1, 4, 192.0.2.0, 24
   ... [skip] ...
```

where 8 is the total length in octets of the next object (lispMapCacheEID of type LispAddressType). Then, the value 1 indicates the IPv4 AF (per [IANA]), the value 4 indicates that the AF is 4-octets in length, 192.0.2.0 is the IPv4 address, and the value 24 is the mask-length in bits. Note that the lispMapCacheEidLength value of 8 is used to compute the length of the fourth

(last) field in lispMapCacheEid to be 1 octet - as computed by 8 - (2 + 1 + 4) = 1.

Example 2: Suppose that the IPv6 EID prefix stored is 2001:db8:a::/48. In this case, the values within lispMapCacheEntry would be:

```
lispMapCacheEidLength  = 20
lispMapCacheEid = 2, 16, 2001:db8:a::, 48
... [skip] ...
```

where 20 is the total length in octets of the next object (lispMapCacheEID of type LispAddressType). Then, the value 2 indicates the IPv4 AF (per [IANA]), the value 16 indicates that the AF is 16-octets in length, 2001:db8:a:: is the IPv6 address, and the value 48 is the mask-length in bits. Note that the lispMapCacheEidLength value of 20 is used to compute the length of the fourth (last) field in lispMapCacheEid to be 1 octet - as computed by 20 - (2 + 1 + 16) = 1.

Example 3: As an example where LCAF is used, suppose that the IPv4 EID prefix stored is 192.0.2.0/24 and it is part of LISP Instance ID 101. In this case, the values within lispMapCacheEntry would be:

```
lispMapCacheEidLength  = 11
lispMapCacheEid = 16387, 7, 2, 101, 1, 192.0.2.0, 24
... [skip] ...
```

where 11 is the total length in octets of the next object (lispMapCacheEID of type LispAddressType). Then, the value 16387 indicates the LCAF AF (see [IANA]), the value 7 indicates that the LCAF AF is 7-octets in length in this case, 2 indicates that LCAF Type 2 encoding is used (see [LCAF]), 101 gives the Instance ID, 1 gives the AFI (per [IANA]) for an IPv4 address, 192.0.2.0 is the IPv4 address, and 24 is the mask-length in bits. Note that the lispMapCacheEidLength value of 11 octets is used to compute the length of the last field in lispMapCacheEid to be 1 octet, as computed by 11 - (2 + 1 + 1 + 1 + 1 + 4) = 1.

Note: all LISP header formats and locations of specific flags, bits, and fields are as given in the base LISP references of RFC6830, RFC6832, and RFC6833."

REFERENCE
"RFC6830, Section 14.2, draft-ietf-lisp-lcaf-02.txt."

```
    SYNTAX OCTET STRING (SIZE (5..39))


   --
   -- Top level components of this MIB.
   --
   lispObjects       OBJECT IDENTIFIER ::= { lispMIB 1 }
   lispConformance   OBJECT IDENTIFIER ::= { lispMIB 2 }



    lispFeaturesTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF LispFeaturesEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This table represents the ON/OFF status of the
            various LISP features that can be enabled on LISP devices."
        REFERENCE
            "RFC6830, Section 4.0., Section 5.5., Section 6.3."
        ::= { lispObjects 1 }

    lispFeaturesEntry OBJECT-TYPE
        SYNTAX     LispFeaturesEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "An entry (conceptual row) in the lispFeaturesTable."
        INDEX      { lispFeaturesInstanceID,
                     lispFeaturesAddressFamily }
        ::= { lispFeaturesTable 1 }

    LispFeaturesEntry ::= SEQUENCE {
        lispFeaturesInstanceID                    Unsigned32,
        lispFeaturesAddressFamily                 AddressFamilyNumbers,
        lispFeaturesItrEnabled                    TruthValue,
        lispFeaturesEtrEnabled                    TruthValue,
        lispFeaturesProxyItrEnabled               TruthValue,
        lispFeaturesProxyEtrEnabled               TruthValue,
        lispFeaturesMapServerEnabled              TruthValue,
        lispFeaturesMapResolverEnabled            TruthValue,
        lispFeaturesMapCacheSize                  Unsigned32,
        lispFeaturesMapCacheLimit                 Unsigned32,
        lispFeaturesEtrMapCacheTtl                Unsigned32,
        lispFeaturesRlocProbeEnabled              TruthValue,
        lispFeaturesEtrAcceptMapDataEnabled       TruthValue,
        lispFeaturesEtrAcceptMapDataVerifyEnabled TruthValue,
        lispFeaturesRouterTimeStamp               TimeStamp
```

```
    }

    lispFeaturesInstanceID OBJECT-TYPE
        SYNTAX     Unsigned32 (0..16777215)
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This represents the Instance ID of the LISP header.
            An Instance ID in the LISP address encoding helps
            uniquely identify the AFI-based address space to which
            a given EID belongs. It's default value is 0."
         DEFVAL { 0 }
         ::= { lispFeaturesEntry 1 }

    lispFeaturesAddressFamily OBJECT-TYPE
        SYNTAX     AddressFamilyNumbers
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "The IANA address family number of destination address
            of packets that this LISP device is enabled to process."
         ::= { lispFeaturesEntry 2 }

    lispFeaturesItrEnabled OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates the status of ITR role on this device. If
            this object is true, then ITR feature is enabled."
        ::= { lispFeaturesEntry 3 }

    lispFeaturesEtrEnabled OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates the status of ETR role on this device. If
            this object is true, then ETR feature is enabled."
        ::= { lispFeaturesEntry 4 }

    lispFeaturesProxyItrEnabled OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates the status of Proxy-ITR role on this device.
            If this object is true, then Proxy-ITR feature is enabled."
```

```
        ::= { lispFeaturesEntry 5 }

    lispFeaturesProxyEtrEnabled OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates the status of Proxy-ETR role on this device.
            If this object is true, then Proxy-ETR feature is enabled."
        ::= { lispFeaturesEntry 6 }

    lispFeaturesMapServerEnabled OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates the status of Map Server role on this device.
            If this object is true, then Map Server feature is
            enabled."
        ::= { lispFeaturesEntry 7 }

    lispFeaturesMapResolverEnabled OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates the status of Map Resolver role on this device.
            If this object is true, then Map Resolver feature is
            enabled."
        ::= { lispFeaturesEntry 8 }

    lispFeaturesMapCacheSize OBJECT-TYPE
        SYNTAX     Unsigned32
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Size of EID-to-RLOC map cache on this device."
        ::= { lispFeaturesEntry 9 }

    lispFeaturesMapCacheLimit OBJECT-TYPE
        SYNTAX     Unsigned32
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Maximum permissible entries in EID-to-RLOC map cache on
            this device."
        ::= { lispFeaturesEntry 10 }
```

```
    lispFeaturesEtrMapCacheTtl OBJECT-TYPE
        SYNTAX     Unsigned32
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The stored Record TTL of the EID-to-RLOC map record in
            the map cache."
        ::= { lispFeaturesEntry 11 }

    lispFeaturesRlocProbeEnabled OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates the status of rloc-probing feature on this
            device.  If this object is true, then this feature is
            enabled."
        ::= { lispFeaturesEntry 12 }

    lispFeaturesEtrAcceptMapDataEnabled OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates the status of accepting piggybacked mapping
            data received in a map-request on this device. If this
            object is true, then this device accepts piggybacked
            mapping data."
        ::= { lispFeaturesEntry 13 }

    lispFeaturesEtrAcceptMapDataVerifyEnabled OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates the status of verifying accepted piggybacked
            mapping data received in a map-request on this device.
            If this object is true, then this device verifies
            accepted piggybacked mapping data."
        ::= { lispFeaturesEntry 14 }

    lispFeaturesRouterTimeStamp OBJECT-TYPE
        SYNTAX     TimeStamp
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The value of sysUpTime at which LISP feature was
            enabled on this device.
```

```
            If this information was present at the most recent
            re-initialization of the local management subsystem,
            then this object contains a zero value."
        DEFVAL { 0 }
        ::= { lispFeaturesEntry 15 }



      lispIidToVrfTable OBJECT-TYPE
          SYNTAX     SEQUENCE OF LispIidToVrfEntry
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
              "This table represents the mapping of LISP Instance ID
              to a VRF."
          REFERENCE
              "RFC6830, Section 5.5. and RFC4382, Section 7."
          ::= { lispObjects 2 }

      lispIidToVrfEntry OBJECT-TYPE
          SYNTAX     LispIidToVrfEntry
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
              "An entry (conceptual row) in the lispIidToVrfTable."
          INDEX      { lispFeaturesInstanceID }
          ::= { lispIidToVrfTable 1 }

      LispIidToVrfEntry ::= SEQUENCE {
          lispIidToVrfName                    MplsL3VpnName
      }

      lispIidToVrfName OBJECT-TYPE
          SYNTAX     MplsL3VpnName
          MAX-ACCESS read-only
          STATUS     current
          DESCRIPTION
              "The identifier for each VPN that is mapped to the
              given LISP Instance ID."
              ::= { lispIidToVrfEntry 1 }


    lispGlobalStatsTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF LispGlobalStatsEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This table provides global statistics for a given
```

```
            Instance ID per address-family on a LISP device."
        REFERENCE
            "RFC6830, Section 6.1."
        ::= { lispObjects 3 }

    lispGlobalStatsEntry OBJECT-TYPE
        SYNTAX     LispGlobalStatsEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "An entry (conceptual row) in the
            lispGlobalStatsTable."
        INDEX      { lispFeaturesInstanceID,
                     lispFeaturesAddressFamily }
        ::= { lispGlobalStatsTable 1 }

    LispGlobalStatsEntry ::= SEQUENCE {
        lispGlobalStatsMapRequestsIn        Counter64,
        lispGlobalStatsMapRequestsOut       Counter64,
        lispGlobalStatsMapRepliesIn         Counter64,
        lispGlobalStatsMapRepliesOut        Counter64,
        lispGlobalStatsMapRegistersIn       Counter64,
        lispGlobalStatsMapRegistersOut      Counter64
    }

    lispGlobalStatsMapRequestsIn OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Total number of map requests received by this device for
            any EID prefix of the given address family and Instance ID.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of LISP features
            being removed, which can be detected by observing the value
            of lispFeaturesRouterTimeStamp."
        ::= { lispGlobalStatsEntry 1 }

    lispGlobalStatsMapRequestsOut OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Total number of map requests sent by this device for any
            EID prefix of the given address family and Instance ID.
```

```
            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of LISP features
            being removed, which can be detected by observing the value
            of lispFeaturesRouterTimeStamp."
        ::= { lispGlobalStatsEntry 2 }

    lispGlobalStatsMapRepliesIn OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Total number of map replies received by this device for any
            EID prefix of the given address family and Instance ID.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of LISP features
            being removed, which can be detected by observing the value
            of lispFeaturesRouterTimeStamp."
        ::= { lispGlobalStatsEntry 3 }

    lispGlobalStatsMapRepliesOut OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Total number of map replies sent by this device for any EID
            prefix of the given address family and Instance ID.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of LISP features
            being removed, which can be detected by observing the value
            of lispFeaturesRouterTimeStamp."
        ::= { lispGlobalStatsEntry 4 }

    lispGlobalStatsMapRegistersIn OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Total number of map registers received by this device for
            any EID prefix of the given address family and Instance ID.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of LISP features
```

```
            being removed, which can be detected by observing the value
            of lispFeaturesRouterTimeStamp."
        ::= { lispGlobalStatsEntry 5 }

    lispGlobalStatsMapRegistersOut OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Total number of map registers sent by this device for any
            EID prefix of the given address family and Instance ID.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of LISP features
            being removed, which can be detected by observing the value
            of lispFeaturesRouterTimeStamp."
        ::= { lispGlobalStatsEntry 6 }


    lispMappingDatabaseTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF LispMappingDatabaseEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This table represents the EID-to-RLOC mapping database
             that contains the EID-prefix to RLOC mappings configured
             on an ETR.

            This table represents all such mappings for the given LISP
            site to which this device belongs."
        REFERENCE
            "RFC6830, Section 6.0."
        ::= { lispObjects 4 }

    lispMappingDatabaseEntry OBJECT-TYPE
        SYNTAX     LispMappingDatabaseEntry

        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "An entry (conceptual row) in lispMappingDatabaseTable."
        INDEX   { lispMappingDatabaseEidLength,
                  lispMappingDatabaseEid }
        ::= { lispMappingDatabaseTable 1 }

    LispMappingDatabaseEntry ::= SEQUENCE {
        lispMappingDatabaseEidLength        Integer32,
```

```
        lispMappingDatabaseEid              LispAddressType,
        lispMappingDatabaseLsb              Unsigned32,
        lispMappingDatabaseEidPartitioned   TruthValue,
        lispMappingDatabaseTimeStamp        TimeStamp,
        lispMappingDatabaseDecapOctets      Counter64,
        lispMappingDatabaseDecapPackets     Counter64,
        lispMappingDatabaseEncapOctets      Counter64,
        lispMappingDatabaseEncapPackets     Counter64
    }

    lispMappingDatabaseEidLength OBJECT-TYPE
        SYNTAX     Integer32 (5..39)
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This object gives the octet-length of
            lispMappingDatabaseEid."
        ::= { lispMappingDatabaseEntry 1 }

    lispMappingDatabaseEid OBJECT-TYPE
        SYNTAX     LispAddressType
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "The EID prefix of the mapping database."
        ::= { lispMappingDatabaseEntry 2 }

    lispMappingDatabaseLsb OBJECT-TYPE
        SYNTAX     Unsigned32 (0..4294967295)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The locator status bits for this EID prefix."
        ::= { lispMappingDatabaseEntry 3 }

    lispMappingDatabaseEidPartitioned OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only

        STATUS     current
        DESCRIPTION
            "Indicates if this device is partitioned from the site that
            contains this EID prefix. If this object is true, then it
            means this device is partitioned from the site."
        ::= { lispMappingDatabaseEntry 4 }

    lispMappingDatabaseTimeStamp OBJECT-TYPE
        SYNTAX     TimeStamp
```

```
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The value of sysUpTime at which the EID Prefix information
            represented by this mapping database entry was configured
            on this device.

            If this information was present at the most recent
            re-initialization of the local management subsystem, then
            this object contains a zero value."
        DEFVAL { 0 }
        ::= { lispMappingDatabaseEntry 5 }

    lispMappingDatabaseDecapOctets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of octets, after decapsulation, of LISP packets
            that were decapsulated by this device addressed to a host
            within this EID-prefix.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of LISP features
            being removed, which can be detected by observing the value
            of lispMappingDatabaseTimeStamp."
        ::= { lispMappingDatabaseEntry 6 }

    lispMappingDatabaseDecapPackets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of LISP packets that were decapsulated by this
            device addressed to a host within this EID-prefix.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of LISP features
            being removed, which can be detected by observing the value
            of lispMappingDatabaseTimeStamp."
        ::= { lispMappingDatabaseEntry 7 }

    lispMappingDatabaseEncapOctets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
```

```
        DESCRIPTION
            "The number of octets, before encapsulation, of LISP packets
            that were encapsulated by this device, whose inner header
            source address matched this EID prefix.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of LISP features
            being removed, which can be detected by observing the value
            of lispMappingDatabaseTimeStamp."
        ::= { lispMappingDatabaseEntry 8 }

    lispMappingDatabaseEncapPackets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of LISP packets that were encapsulated by this
            device whose inner header source address matched this EID
            prefix.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of LISP features
            being removed, which can be detected by observing the value
            of lispMappingDatabaseTimeStamp."
        ::= { lispMappingDatabaseEntry 9 }


    lispMappingDatabaseLocatorTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF LispMappingDatabaseLocatorEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This table represents the set of routing locators per EID
            prefix contained in the EID-to-RLOC database configured on
            this ETR."
        REFERENCE
            "RFC6830, Section 6.2."
        ::= { lispObjects 5 }

    lispMappingDatabaseLocatorEntry OBJECT-TYPE
        SYNTAX     LispMappingDatabaseLocatorEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "An entry (conceptual row) in the
            lispMappingDatabaseLocatorTable."
```

```
        INDEX  { lispMappingDatabaseEidLength,
                 lispMappingDatabaseEid,
                 lispMappingDatabaseLocatorRlocLength,
                 lispMappingDatabaseLocatorRloc }
        ::= { lispMappingDatabaseLocatorTable 1 }

    LispMappingDatabaseLocatorEntry ::= SEQUENCE {
        lispMappingDatabaseLocatorRlocLength        Integer32,
        lispMappingDatabaseLocatorRloc              LispAddressType,
        lispMappingDatabaseLocatorRlocPriority      Integer32,
        lispMappingDatabaseLocatorRlocWeight        Integer32,
        lispMappingDatabaseLocatorRlocMPriority     Integer32,
        lispMappingDatabaseLocatorRlocMWeight       Integer32,
        lispMappingDatabaseLocatorRlocState         INTEGER,
        lispMappingDatabaseLocatorRlocLocal         INTEGER,
        lispMappingDatabaseLocatorRlocTimeStamp     TimeStamp,
        lispMappingDatabaseLocatorRlocDecapOctets   Counter64,
        lispMappingDatabaseLocatorRlocDecapPackets  Counter64,
        lispMappingDatabaseLocatorRlocEncapOctets   Counter64,
        lispMappingDatabaseLocatorRlocEncapPackets  Counter64
    }

    lispMappingDatabaseLocatorRlocLength OBJECT-TYPE
        SYNTAX     Integer32 (5..39)
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This object is used to get the octet-length of
            lispMappingDatabaseLocatorRloc."
        ::= { lispMappingDatabaseLocatorEntry 1 }

    lispMappingDatabaseLocatorRloc OBJECT-TYPE
        SYNTAX     LispAddressType
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This object is a locator for the given EID prefix in
            the mapping database."
        ::= { lispMappingDatabaseLocatorEntry 2 }

    lispMappingDatabaseLocatorRlocPriority OBJECT-TYPE
        SYNTAX     Integer32 (0..255)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The unicast priority of the RLOC."
        ::= { lispMappingDatabaseLocatorEntry 3 }
```

```
   lispMappingDatabaseLocatorRlocWeight OBJECT-TYPE
       SYNTAX     Integer32 (0..100)
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
           "The unicast weight of the RLOC."
       ::= { lispMappingDatabaseLocatorEntry 4 }

   lispMappingDatabaseLocatorRlocMPriority OBJECT-TYPE
       SYNTAX     Integer32 (0..255)
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
           "The multicast priority of the RLOC."
       ::= { lispMappingDatabaseLocatorEntry 5 }

   lispMappingDatabaseLocatorRlocMWeight OBJECT-TYPE
       SYNTAX     Integer32 (0..100)
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
           "The multicast weight of the RLOC."
       ::= { lispMappingDatabaseLocatorEntry 6 }

   lispMappingDatabaseLocatorRlocState OBJECT-TYPE
       SYNTAX     INTEGER {
                     up (1),
                     down (2),
                     unreachable (3)
                  }
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
           "The state of this RLOC as per this device.
           (1 = RLOC is up; 2 = RLOC is down; 3 = RLOC is unreachable)."
       ::= { lispMappingDatabaseLocatorEntry 7 }

   lispMappingDatabaseLocatorRlocLocal OBJECT-TYPE
       SYNTAX     INTEGER {
                     siteself (1),
                     sitelocal (2)
                  }
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
           "Indicates whether the RLOC is local to this device
           (or remote, meaning local to another device in the same LISP
           site). (1 = RLOC is an address on this device; 2 = RLOC is
```

```
            an address on another device)."
        ::= { lispMappingDatabaseLocatorEntry 8 }

    lispMappingDatabaseLocatorRlocTimeStamp OBJECT-TYPE
        SYNTAX     TimeStamp
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The value of sysUpTime at which the RLOC of the EID Prefix
            represented by this mapping database entry was configured
            on this device.

            If this information was present at the most recent
            re-initialization of the local management subsystem, then
            this object contains a zero value."
        DEFVAL { 0 }
        ::= { lispMappingDatabaseLocatorEntry 9 }

    lispMappingDatabaseLocatorRlocDecapOctets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of octets of LISP packets that were
            addressed to this RLOC of the EID-prefix and
            were decapsulated.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of database
            mappings getting re-configured or RLOC status changes, which
            can be detected by observing the value of
            lispMappingDatabaseLocatorRlocTimeStamp."
        ::= { lispMappingDatabaseLocatorEntry 10 }

    lispMappingDatabaseLocatorRlocDecapPackets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of LISP packets that were addressed to this RLOC
            of the EID-prefix and were decapsulated.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of database
            mappings getting re-configured or RLOC status changes, which
            can be detected by observing the value of
```

```
            lispMappingDatabaseLocatorRlocTimeStamp."
        ::= { lispMappingDatabaseLocatorEntry 11 }

    lispMappingDatabaseLocatorRlocEncapOctets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of octets of LISP packets that were encapsulated
            by this device using this RLOC address as the source, and
            that were sourced by an address of this EID-prefix.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of database
            mappings getting re-configured or RLOC status changes, which
            can be detected by observing the value of
            lispMappingDatabaseLocatorRlocTimeStamp."
        ::= { lispMappingDatabaseLocatorEntry 12 }

    lispMappingDatabaseLocatorRlocEncapPackets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of LISP packets that were encapsulated by this
            device using this RLOC address as the source, and that were
            sourced by an address of this EID-prefix.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of database
            mappings getting re-configured or RLOC status changes, which
            can be detected by observing the value of
            lispMappingDatabaseLocatorRlocTimeStamp."
        ::= { lispMappingDatabaseLocatorEntry 13 }


     lispMapCacheTable OBJECT-TYPE
         SYNTAX     SEQUENCE OF LispMapCacheEntry
         MAX-ACCESS not-accessible
         STATUS     current
         DESCRIPTION
             "This table represents the short-lived, on-demand table on
             an ITR that stores, tracks, and is responsible for
             timing-out and otherwise validating EID-to-RLOC mappings."
         REFERENCE
             "RFC6830, Section 6.0., Section 12.0."
```

```
        ::= { lispObjects 6 }

    lispMapCacheEntry OBJECT-TYPE
        SYNTAX     LispMapCacheEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "An entry (conceptual row) in the
            lispMapCacheTable."
        INDEX      { lispMapCacheEidLength,
                     lispMapCacheEid }
        ::= { lispMapCacheTable 1 }

    LispMapCacheEntry ::= SEQUENCE {
        lispMapCacheEidLength           Integer32,
        lispMapCacheEid                 LispAddressType,
        lispMapCacheEidTimeStamp        TimeStamp,
        lispMapCacheEidExpiryTime       TimeTicks,
        lispMapCacheEidState            TruthValue,
        lispMapCacheEidAuthoritative    TruthValue,
        lispMapCacheEidDecapOctets      Counter64,
        lispMapCacheEidDecapPackets     Counter64,
        lispMapCacheEidEncapOctets      Counter64,
        lispMapCacheEidEncapPackets     Counter64
    }

    lispMapCacheEidLength OBJECT-TYPE
        SYNTAX     Integer32 (5..39)
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This object is used to get the octet-length of
            lispMapCacheEid."
        ::= { lispMapCacheEntry 1 }

    lispMapCacheEid OBJECT-TYPE
        SYNTAX     LispAddressType
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "The EID prefix in the mapping cache."
        ::= { lispMapCacheEntry 2 }

    lispMapCacheEidTimeStamp OBJECT-TYPE
        SYNTAX     TimeStamp
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
```

```
            "The value of sysUpTime at which the EID Prefix information
            represented by this entry was learned by this device.

            If this information was present at the most recent
            re-initialization of the local management subsystem, then
            this object contains a zero value."
        DEFVAL { 0 }
        ::= { lispMapCacheEntry 3 }

    lispMapCacheEidExpiryTime OBJECT-TYPE
        SYNTAX     TimeTicks
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The time remaining before the ITR times-out this
            EID prefix."
        ::= { lispMapCacheEntry 4 }

    lispMapCacheEidState OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "This object is used to indicate the activity of this EID
            prefix. If this object is true, then it means this EID
            prefix is seeing activity."
        ::= { lispMapCacheEntry 5 }

    lispMapCacheEidAuthoritative OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "This object is used to indicate whether the EID prefix was
            installed by an authoritative map-reply. If this object is
            true, then it means this EID prefix was installed by an
            authoritative map-reply."
        ::= { lispMapCacheEntry 6 }

    lispMapCacheEidDecapOctets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of octets of LISP packets that were decapsulated
            by this device and were sourced from a remote host within
            this EID-prefix.
```

```
            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of cache being
            removed and replaced, which can be detected by observing the
            value of lispMapCacheEidTimeStamp."
        ::= { lispMapCacheEntry 7 }

    lispMapCacheEidDecapPackets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of LISP packets that were decapsulated by this
            device and were sourced from a remote host within this
            EID-prefix.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of cache being
            removed and replaced, which can be detected by observing the
            value of lispMapCacheEidTimeStamp."
        ::= { lispMapCacheEntry 8 }

    lispMapCacheEidEncapOctets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of octets of LISP packets that were encapsulated
            by this device using the given EID-prefix in the map cache.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of cache being
            removed and replaced, which can be detected by observing the
            value of lispMapCacheEidTimeStamp."
        ::= { lispMapCacheEntry 9 }

    lispMapCacheEidEncapPackets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of LISP packets that were encapsulated by this
            device using the given EID-prefix in the map cache.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
```

```
            Discontinuities can also occur as a result of cache being
            removed and replaced, which can be detected by observing the
            value of lispMapCacheEidTimeStamp."
        ::= { lispMapCacheEntry 10 }


    lispMapCacheLocatorTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF LispMapCacheLocatorEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This table represents the set of locators per EID prefix
            contained in the map-cache table of an ITR."
        REFERENCE
            "RFC6830, Section 6.3."
        ::= { lispObjects 7 }

    lispMapCacheLocatorEntry OBJECT-TYPE
        SYNTAX     LispMapCacheLocatorEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "An entry (conceptual row) in the
            lispMapCacheLocatorTable."
        INDEX      { lispMapCacheEidLength,
                     lispMapCacheEid,
                     lispMapCacheLocatorRlocLength,
                     lispMapCacheLocatorRloc }
        ::= { lispMapCacheLocatorTable 1 }

    LispMapCacheLocatorEntry ::= SEQUENCE {
        lispMapCacheLocatorRlocLength              Integer32,
        lispMapCacheLocatorRloc                    LispAddressType,
        lispMapCacheLocatorRlocPriority            Integer32,
        lispMapCacheLocatorRlocWeight              Integer32,
        lispMapCacheLocatorRlocMPriority           Integer32,
        lispMapCacheLocatorRlocMWeight             Integer32,
        lispMapCacheLocatorRlocState               INTEGER,
        lispMapCacheLocatorRlocTimeStamp           TimeStamp,
        lispMapCacheLocatorRlocLastPriorityChange  TimeTicks,
        lispMapCacheLocatorRlocLastWeightChange    TimeTicks,
        lispMapCacheLocatorRlocLastMPriorityChange TimeTicks,
        lispMapCacheLocatorRlocLastMWeightChange   TimeTicks,
        lispMapCacheLocatorRlocLastStateChange     TimeTicks,
        lispMapCacheLocatorRlocRtt                 TimeTicks,
        lispMapCacheLocatorRlocDecapOctets         Counter64,
        lispMapCacheLocatorRlocDecapPackets        Counter64,
        lispMapCacheLocatorRlocEncapOctets         Counter64,
```

```
        lispMapCacheLocatorRlocEncapPackets        Counter64
    }

    lispMapCacheLocatorRlocLength OBJECT-TYPE
        SYNTAX     Integer32 (5..39)
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This object is used to get the octet-length of
            lispMapCacheLocatorRloc."
        ::= { lispMapCacheLocatorEntry 1 }

    lispMapCacheLocatorRloc OBJECT-TYPE
        SYNTAX     LispAddressType
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "The locator for the EID prefix in the mapping cache."
        ::= { lispMapCacheLocatorEntry 2 }

    lispMapCacheLocatorRlocPriority OBJECT-TYPE
        SYNTAX     Integer32 (0..255)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The unicast priority of the RLOC for this EID prefix
            (0-255); lower more preferred. "
        ::= { lispMapCacheLocatorEntry 3 }

    lispMapCacheLocatorRlocWeight OBJECT-TYPE
        SYNTAX     Integer32 (0..100)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The unicast weight of the RLOC for this EID prefix
            (0 - 100) percentage. "
        ::= { lispMapCacheLocatorEntry 4 }

    lispMapCacheLocatorRlocMPriority OBJECT-TYPE
        SYNTAX     Integer32 (0..255)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The multicast priority of the RLOC for this EID prefix
            (0-255); lower more preferred."
        ::= { lispMapCacheLocatorEntry 5 }

    lispMapCacheLocatorRlocMWeight OBJECT-TYPE
```

```
        SYNTAX     Integer32 (0..100)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The multicast weight of the RLOC for this EID prefix
            (0 - 100) percentage."
        ::= { lispMapCacheLocatorEntry 6 }

    lispMapCacheLocatorRlocState OBJECT-TYPE
        SYNTAX     INTEGER {
                     up (1),
                     down (2),
                     unreachable (3)
                   }
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The state of this RLOC as per this device
            (1 = RLOC is up; 2 = RLOC is down; 3 = RLOC is unreachable)."
        ::= { lispMapCacheLocatorEntry 7 }

    lispMapCacheLocatorRlocTimeStamp OBJECT-TYPE
        SYNTAX     TimeStamp
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The value of sysUpTime at which the RLOC of EID prefix
            information represented by this entry was learned by
            this device.

            If this information was present at the most recent
            re-initialization of the local management subsystem,
            then this object contains a zero value."
        DEFVAL { 0 }
        ::= { lispMapCacheLocatorEntry 8 }

    lispMapCacheLocatorRlocLastPriorityChange OBJECT-TYPE
        SYNTAX     TimeTicks
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Time elapsed since the last change of the unicast priority
            of the RLOC for this EID prefix. Note that this is
            independent of lispMapCacheLocatorRlocTimeStamp."
        ::= { lispMapCacheLocatorEntry 9 }

    lispMapCacheLocatorRlocLastWeightChange OBJECT-TYPE
        SYNTAX     TimeTicks
```

```
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Time elapsed since the last change of the unicast weight
            of the RLOC for this EID prefix.  Note that this is
            independent of lispMapCacheLocatorRlocTimeStamp."
        ::= { lispMapCacheLocatorEntry 10 }

    lispMapCacheLocatorRlocLastMPriorityChange OBJECT-TYPE
        SYNTAX     TimeTicks
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Time since the last change of the multicast priority of the
            RLOC for this EID prefix."
        ::= { lispMapCacheLocatorEntry 11 }

    lispMapCacheLocatorRlocLastMWeightChange OBJECT-TYPE
        SYNTAX     TimeTicks
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Time since the last change of the multicast weight of the
            RLOC for this EID prefix."
        ::= { lispMapCacheLocatorEntry 12 }

    lispMapCacheLocatorRlocLastStateChange OBJECT-TYPE
        SYNTAX     TimeTicks
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Time since the last change of the up/down state of the
            RLOC for this EID prefix."
        ::= { lispMapCacheLocatorEntry 13 }

    lispMapCacheLocatorRlocRtt OBJECT-TYPE
        SYNTAX     TimeTicks
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Round trip time of RLOC probe and map-reply for this RLOC
            address for this prefix."
        ::= { lispMapCacheLocatorEntry 14 }

    lispMapCacheLocatorRlocDecapOctets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
```

```
        DESCRIPTION
            "The number of octets of LISP packets that were decapsulated
            by this device and were sourced from a remote host within
            this EID-prefix and were encapsulated for this RLOC.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of RLOC of cache
            being removed and replaced, which can be detected by
            observing the value of lispMapCacheLocatorRlocTimeStamp."
        ::= { lispMapCacheLocatorEntry 15 }

    lispMapCacheLocatorRlocDecapPackets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of LISP packets that were decapsulated by this
            device and were sourced from a remote host within this
            EID-prefix and were encapsulated for this RLOC.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of RLOC of cache
            being removed and replaced, which can be detected by
            observing the value of lispMapCacheLocatorRlocTimeStamp."
        ::= { lispMapCacheLocatorEntry 16 }

    lispMapCacheLocatorRlocEncapOctets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of octets of LISP packets that matched this
            EID prefix and were encapsulated using this RLOC address.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of RLOC of cache
            being removed and replaced, which can be detected by
            observing the value of lispMapCacheLocatorRlocTimeStamp."
        ::= { lispMapCacheLocatorEntry 17 }

    lispMapCacheLocatorRlocEncapPackets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
```

```
            "The number of LISP packets that matched this EID prefix
            and were encapsulated using this RLOC address.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of RLOC of cache
            being removed and replaced, which can be detected by
            observing the value of lispMapCacheLocatorRlocTimeStamp."
        ::= { lispMapCacheLocatorEntry 18 }


    lispConfiguredLocatorTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF LispConfiguredLocatorEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This table represents the set of routing locators
            configured on this device. Note that the Proxy-ITR
            configured addresses are treated as routing locators
            and therefore can be part of this table."
        REFERENCE
            "RFC6830, Section 6.3."
        ::= { lispObjects 8 }

    lispConfiguredLocatorEntry OBJECT-TYPE
        SYNTAX     LispConfiguredLocatorEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "An entry (conceptual row) in the
            lispConfiguredLocatorTable."
        INDEX  { lispConfiguredLocatorRlocLength,
                 lispConfiguredLocatorRloc }
        ::= { lispConfiguredLocatorTable 1 }

    LispConfiguredLocatorEntry ::= SEQUENCE {
        lispConfiguredLocatorRlocLength        Integer32,
        lispConfiguredLocatorRloc              LispAddressType,
        lispConfiguredLocatorRlocState         INTEGER,
        lispConfiguredLocatorRlocLocal         INTEGER,
        lispConfiguredLocatorRlocTimeStamp     TimeStamp,
        lispConfiguredLocatorRlocDecapOctets   Counter64,
        lispConfiguredLocatorRlocDecapPackets  Counter64,
        lispConfiguredLocatorRlocEncapOctets   Counter64,
        lispConfiguredLocatorRlocEncapPackets  Counter64
    }

    lispConfiguredLocatorRlocLength OBJECT-TYPE
```

```
        SYNTAX     Integer32 (5..39)
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This object is used to get the octet-length of
            lispConfiguredLocatorRloc."
        ::= { lispConfiguredLocatorEntry 1 }

    lispConfiguredLocatorRloc OBJECT-TYPE
        SYNTAX     LispAddressType
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This object is a RLOC address configured on this device.
            It can be an RLOC that is local to this device or can be an
            RLOC which belongs to another ETR within the same site.
            Proxy-ITR address is treated as an RLOC."
        ::= { lispConfiguredLocatorEntry 2 }

    lispConfiguredLocatorRlocState OBJECT-TYPE
        SYNTAX     INTEGER {
                      up (1),
                      down (2),
                      unreachable (3)
                   }
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The state of this RLOC as per this device. (1 = RLOC is up;
            2 = RLOC is down; 3 = RLOC is unreachable)."
        ::= { lispConfiguredLocatorEntry 3 }

    lispConfiguredLocatorRlocLocal OBJECT-TYPE
        SYNTAX     INTEGER {
                      siteself (1),
                      sitelocal (2)
                   }
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates whether the RLOC is local to this device (or
            remote, meaning local to another device in the same LISP
            site). (1 = RLOC is an address on this device; 2 = RLOC is
            an address on another device)."
        ::= { lispConfiguredLocatorEntry 4 }

    lispConfiguredLocatorRlocTimeStamp OBJECT-TYPE
        SYNTAX     TimeStamp
```

```
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The value of sysUpTime at which the RLOC was configured on
            this device.

            If this information was present at the most recent
            re-initialization of the local management subsystem, then
            this object contains a zero value."
        DEFVAL { 0 }
        ::= { lispConfiguredLocatorEntry 5 }

    lispConfiguredLocatorRlocDecapOctets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of octets of LISP packets that were addressed to
            this RLOC and were decapsulated.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of configured
            RLOC being removed and replaced, which can be detected by
            observing the value of lispConfiguredLocatorRlocTimeStamp."
        ::= { lispConfiguredLocatorEntry 6 }

    lispConfiguredLocatorRlocDecapPackets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of LISP packets that were addressed to this RLOC
            and were decapsulated.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of configured
            RLOC being removed and replaced, which can be detected by
            observing the value of lispConfiguredLocatorRlocTimeStamp."
        ::= { lispConfiguredLocatorEntry 7 }

    lispConfiguredLocatorRlocEncapOctets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of octets of LISP packets that were encapsulated
```

```
            by this device using this RLOC address as the source.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of configured
            RLOC being removed and replaced, which can be detected by
            observing the value of lispConfiguredLocatorRlocTimeStamp."
        ::= { lispConfiguredLocatorEntry 8 }

    lispConfiguredLocatorRlocEncapPackets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of LISP packets that were encapsulated by this
            device using this RLOC address as the source.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of configured
            RLOC being removed and replaced, which can be detected by
            observing the value of lispConfiguredLocatorRlocTimeStamp."
        ::= { lispConfiguredLocatorEntry 9 }



    lispEidRegistrationTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF LispEidRegistrationEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This table provides the properties of each LISP EID prefix
            that is registered with this device when configured to be
            a Map-Server."
        REFERENCE
            "RFC6833, Section 4.0."
        ::= { lispObjects 9 }

    lispEidRegistrationEntry OBJECT-TYPE
        SYNTAX     LispEidRegistrationEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "An entry (conceptual row) in the lispEidRegistrationTable."
        INDEX      { lispEidRegistrationEidLength,
                     lispEidRegistrationEid }
        ::= { lispEidRegistrationTable 1 }
```

```
    LispEidRegistrationEntry ::= SEQUENCE {
        lispEidRegistrationEidLength                 Integer32,
        lispEidRegistrationEid                       LispAddressType,
        lispEidRegistrationSiteName                  OCTET STRING,
        lispEidRegistrationSiteDescription           OCTET STRING,
        lispEidRegistrationIsRegistered              TruthValue,
        lispEidRegistrationFirstTimeStamp            TimeStamp,
        lispEidRegistrationLastTimeStamp             TimeStamp,
        lispEidRegistrationLastRegisterSenderLength  Integer32,
        lispEidRegistrationLastRegisterSender        LispAddressType,
        lispEidRegistrationAuthenticationErrors      Counter64,
        lispEidRegistrationRlocsMismatch             Counter64
    }

    lispEidRegistrationEidLength OBJECT-TYPE
        SYNTAX     Integer32 (5..39)
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This object is used to get the octet-length of
            lispEidRegistrationEid."
        ::= { lispEidRegistrationEntry 1 }

    lispEidRegistrationEid OBJECT-TYPE
        SYNTAX     LispAddressType
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "The EID prefix that is being registered."
         ::= { lispEidRegistrationEntry 2 }

    lispEidRegistrationSiteName OBJECT-TYPE
        SYNTAX     OCTET STRING (SIZE(0..63))
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Site name used by a Map-Server to distinguish different
            LISP sites that are registering with it."
        ::= { lispEidRegistrationEntry 3 }

    lispEidRegistrationSiteDescription OBJECT-TYPE
        SYNTAX     OCTET STRING (SIZE(0..255))
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Description for a site name used by a Map-Server. The EID
            prefix that is being registered belongs to this site."
        ::= { lispEidRegistrationEntry 4 }
```

```
    lispEidRegistrationIsRegistered OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates the registration status of the given EID prefix.
            If this object is true, then it means the EID prefix is
            registered.

            The value false implies the EID prefix is not registered
            with the Map Server. There are multiple scenarios when this
            could happen like authentication failures, routing problems,
            misconfigs to name a few."
        ::= { lispEidRegistrationEntry 5 }

    lispEidRegistrationFirstTimeStamp OBJECT-TYPE
        SYNTAX     TimeStamp
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The value of sysUpTime at which the first valid register
            message for the EID Prefix information represented by this
            entry was received by this device.

            If this information was present at the most recent
            re-initialization of the local management subsystem, then
            this object contains a zero value."
        DEFVAL { 0 }
        ::= { lispEidRegistrationEntry 6 }

    lispEidRegistrationLastTimeStamp OBJECT-TYPE
        SYNTAX     TimeStamp
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The value of sysUpTime at which the last valid register
            message for the EID Prefix information represented by this
            entry was received by this device.

            If this information was present at the most recent
            re-initialization of the local management subsystem, then
            this object contains a zero value."
        DEFVAL { 0 }
        ::= { lispEidRegistrationEntry 7 }

    lispEidRegistrationLastRegisterSenderLength OBJECT-TYPE
        SYNTAX     Integer32 (5..39)
        MAX-ACCESS read-only
```

```
        STATUS     current
        DESCRIPTION
            "This object is used to get the octet-length of
            lispEidRegistrationLastRegisterSender, the next
            object."
        ::= { lispEidRegistrationEntry 8 }

    lispEidRegistrationLastRegisterSender OBJECT-TYPE
        SYNTAX     LispAddressType
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Source address of the last valid register message for the
            given EID prefix that was received by this device."
        ::= { lispEidRegistrationEntry 9 }

    lispEidRegistrationAuthenticationErrors OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Count of total authentication errors of map-registers
            received for the given EID prefix.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of site config
            changes, which can be detected by observing the value of
            lispEidRegistrationFirstTimeStamp."
        ::= { lispEidRegistrationEntry 10 }

    lispEidRegistrationRlocsMismatch OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Count of total map-registers received that had at least one
            RLOC that was not in the allowed list of RLOCs for the given
            EID prefix.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of site config
            changes, which can be detected by observing the value of
            lispEidRegistrationFirstTimeStamp."
        ::= { lispEidRegistrationEntry 11 }
```

```
    lispEidRegistrationEtrTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF LispEidRegistrationEtrEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This table provides the properties of ETRs that register
            the given EID prefix with this device when configured to
            be a Map-Server."
        REFERENCE
            "RFC6830, Section 6.1."
        ::= { lispObjects 10 }

    lispEidRegistrationEtrEntry OBJECT-TYPE
        SYNTAX     LispEidRegistrationEtrEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "An entry (conceptual row) in the
            lispEidRegistrationEtrTable."
        INDEX      { lispEidRegistrationEidLength,
                     lispEidRegistrationEid,
                     lispEidRegistrationEtrSenderLength,
                     lispEidRegistrationEtrSender }
        ::= { lispEidRegistrationEtrTable 1 }

    LispEidRegistrationEtrEntry ::= SEQUENCE {
        lispEidRegistrationEtrSenderLength          Integer32,
        lispEidRegistrationEtrSender                LispAddressType,
        lispEidRegistrationEtrLastTimeStamp         TimeStamp,
        lispEidRegistrationEtrTtl                   Unsigned32,
        lispEidRegistrationEtrProxyReply            TruthValue,
        lispEidRegistrationEtrWantsMapNotify        TruthValue
    }

    lispEidRegistrationEtrSenderLength OBJECT-TYPE
        SYNTAX     Integer32 (5..39)
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This object is used to get the octet-length of
            lispEidRegistrationEtrSender."
        ::= { lispEidRegistrationEtrEntry 1 }

    lispEidRegistrationEtrSender OBJECT-TYPE
        SYNTAX     LispAddressType
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
```

```
            "Source address of the ETR that is sending valid register
            messages for this EID prefix to this device."
        ::= { lispEidRegistrationEtrEntry 2 }

    lispEidRegistrationEtrLastTimeStamp OBJECT-TYPE
        SYNTAX     TimeStamp
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The value of sysUpTime at which the last valid register
            message from this ETR for the EID Prefix information
            represented by this entry was received by this device.

            If this information was present at the most recent
            re-initialization of the local management subsystem,
            then this object contains a zero value."
        DEFVAL { 0 }
        ::= { lispEidRegistrationEtrEntry 3 }

    lispEidRegistrationEtrTtl OBJECT-TYPE
        SYNTAX     Unsigned32
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The Record TTL of the registering ETR device for this
            EID prefix."
        ::= { lispEidRegistrationEtrEntry 4 }

    lispEidRegistrationEtrProxyReply OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates proxy-replying status of the registering ETR for
            this EID prefix. If this object is true, then it means the
            Map-Server can proxy-reply."
        ::= { lispEidRegistrationEtrEntry 5 }

    lispEidRegistrationEtrWantsMapNotify OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates whether the EID prefix wants Map-Notifications.
            If this object is true, then it means the EID prefix wants
            Map-Notifications."
        ::= { lispEidRegistrationEtrEntry 6 }
```

```
   lispEidRegistrationLocatorTable OBJECT-TYPE
       SYNTAX     SEQUENCE OF LispEidRegistrationLocatorEntry
       MAX-ACCESS not-accessible
       STATUS     current
       DESCRIPTION
           "This table provides the properties of all locators per
           LISP site that are served by this device when configured
           to be a Map-Server."
       REFERENCE
           "RFC6830, Section 6.1."
       ::= { lispObjects 11 }

   lispEidRegistrationLocatorEntry OBJECT-TYPE
       SYNTAX     LispEidRegistrationLocatorEntry
       MAX-ACCESS not-accessible
       STATUS     current
       DESCRIPTION
           "An entry (conceptual row) in the
           lispEidRegistrationLocatorTable."
       INDEX      { lispEidRegistrationEidLength,
                    lispEidRegistrationEid,
                    lispEidRegistrationEtrSenderLength,
                    lispEidRegistrationEtrSender,
                    lispEidRegistrationLocatorRlocLength,
                    lispEidRegistrationLocatorRloc }
       ::= { lispEidRegistrationLocatorTable 1 }

   LispEidRegistrationLocatorEntry ::= SEQUENCE {
       lispEidRegistrationLocatorRlocLength         Integer32,
       lispEidRegistrationLocatorRloc               LispAddressType,
       lispEidRegistrationLocatorRlocState          INTEGER,
       lispEidRegistrationLocatorIsLocal            TruthValue,
       lispEidRegistrationLocatorPriority           Integer32,
       lispEidRegistrationLocatorWeight             Integer32,
       lispEidRegistrationLocatorMPriority          Integer32,
       lispEidRegistrationLocatorMWeight            Integer32
   }

   lispEidRegistrationLocatorRlocLength OBJECT-TYPE
       SYNTAX     Integer32 (5..39)
       MAX-ACCESS not-accessible
       STATUS     current
       DESCRIPTION
           "This object is used to get the octet-length of
           lispEidRegistrationLocatorRloc."
       ::= { lispEidRegistrationLocatorEntry 1 }

   lispEidRegistrationLocatorRloc OBJECT-TYPE
```

```
        SYNTAX     LispAddressType
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "The locator of the given EID prefix being registered by the
            given ETR with this device."
        ::= { lispEidRegistrationLocatorEntry 2 }

    lispEidRegistrationLocatorRlocState OBJECT-TYPE
        SYNTAX     INTEGER {
                      up (1),
                      down (2)
                   }
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The cached state of this RLOC received in map-register from
            the ETR by the device, in the capacity of a Map-Server.
            Value 1 refers to up, value 2 refers to down."
        ::= { lispEidRegistrationLocatorEntry 3 }

    lispEidRegistrationLocatorIsLocal OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates if the given locator is local to the registering
            ETR. If this object is true,  it means the locator is local."
        ::= { lispEidRegistrationLocatorEntry 4 }

    lispEidRegistrationLocatorPriority OBJECT-TYPE
        SYNTAX     Integer32 (0..255)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The unicast priority of the RLOC for this EID prefix in the
            register message sent by the given ETR."
        ::= { lispEidRegistrationLocatorEntry 5 }

    lispEidRegistrationLocatorWeight OBJECT-TYPE
        SYNTAX     Integer32 (0..100)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The unicast weight of the RLOC for this EID prefix in the
            register message sent by the given ETR."
        ::= { lispEidRegistrationLocatorEntry 6 }
```

```
   lispEidRegistrationLocatorMPriority OBJECT-TYPE
       SYNTAX     Integer32 (0..255)
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
           "The multicast priority of the RLOC for this EID prefix in
           the register message sent by the given ETR."
       ::= { lispEidRegistrationLocatorEntry 7 }

   lispEidRegistrationLocatorMWeight OBJECT-TYPE
       SYNTAX     Integer32 (0..100)
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
           "The multicast weight of the RLOC for this EID prefix in the
           register message sent by the given ETR."
       ::= { lispEidRegistrationLocatorEntry 8 }


      lispUseMapServerTable OBJECT-TYPE
          SYNTAX     SEQUENCE OF LispUseMapServerEntry
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
              "This table provides the properties of the map-server(s)
              with which this device is configured to register."
          REFERENCE
              "RFC6833, Section 4.3."
          ::= { lispObjects 12 }

      lispUseMapServerEntry OBJECT-TYPE
          SYNTAX     LispUseMapServerEntry
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
              "An entry (conceptual row) in the lispUseMapServerTable."
          INDEX      { lispUseMapServerAddressLength,
                       lispUseMapServerAddress }
          ::= { lispUseMapServerTable 1 }

      LispUseMapServerEntry ::= SEQUENCE {
          lispUseMapServerAddressLength Integer32,
          lispUseMapServerAddress       LispAddressType,
          lispUseMapServerState         INTEGER
      }

      lispUseMapServerAddressLength OBJECT-TYPE
          SYNTAX     Integer32 (5..39)
```

```
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This object is used to get the octet-length of
            lispUseMapServerAddress."
        ::= { lispUseMapServerEntry 1 }

    lispUseMapServerAddress OBJECT-TYPE
        SYNTAX     LispAddressType
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "Address of Map-Server configured on this device."
        ::= { lispUseMapServerEntry 2 }

    lispUseMapServerState OBJECT-TYPE
        SYNTAX     INTEGER {
                      up (1),
                      down (2),
                      unreachable (3)
                   }
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "State of this Map-Server configured on this device
            (1 = Map-Server is up; 2 = Map-Server is down)."
        ::= { lispUseMapServerEntry 3 }



    lispUseMapResolverTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF LispUseMapResolverEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This table provides the properties of the map-resolver(s)
            this device is configured to use."
        REFERENCE
            "RFC6833, Section 4.4."
        ::= { lispObjects 13 }

    lispUseMapResolverEntry OBJECT-TYPE
        SYNTAX     LispUseMapResolverEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "An entry (conceptual row) in the
            lispUseMapResolverTable."
```

```
        INDEX      { lispUseMapResolverAddressLength,
                     lispUseMapResolverAddress }
        ::= { lispUseMapResolverTable 1 }

    LispUseMapResolverEntry ::= SEQUENCE {
        lispUseMapResolverAddressLength   Integer32,
        lispUseMapResolverAddress         LispAddressType,
        lispUseMapResolverState           INTEGER
    }

    lispUseMapResolverAddressLength OBJECT-TYPE
        SYNTAX     Integer32 (5..39)
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This object is used to get the octet-length of
            lispUseMapResolverAddress."
        ::= { lispUseMapResolverEntry 1 }

    lispUseMapResolverAddress OBJECT-TYPE
        SYNTAX     LispAddressType
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "Address of map-resolver configured on this device."
        ::= { lispUseMapResolverEntry 2 }

    lispUseMapResolverState OBJECT-TYPE
        SYNTAX     INTEGER {
                      up (1),
                      down (2)
                   }
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "State of this Map-Resolver configured on this device
            (1 = Map-Resolver is up; 2 = Map-Resolver is down)."
        ::= { lispUseMapResolverEntry 3 }



    lispUseProxyEtrTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF LispUseProxyEtrEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This table provides the properties of all Proxy ETRs that
            this device is configured to use."
```

```
        REFERENCE
            "RFC6830, Section 6.0."
        ::= { lispObjects 14 }

    lispUseProxyEtrEntry OBJECT-TYPE
        SYNTAX     LispUseProxyEtrEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "An entry (conceptual row) in the
            lispUseProxyEtrTable."
        INDEX      { lispUseProxyEtrAddressLength,
                     lispUseProxyEtrAddress }
        ::= { lispUseProxyEtrTable 1 }

    LispUseProxyEtrEntry ::= SEQUENCE {
        lispUseProxyEtrAddressLength        Integer32,
        lispUseProxyEtrAddress              LispAddressType,
        lispUseProxyEtrPriority             Integer32,
        lispUseProxyEtrWeight               Integer32,
        lispUseProxyEtrMPriority            Integer32,
        lispUseProxyEtrMWeight              Integer32,
        lispUseProxyEtrState                INTEGER
    }

    lispUseProxyEtrAddressLength OBJECT-TYPE
        SYNTAX     Integer32 (5..39)
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This object is used to get the octet-length of
            lispUseProxyEtrAddress."
        ::= { lispUseProxyEtrEntry 1 }

    lispUseProxyEtrAddress OBJECT-TYPE
        SYNTAX     LispAddressType
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "Address of Proxy ETR configured on this device."
        ::= { lispUseProxyEtrEntry 2 }

    lispUseProxyEtrPriority OBJECT-TYPE
        SYNTAX     Integer32 (0..255)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The unicast priority of the PETR locator."
```

```
        ::= { lispUseProxyEtrEntry 3 }

    lispUseProxyEtrWeight OBJECT-TYPE
        SYNTAX     Integer32 (0..100)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The unicast weight of the PETR locator."
        ::= { lispUseProxyEtrEntry 4 }

    lispUseProxyEtrMPriority OBJECT-TYPE
        SYNTAX     Integer32 (0..255)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The multicast priority of the PETR locator."
        ::= { lispUseProxyEtrEntry 5 }

    lispUseProxyEtrMWeight OBJECT-TYPE
        SYNTAX     Integer32 (0..100)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The multicast weight of the PETR locator."
        ::= { lispUseProxyEtrEntry 6 }

    lispUseProxyEtrState OBJECT-TYPE
        SYNTAX     INTEGER {
                      down (0),
                      up (1)
                   }
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "State of this Proxy ETR configured on this device
            (0 = Proxy ETR is down; 1 = Proxy ETR is up)."
        ::= { lispUseProxyEtrEntry 7 }
```

```
      --
      -- Conformance Information
      --

      lispCompliances OBJECT IDENTIFIER ::= { lispConformance 1 }
      lispGroups      OBJECT IDENTIFIER ::= { lispConformance 2 }


   --
   -- Compliance Statements
   --

   lispMIBComplianceEtr MODULE-COMPLIANCE
       STATUS  current
       DESCRIPTION
               "The compliance statement for LISP ETRs. It conveys
               information if device supports ETR feature, and relevant
               state associated with that feature."
       MODULE  -- this module
       MANDATORY-GROUPS { lispMIBEtrGroup }

         GROUP   lispMIBItrGroup
         DESCRIPTION
             "This group is optional."

         GROUP   lispMIBPetrGroup
         DESCRIPTION
             "This group is optional."

         GROUP   lispMIBPitrGroup
         DESCRIPTION
             "This group is optional."

         GROUP   lispMIBMapServerGroup
         DESCRIPTION
             "This group is optional."

         GROUP   lispMIBMapResolverGroup
         DESCRIPTION
             "This group is optional."

         GROUP   lispMIBEtrExtendedGroup
         DESCRIPTION
             "This group is optional."

         GROUP   lispMIBItrExtendedGroup
         DESCRIPTION
             "This group is optional."
```

```
        GROUP   lispMIBMapServerExtendedGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBTuningParametersGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBEncapStatisticsGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBDecapStatisticsGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBDiagnosticsGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBVrfGroup
        DESCRIPTION
            "This group is optional."

      ::= { lispCompliances 1 }

  lispMIBComplianceItr MODULE-COMPLIANCE
      STATUS  current
      DESCRIPTION
              "The compliance statement for LISP ITRs. It conveys
              information if device supports ITR feature, and any
              state associated with that feature."
      MODULE  -- this module
      MANDATORY-GROUPS { lispMIBItrGroup }

        GROUP   lispMIBEtrGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBPetrGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBPitrGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBMapServerGroup
```

```
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBMapResolverGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBEtrExtendedGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBItrExtendedGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBMapServerExtendedGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBTuningParametersGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBEncapStatisticsGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBDecapStatisticsGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBDiagnosticsGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBVrfGroup
        DESCRIPTION
            "This group is optional."

      ::= { lispCompliances 2 }

  lispMIBCompliancePetr MODULE-COMPLIANCE
      STATUS  current
      DESCRIPTION
              "The compliance statement for LISP Proxy-ETRs. It conveys
              information if given device supports Proxy-ETR feature,
              and relevant state associated with that feature."
      MODULE  -- this module
```

```
        MANDATORY-GROUPS { lispMIBPetrGroup }

          GROUP   lispMIBEtrGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBItrGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBPitrGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBMapServerGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBMapResolverGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBEtrExtendedGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBItrExtendedGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBMapServerExtendedGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBTuningParametersGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBEncapStatisticsGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBDecapStatisticsGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBDiagnosticsGroup
          DESCRIPTION
```

```
            "This group is optional."

        GROUP   lispMIBVrfGroup
        DESCRIPTION
            "This group is optional."

      ::= { lispCompliances 3 }

   lispMIBCompliancePitr MODULE-COMPLIANCE
      STATUS  current
      DESCRIPTION
              "The compliance statement for LISP Proxy-ITRs. It conveys
              information if device supports Proxy-ITR feature, and
              relevant state associated with that feature."
      MODULE  -- this module
      MANDATORY-GROUPS { lispMIBPitrGroup }

        GROUP   lispMIBEtrGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBItrGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBPetrGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBMapServerGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBMapResolverGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBEtrExtendedGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBItrExtendedGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBMapServerExtendedGroup
        DESCRIPTION
            "This group is optional."
```

```
        GROUP   lispMIBTuningParametersGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBEncapStatisticsGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBDecapStatisticsGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBDiagnosticsGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBVrfGroup
        DESCRIPTION
            "This group is optional."

    ::= { lispCompliances 4 }

  lispMIBComplianceMapServer MODULE-COMPLIANCE
      STATUS  current
      DESCRIPTION
              "The compliance statement for LISP Map Servers. It
              conveys information if device supports Map Server
              feature, and relevant state associated with that
              feature."
      MODULE  -- this module
      MANDATORY-GROUPS { lispMIBMapServerGroup }

        GROUP   lispMIBEtrGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBItrGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBPetrGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBPitrGroup
        DESCRIPTION
            "This group is optional."
```

```
        GROUP   lispMIBMapResolverGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBEtrExtendedGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBItrExtendedGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBMapServerExtendedGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBTuningParametersGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBEncapStatisticsGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBDecapStatisticsGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBDiagnosticsGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBVrfGroup
        DESCRIPTION
            "This group is optional."

      ::= { lispCompliances 5 }

   lispMIBComplianceMapResolver MODULE-COMPLIANCE
       STATUS  current
       DESCRIPTION
               "The compliance statement for LISP Map Resolvers. It
               conveys information if device supports Map Server
               feature, and relevant state associated with that
               feature."
       MODULE  -- this module
       MANDATORY-GROUPS { lispMIBMapResolverGroup }
```

```
        GROUP   lispMIBEtrGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBItrGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBPetrGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBPitrGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBMapServerGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBEtrExtendedGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBItrExtendedGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBMapServerExtendedGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBTuningParametersGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBEncapStatisticsGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBDecapStatisticsGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBDiagnosticsGroup
        DESCRIPTION
            "This group is optional."
```

```
        GROUP   lispMIBVrfGroup
        DESCRIPTION
            "This group is optional."

    ::= { lispCompliances 6 }


--
-- Units of Conformance
--

lispMIBEtrGroup OBJECT-GROUP
    OBJECTS { lispFeaturesEtrEnabled,
              lispMappingDatabaseLsb,
              lispMappingDatabaseLocatorRlocPriority,
              lispMappingDatabaseLocatorRlocWeight,
              lispMappingDatabaseLocatorRlocMPriority,
              lispMappingDatabaseLocatorRlocMWeight,
              lispMappingDatabaseLocatorRlocState,
              lispMappingDatabaseLocatorRlocLocal,
              lispConfiguredLocatorRlocState,
              lispConfiguredLocatorRlocLocal,
              lispUseMapServerState
            }
    STATUS  current
    DESCRIPTION
            "A collection of objects to support reporting of basic
             LISP ETR parameters."
    ::= { lispGroups 1 }

lispMIBItrGroup OBJECT-GROUP
    OBJECTS { lispFeaturesItrEnabled,
              lispFeaturesMapCacheSize,
              lispMappingDatabaseLsb,
              lispMapCacheLocatorRlocPriority,
              lispMapCacheLocatorRlocWeight,
              lispMapCacheLocatorRlocMPriority,
              lispMapCacheLocatorRlocMWeight,
              lispMapCacheLocatorRlocState,
              lispMapCacheEidTimeStamp,
              lispMapCacheEidExpiryTime,
              lispUseMapResolverState,
              lispUseProxyEtrPriority,
              lispUseProxyEtrWeight,
              lispUseProxyEtrMPriority,
              lispUseProxyEtrMWeight,
              lispUseProxyEtrState
            }
```

```
        STATUS  current
        DESCRIPTION
                "A collection of objects to support reporting of basic
                 LISP ITR parameters."
        ::= { lispGroups 2 }

    lispMIBPetrGroup OBJECT-GROUP
        OBJECTS { lispFeaturesProxyEtrEnabled
                }
        STATUS  current
        DESCRIPTION
                "A collection of objects to support reporting of basic
                 LISP Proxy-ETR parameters."
        ::= { lispGroups 3 }

    lispMIBPitrGroup OBJECT-GROUP
        OBJECTS { lispFeaturesProxyItrEnabled,
                  lispConfiguredLocatorRlocState,
                  lispConfiguredLocatorRlocLocal
                }

        STATUS  current
        DESCRIPTION
                "A collection of objects to support reporting of basic
                 LISP Proxy-ITR parameters."
        ::= { lispGroups 4 }

    lispMIBMapServerGroup OBJECT-GROUP
        OBJECTS { lispFeaturesMapServerEnabled,
                  lispEidRegistrationIsRegistered,
                  lispEidRegistrationLocatorRlocState
                }
        STATUS  current
        DESCRIPTION
                "A collection of objects to support reporting of basic
                 LISP Map Server parameters."
        ::= { lispGroups 5 }

    lispMIBMapResolverGroup OBJECT-GROUP
        OBJECTS { lispFeaturesMapResolverEnabled
                }
        STATUS  current
        DESCRIPTION
                "A collection of objects to support reporting of basic
                 LISP Map Resolver parameters."
        ::= { lispGroups 6 }

    lispMIBEtrExtendedGroup OBJECT-GROUP
```

```
        OBJECTS { lispFeaturesRlocProbeEnabled,
                  lispFeaturesEtrAcceptMapDataEnabled,
                  lispFeaturesEtrAcceptMapDataVerifyEnabled,
                  lispMappingDatabaseEidPartitioned
                }
        STATUS  current
        DESCRIPTION
                "A collection of objects to support reporting of
                 LISP features and properties on ETRs."
        ::= { lispGroups 7 }

    lispMIBItrExtendedGroup OBJECT-GROUP
        OBJECTS { lispFeaturesRlocProbeEnabled,
                  lispMapCacheEidState,
                  lispMapCacheEidAuthoritative,
                  lispMapCacheLocatorRlocTimeStamp,
                  lispMapCacheLocatorRlocLastPriorityChange,
                  lispMapCacheLocatorRlocLastWeightChange,
                  lispMapCacheLocatorRlocLastMPriorityChange,
                  lispMapCacheLocatorRlocLastMWeightChange,
                  lispMapCacheLocatorRlocLastStateChange,
                  lispMapCacheLocatorRlocRtt
                }
        STATUS  current
        DESCRIPTION
                "A collection of objects to support reporting of
                 LISP features and properties on ITRs."
        ::= { lispGroups 8 }

    lispMIBMapServerExtendedGroup OBJECT-GROUP
        OBJECTS { lispEidRegistrationSiteName,
                  lispEidRegistrationSiteDescription,
                  lispEidRegistrationIsRegistered,
                  lispEidRegistrationFirstTimeStamp,
                  lispEidRegistrationLastTimeStamp,
                  lispEidRegistrationLastRegisterSenderLength,
                  lispEidRegistrationLastRegisterSender,
                  lispEidRegistrationEtrLastTimeStamp,
                  lispEidRegistrationEtrTtl,
                  lispEidRegistrationEtrProxyReply,
                  lispEidRegistrationEtrWantsMapNotify,
                  lispEidRegistrationLocatorIsLocal,
                  lispEidRegistrationLocatorPriority,
                  lispEidRegistrationLocatorWeight,
                  lispEidRegistrationLocatorMPriority,
                  lispEidRegistrationLocatorMWeight
                }
        STATUS  current
```

```
        DESCRIPTION
                "A collection of objects to support reporting of
                 LISP features and properties on Map Servers
                 related to EID registrations."
        ::= { lispGroups 9 }

    lispMIBTuningParametersGroup OBJECT-GROUP
        OBJECTS { lispFeaturesMapCacheLimit,
                  lispFeaturesEtrMapCacheTtl
                }
        STATUS  current
        DESCRIPTION
                "A collection of objects used to support reporting of
                 parameters used to control LISP behavior and to tune
                 performance."
        ::= { lispGroups 10 }

    lispMIBEncapStatisticsGroup OBJECT-GROUP
        OBJECTS { lispMappingDatabaseTimeStamp,
                  lispMappingDatabaseEncapOctets,
                  lispMappingDatabaseEncapPackets,
                  lispMappingDatabaseLocatorRlocTimeStamp,
                  lispMappingDatabaseLocatorRlocEncapOctets,
                  lispMappingDatabaseLocatorRlocEncapPackets,
                  lispMapCacheEidTimeStamp,
                  lispMapCacheEidEncapOctets,
                  lispMapCacheEidEncapPackets,
                  lispMapCacheLocatorRlocTimeStamp,
                  lispMapCacheLocatorRlocEncapOctets,
                  lispMapCacheLocatorRlocEncapPackets,
                  lispConfiguredLocatorRlocTimeStamp,
                  lispConfiguredLocatorRlocEncapOctets,
                  lispConfiguredLocatorRlocEncapPackets
                }
        STATUS  current
        DESCRIPTION
                "A collection of objects used to support reporting of
                 LISP encapsulation statistics for the device."
        ::= { lispGroups 11 }

    lispMIBDecapStatisticsGroup OBJECT-GROUP
        OBJECTS { lispMappingDatabaseTimeStamp,
                  lispMappingDatabaseDecapOctets,
                  lispMappingDatabaseDecapPackets,
                  lispMappingDatabaseLocatorRlocTimeStamp,
                  lispMappingDatabaseLocatorRlocDecapOctets,
                  lispMappingDatabaseLocatorRlocDecapPackets,
                  lispMapCacheEidTimeStamp,
```

```
                lispMapCacheEidDecapOctets,
                lispMapCacheEidDecapPackets,
                lispMapCacheLocatorRlocTimeStamp,
                lispMapCacheLocatorRlocDecapOctets,
                lispMapCacheLocatorRlocDecapPackets,
                lispConfiguredLocatorRlocTimeStamp,
                lispConfiguredLocatorRlocDecapOctets,
                lispConfiguredLocatorRlocDecapPackets
              }
    STATUS  current
    DESCRIPTION
            "A collection of objects used to support reporting of
             LISP decapsulation statistics for the device."
    ::= { lispGroups 12 }

lispMIBDiagnosticsGroup OBJECT-GROUP
    OBJECTS { lispFeaturesRouterTimeStamp,
              lispGlobalStatsMapRequestsIn,
              lispGlobalStatsMapRequestsOut,
              lispGlobalStatsMapRepliesIn,
              lispGlobalStatsMapRepliesOut,
              lispGlobalStatsMapRegistersIn,
              lispGlobalStatsMapRegistersOut,
              lispEidRegistrationAuthenticationErrors,
              lispEidRegistrationRlocsMismatch
            }
    STATUS  current
    DESCRIPTION
            "A collection of objects used to support reporting of
             additional diagnostics related to the LISP control plane
             state of a LISP device."
    ::= { lispGroups 13 }

lispMIBVrfGroup OBJECT-GROUP
    OBJECTS { lispIidToVrfName
            }
    STATUS  current
    DESCRIPTION
            "A collection of objects used to support reporting of
             VRF-related information on a LISP device."
    ::= { lispGroups 14 }


END
```

8. Relationship to Other MIB Modules

8.1. MIB modules required for IMPORTS

The LISP MIB imports the TEXTUAL-CONVENTION AddressFamilyNumbers from the IANA-ADDRESS-FAMILY-NUMBERS-MIB DEFINITIONS [IANA] http://www.iana.org/assignments/ianaaddressfamilynumbers-mib

The LISP MIB imports mib-2, Unsigned32, Counter64, Integer32, and TimeTicks from SNMPv2-SMI -- [RFC2578].

The LISP MIB imports TruthValue, TEXTUAL-CONVENTION, TimeStamp, and TimeTicks from SNMPv2-TC -- [RFC2579].

The LISP MIB imports MODULE-COMPLIANCE from SNMPv2-TC -- [RFC2580].

The LISP MIB imports MplsL3VpnName from MPLS-L3VPN-STD-MIB -- [RFC4382].

9. Security Considerations

There are no management objects defined in this MIB module that have a MAX-ACCESS clause of read-write and/or read-create. So, if this MIB module is implemented correctly, then there is no risk that an intruder can alter or create any management objects of this MIB module via direct SNMP SET operations.

There are no readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) that are considered sensitive.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

Implementations SHOULD provide the security features described by the SNMPv3 framework (see [RFC3410]), and implementations claiming compliance to the SNMPv3 standard MUST include full support for authentication and privacy via the User-based Security Model (USM) [RFC3414] with the AES cipher algorithm [RFC3826]. Implementations MAY also provide support for the Transport Security Model (TSM) [RFC5591] in combination with a secure transport such as SSH [RFC5592] or TLS/DTLS [RFC6353].

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to

enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

10. IANA Considerations

The MIB module in this document uses the following IANA-assigned OBJECT IDENTIFIER values recorded in the SMI Numbers registry:

Descriptor	OBJECT IDENTIFIER value
lispMIB	{ mib-2 XXX }

This document instructs IANA to allocate a new value in the "SMI Network Management MGMT Codes Internet-standard MIB" subregistry of the "Network Management Parameters" registry, according to the following registration data: Decimal: [TBD by IANA] Name: lispMIB Description: Locator/ID Separation Protocol (LISP) References: [RFC XXXX (this RFC)]

11. References

11.1. Normative References

[IANA] "IANA-ADDRESS-FAMILY-NUMBERS-MIB DEFINITIONS", <http://www.iana.org/assignments/ianaaddressfamilynumbers-mib>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[RFC3414] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management

Protocol (SNMPv3)", STD 62, RFC 3414, December 2002.

[RFC3826] Blumenthal, U., Maino, F., and K. McCloghrie, "The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model", RFC 3826, June 2004.

[RFC4382] Nadeau, T. and H. van der Linde, "MPLS/BGP Layer 3 Virtual Private Network (VPN) Management Information Base", RFC 4382, February 2006.

[RFC5591] Harrington, D. and W. Hardaker, "Transport Security Model for the Simple Network Management Protocol (SNMP)", RFC 5591, June 2009.

[RFC5592] Harrington, D., Salowey, J., and W. Hardaker, "Secure Shell Transport Model for the Simple Network Management Protocol (SNMP)", RFC 5592, June 2009.

[RFC6353] Hardaker, W., "Transport Layer Security (TLS) Transport Model for the Simple Network Management Protocol (SNMP)", RFC 6353, July 2011.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, January 2013.

[RFC6832] Lewis, D., Meyer, D., Farinacci, D., and V. Fuller, "Interworking between Locator/ID Separation Protocol (LISP) and Non-LISP Sites", RFC 6832, January 2013.

[RFC6833] Fuller, V. and D. Farinacci, "Locator/ID Separation Protocol (LISP) Map-Server Interface", RFC 6833, January 2013.

11.2. Informative References

[LCAF] Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format", draft-ietf-lisp-lcaf-02.txt (work in progress), March 2013.

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

Appendix A. Acknowledgments

A thank you is owed to Dino Farinacci for his inputs and review comments on the initial versions of this draft. In addition, the

authors would like to gratefully acknowledge several others who have reviewed and commented on this draft. They include: Darrel Lewis, Isidor Kouvelas, Jesper Skriver, Selina Heimlich, Parna Agrawal, Dan Romascanu, and Luigi Iannone. Special thanks are owed to Brian Haberman, the Internet Area AD, for his very detailed review, Miguel Garcia for reviewing this document as part of the General Area Review Team, and Harrie Hazewinkel for the detailed MIB review comments.

Authors' Addresses

Gregg Schudel
cisco Systems
Tasman Drive
San Jose, CA 95134
USA

EMail: gschudel@cisco.com

Amit Jain
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
USA

EMail: atjain@juniper.net

Victor Moreno
cisco Systems
Tasman Drive
San Jose, CA 95134
USA

EMail: vimoreno@cisco.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: October 10, 2011

L. Jakab
A. Cabellos-Aparicio
F. Coras
J. Domingo-Pascual
Technical University of Catalonia
D. Lewis
Cisco Systems
April 8, 2011

LISP Network Element Deployment Considerations
draft-jakab-lisp-deployment-03.txt

Abstract

This document discusses the different scenarios for the deployment of the new network elements introduced by the Locator/Identifier Separation Protocol (LISP).

Status of this Memo

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

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

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

This Internet-Draft will expire on October 10, 2011.

Copyright Notice

Table of Contents

1. Introduction

The Locator/Identifier Separation Protocol (LISP) addresses the scaling issues of the global Internet routing system by separating the current addressing scheme into Endpoint IDentifiers (EIDs) and Routing LOCators (RLOCs). The main protocol specification [I-D.ietf-lisp] describes how the separation is achieved, which new network elements are introduced, and details the packet formats for the data and control planes.

While the boundary between the core and edge is not strictly defined, one widely accepted definition places it at the border routers of stub autonomous systems, which may carry a partial or complete default-free zone (DFZ) routing table. The initial design of LISP took this location as a baseline for protocol development. However, the applications of LISP go beyond of just decreasing the size of the DFZ routing table, and include improved multihoming and ingress traffic engineering (TE) support for edge networks, and even individual hosts. Throughout the draft we will use the term LISP site to refer to these networks/hosts behind a LISP Tunnel Router. We formally define it as:

LISP site: A single host or a set of network elements in an edge network under the administrative control of a single organization, delimited from other networks by LISP Tunnel Router(s).

Since LISP is a protocol which can be used for different purposes, it is important to identify possible deployment scenarios and the additional requirements they may impose on the protocol specification and other protocols. The main specification [I-D.ietf-lisp] mentions positioning of tunnel routers, but without an in-depth discussion. This document fills that gap, by exploring the most common cases. While the theoretical combinations of device placements are quite numerous, the more practical scenarios are given preference in the following.

Additionally, this documents is intended as a guide for the operational community for LISP deployments in their networks. It is expected to evolve as LISP deployment progresses, and the described scenarios are better understood or new scenarios are discovered.

Each subsection considers an element type, discussing the impact of deployment scenarios on the protocol specification. For definition of terms, please refer to the appropriate documents (as cited in the respective sections).

Comments and discussions about this memo should be directed to the LISP working group mailing list: lisp@ietf.org.

2. Tunnel Routers

LISP is a map-and-encap protocol, with the main goal of improving global routing scalability. To achieve its goal, it introduces several new network elements, each performing specific functions necessary to separate the edge from the core. The device that is the gateway between the edge and the core is called Tunnel Router (xTR), performing one or both of two separate functions:

1. Encapsulating packets originating from an end host to be transported over intermediary (transit) networks towards the other end-point of the communication

2. Decapsulating packets entering from intermediary (transit) networks, originated at a remote end host.

The first function is performed by an Ingress Tunnel Router (ITR), the second by an Egress Tunnel Router (ETR).

Section 8 of the main LISP specification [I-D.ietf-lisp] has a short discussion of where Tunnel Routers can be deployed and some of the associated advantages and disadvantages. This section adds more detail to the scenarios presented there, and provides additional scenarios as well.

2.1. Customer Edge

LISP was designed with deployment at the core-edge boundary in mind, which can be approximated as the set of DFZ routers belonging to non-transit ASes. For the purposes of this document, we will consider this boundary to be consisting of the routers connecting LISP sites to their upstreams. As such, this is the most common expected scenario for xTRs, and this document considers it the reference location, comparing the other scenarios to this one.

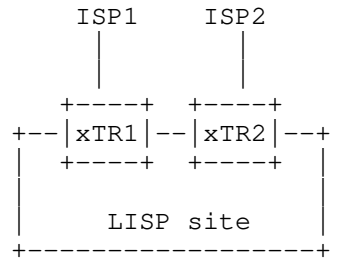


Figure 1: xTRs at the customer edge

From the LISP site perspective the main advantage of this type of

deployment (compared to the one described in the next section) is having direct control over its ingress traffic engineering. This makes it is easy to set up and maintain active/active, active/backup, or more complex TE policies, without involving third parties.

Being under the same administrative control, reachability information of all ETRs is easier to synchronize, because the necessary control traffic can be allowed between the locators of the ETRs. A correct synchronous global view of the reachability status is thus available, and the Loc-Status-Bits can be set correctly in the LISP data header of outgoing packets.

By placing the tunnel router at the edge of the site, existing internal network configuration does not need to be modified. Firewall rules, router configurations and address assignments inside the LISP site remain unchanged. This helps with incremental deployment and allows a quick upgrade path to LISP. For larger sites with many external connections, distributed in geographically diverse PoPs, and complex internal topology, it may however make more sense to both encapsulate and decapsulate as soon as possible, to benefit from the information in the IGP to choose the best path (see Section 2.3 for a discussion of this scenario).

Another thing to consider when placing tunnel routers are MTU issues. Since encapsulating packets increases overhead, the MTU of the end-to-end path may decrease, when encapsulated packets need to travel over segments having close to minimum MTU. Some transit networks are known to provide larger MTU than the typical value of 1500 bytes of popular access technologies used at end hosts (e.g., IEEE 802.3 and 802.11). However, placing the LISP router connecting to such a network at the customer edge could possibly bring up MTU issues, depending on the link type to the provider as opposed to the following scenario.

2.2. Provider Edge

The other location at the core-edge boundary for deploying LISP routers is at the Internet service provider edge. The main incentive for this case is that the customer does not have to upgrade the CE router(s), or change the configuration of any equipment. Encapsulation/decapsulation happens in the provider's network, which may be able to serve several customers with a single device. For large ISPs with many residential/business customers asking for LISP this can lead to important savings, since there is no need to upgrade the software (or hardware, if it's the case) at each client's location. Instead, they can upgrade the software (or hardware) on a few PE routers serving the customers. This scenario is depicted in Figure 2.

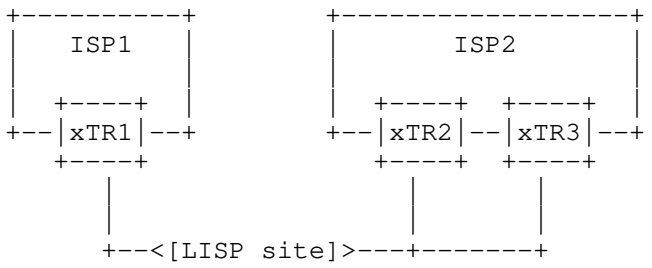


Figure 2: xTR at the PE

While this approach can make transition easy for customers and may be cheaper for providers, the LISP site looses one of the main benefits of LISP: ingress traffic engineering. Since the provider controls the ETRs, additional complexity would be needed to allow customers to modify their mapping entries.

The problem is aggravated when the LISP site is multihomed. Consider the scenario in Figure 2: whenever a change to TE policies is required, the customer contacts both ISP1 and ISP2 to make the necessary changes on the routers (if they provide this possibility). It is however unlikely, that both ISPs will apply changes simultaneously, which may lead to inconsistent state for the mappings of the LISP site (e.g., weights for the same priority don't sum 100). Since the different upstream ISPs are usually competing business entities, the ETRs may even be configured to compete, either to attract all the traffic or to get no traffic. The former will happen if the customer pays per volume, the latter if the connectivity has a fixed price. A solution could be to have the mappings in the Map-Server(s), and have their operator give control over the entries to customer, much like in today's DNS.

Additionally, since xTR1, xTR2, and xTR3 are in different administrative domains, locator reachability information is unlikely to be exchanged among them, making it difficult to set Loc-Status-Bits correctly on encapsulated packets.

Compared to the customer edge scenario, deploying LISP at the provider edge might have the advantage of diminishing potential MTU issues, because the tunnel router is closer to the core, where links typically have higher MTUs than edge network links.

2.3. Split ITR/ETR

In a simple LISP deployment, xTRs are located at the border of the LISP site (see Section 2.1). In this scenario packets are routed inside the domain according to the EID. However, more complex

networks may want to route packets according to the destination RLOC. This would enable them to choose the best egress point.

The LISP specification separates the ITR and ETR functionality and considers that both entities can be deployed in separated network equipment. ITRs can be deployed closer to the host (i.e., access routers). This way packets are encapsulated as soon as possible, and packets exit the network through the best egress point in terms of BGP policy. In turn, ETRs can be deployed at the border routers of the network, and packets are decapsulated as soon as possible. Again, once decapsulated packets are routed according to the EID, and can follow the best path according to internal routing policy.

In the following figure we can see an example. The Source (S) transmits packets using its EID and in this particular case packets are encapsulated at ITR_1. The encapsulated packets are routed inside the domain according to the destination RLOC, and can egress the network through the best point (i.e., closer to the RLOC's AS). On the other hand, inbound packets are received by ETR_1 which decapsulates them. Then packets are routed towards S according to the EID, again following the best path.

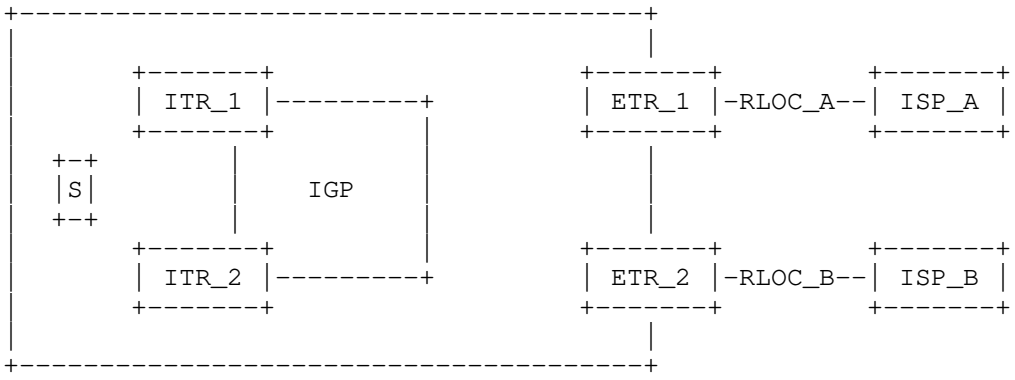


Figure 3: Split ITR/ETR Scenario

This scenario has a set of implications:

- The site must carry at least partial BGP routes in order to choose the best egress point, increasing the complexity of the network. However, this is usually already the case for LISP sites that would benefit from this scenario.

- If the site is multihomed to different ISPs and any of the upstream ISPs is doing uRPF filtering, this scenario may become impractical. ITRs need to determine the exit ETR, for setting the

correct source RLOC in the encapsulation header. This adds complexity and reliability concerns.

- In LISP, ITRs set the reachability bits when encapsulating data packets. Hence, ITRs need a mechanism to be aware of the liveness of ETRs.

- ITRs encapsulate packets and in order to achieve efficient communications, the MTU of the site must be large enough to accommodate this extra header.

- In this scenario, each ITR is serving fewer hosts than in the case when it is deployed at the border of the network. It has been shown that cache hit ratio grows logarithmically with the amount of users [cache]. Taking this into account, when ITRs are deployed closer to the host the effectiveness of the mapping cache may be lower (i.e., the miss ratio is higher). Another consequence of this is that the site will transmit a higher amount of Map-Requests, increasing the load on the distributed mapping database.

2.4. Inter-Service Provider Traffic Engineering

With LISP, two LISP sites can route packets among them and control their ingress TE policies. Typically, LISP is seen as applicable to stub networks, however the LISP protocol can also be applied to transit networks recursively.

Consider the scenario depicted in Figure 4. Packets originating from the LISP site Stub1, client of ISP_A, with destination Stub4, client of ISP_B, are LISP encapsulated at their entry point into the ISP_A's network. The external IP header now has as the source RLOC an IP from ISP_A's address space (R_A1, R_A2, or R_A3) and destination RLOC from ISP_B's address space (R_B1 or R_B2). One or more ASes separate ISP_A from ISP_B. With a single level of LISP encapsulation, Stub4 has control over its ingress traffic. However, ISP_B only has the current tools (such as BGP prefix deaggregation) to control on which of his own upstream or peering links should packets enter. This is either not feasible (if fine-grained per-customer control is required, the very specific prefixes may not be propagated) or increases DFZ table size.

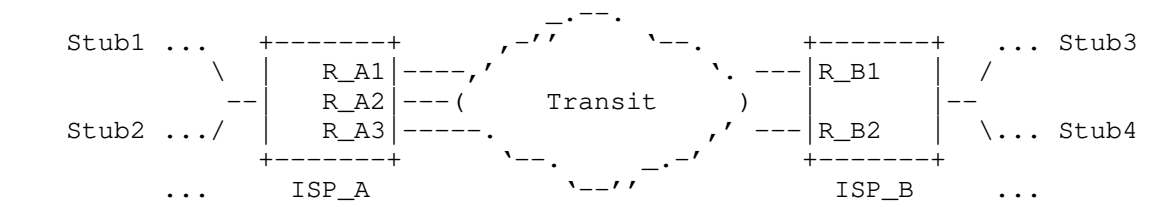


Figure 4: Inter-Service provider TE scenario

A solution for this is to apply LISP recursively. ISP_A and ISP_B may reach a bilateral agreement to deploy their own private mapping system. ISP_A then encapsulates packets destined for the prefixes of ISP_B, which are listed in the shared mapping system. Note that in this case the packet is double-encapsulated. ISP_B's ETR removes the outer, second layer of LISP encapsulation from the incoming packet, and routes it towards the original RLOC, the ETR of Stub4, which does the final decapsulation.

If ISP_A and ISP_B agree to share a private distributed mapping database, both can control their ingress TE without the need of disaggregating prefixes. In this scenario the private database contains RLOC-to-RLOC bindings. The convergence time on the TE policies updates is expected to be fast, since ISPs only have to update/query a mapping to/from the database.

This deployment scenario includes two important recommendations. First, it is intended to be deployed only between two ISPs (ISP_A and ISP_B in Figure 4). If more than two ISPs use this approach, then the xTRs deployed at the participating ISPs must either query multiple mapping systems, or the ISPs must agree on a common shared mapping system. Second, the scenario is only recommended for ISPs providing connectivity to LISP sites, such that source RLOCs of packets to be reencapsulated belong to said ISP. Otherwise the participating ISPs must register prefixes they do not own in the above mentioned private mapping system. Failure to follow these recommendations may lead to operational and security issues when deploying this scenario.

Besides these recommendations, the main disadvantages of this deployment case are:

- Extra LISP header is needed. This increases the packet size and, for efficient communications, it requires that the MTU between both ISPs can accommodate double-encapsulated packets.

- The ISP ITR must encapsulate packets and therefore must know the RLOC-to-RLOC binding. These bindings are stored in a mapping

database and may be cached in the ITR's mapping cache. Cache misses lead to an extra lookup latency, unless NERD [I-D.lear-lisp-nerd] is used for the lookups.

- The operational overhead of maintaining the shared mapping database.

2.5. Tunnel Routers Behind NAT

NAT in this section refers to IPv4 network address and port translation.

2.5.1. ITR

Packets encapsulated by an ITR are just UDP packets from a NAT device's point of view, and they are handled like any UDP packet, there are no additional requirements for LISP data packets.

Map-Requests sent by an ITR, which create the state in the NAT table have a different 5-tuple in the IP header than the Map-Reply generated by the authoritative ETR. Since the source address of this packet is different from the destination address of the request packet, no state will be matched in the NAT table and the packet will be dropped. To avoid this, the NAT device has to do the following:

- Send all UDP packets with source port 4342, regardless of the destination port, to the RLOC of the ITR. The most simple way to achieve this is configuring 1:1 NAT mode from the external RLOC of the NAT device to the ITR's RLOC (Called "DMZ" mode in consumer broadband routers).

- Rewrite the ITR-AFI and "Originating ITR RLOC Address" fields in the payload.

This setup supports a single ITR behind the NAT device.

2.5.2. ETR

An ETR placed behind NAT is reachable from the outside by the Internet-facing locator of the NAT device. It needs to know this locator (and configure a loopback interface with it), so that it can use it in Map-Reply and Map-Register messages. Thus support for dynamic locators for the mapping database is needed in LISP equipment.

Again, only one ETR behind the NAT device is supported.

An implication of the issues described above is that LISP sites with

xTRs can not be behind carrier based NATs, since two different sites would collide on the port forwarding.

2.6. Summary and Feature Matrix

Feature	CE	PE	Split	Rec.
Control of ingress TE	x	-	x	x
No modifications to existing int. network infrastructure	x	x	-	-
Loc-Status-Bits sync	x	-	x	x
MTU/PMTUD issues minimized	-	x	-	x

3. Map-Resolvers and Map-Servers

3.1. Map-Servers

The Map-Server learns EID-to-RLOC mapping entries from an authoritative source and publishes them in the distributed mapping database. These entries are learned through authenticated Map-Register messages sent by authoritative ETRs. Also, upon reception of a Map-Request, the Map-Server verifies that the destination EID matches an EID-prefix for which it is responsible for, and then re-encapsulates and forwards it to a matching ETR. Map-Server functionality is described in detail in [I-D.ietf-lisp-ms].

The Map-Server is provided by a Mapping Service Provider (MSP). A MSP can be any of the following:

- EID registrar. Since the IPv4 address space is nearing exhaustion, IPv4 EIDs will come from already allocated Provider Independent (PI) space. The registrars in this case remain the current five Regional Internet Registries (RIRs). In the case of IPv6, the possibility of reserving a /16 block as EID space is currently under consideration [I-D.meyer-lisp-eid-block]. If granted by IANA, the community will have to determine the body responsible for allocations from this block, and the associated policies. For already allocated IPv6 prefixes the principles from IPv4 should be applied.

- Third parties. Participating in the LISP mapping system is similar to participating in global routing or DNS: as long as there is at least another already participating entity willing to forward the newcomer's traffic, there is no barrier to entry. Still, just like routing and DNS, LISP mappings have the issue of trust, with efforts underway to make the published information verifiable. When these mechanisms will be deployed in the LISP

mapping system, the burden of providing and verifying trust should be kept away from MSPs, which will simply host the secured mappings. This will keep the low barrier of entry to become an MSP for third parties.

In all cases, the MSP configures its Map-Server(s) to publish the prefixes of its clients in the distributed mapping database and start encapsulating and forwarding Map-Requests to the ETRs of the AS. These ETRs register their prefix(es) with the Map-Server(s) through periodic authenticated Map-Register messages. In this context, for some LISP end sites, there is a need for mechanisms to:

- Automatically distribute EID prefix(es) shared keys between the ETRs and the EID-registrar Map-Server.

- Dynamically obtain the address of the Map-Server in the ETR of the AS.

The Map-Server plays a key role in the reachability of the EID-prefixes it is serving. On the one hand it is publishing these prefixes into the distributed mapping database and on the other hand it is encapsulating and forwarding Map-Requests to the authoritative ETRs of these prefixes. ITRs encapsulating towards EIDs under the responsibility of a failed Map-Server will be unable to look up any of their covering prefixes. The only exception are the ITRs that already contain the mappings in their local cache. In this case ITRs can reach ETRs until the entry expires (typically 24 hours). For this reason, redundant Map-Server deployments are desirable. A set of Map-Servers providing high-availability service to the same set of prefixes is called a redundancy group. ETRs are configured to send Map-Register messages to all Map-Servers in the redundancy group. To achieve fail-over (or load-balancing, if desired), current known BGP practices can be used on the LISP+ALT BGP overlay network.

Additionally, if a Map-Server has no reachability for any ETR serving a given EID block, it should not originate that block into the mapping system.

3.2. Map-Resolvers

A Map-Resolver a is a network infrastructure component which accepts LISP encapsulated Map-Requests, typically from an ITR, and finds the appropriate EID-to-RLOC mapping by either consulting its local cache or by consulting the distributed mapping database. Map-Resolver functionality is described in detail in [I-D.ietf-lisp-ms].

Anyone with access to the distributed mapping database can set up a Map-Resolver and provide EID-to-RLOC mapping lookup service. In the

case of the LISP+ALT mapping system, the Map-Resolver needs to become part of the ALT overlay so that it can forward packets to the appropriate Map-Servers. For more detail on how the ALT overlay works, see [I-D.ietf-lisp-alt]

For performance reasons, it is recommended that LISP sites use Map-Resolvers that are topologically close to their ITRs. ISPs supporting LISP will provide this service to their customers, possibly restricting access to their user base. LISP sites not in this position can use open access Map-Resolvers, if available. However, regardless of the availability of open access resolvers, the MSP providing the Map-Server(s) for a LISP site should also make available Map-Resolver(s) for the use of that site.

In medium to large-size ASes, ITRs must be configured with the RLOC of a Map-Resolver, operation which can be done manually. However, in Small Office Home Office (SOHO) scenarios a mechanism for autoconfiguration should be provided.

One solution to avoid manual configuration in LISP sites of any size is the use of anycast RLOCs for Map-Resolvers similar to the DNS root server infrastructure. Since LISP uses UDP encapsulation, the use of anycast would not affect reliability. LISP routers are then shipped with a preconfigured list of well know Map-Resolver RLOCs, which can be edited by the network administrator, if needed.

The use of anycast also helps improving mapping lookup performance. Large MSPs can increase the number and geographical diversity of their Map-Resolver infrastructure, using a single anycasted RLOC. Once LISP deployment is advanced enough, very large content providers may also be interested running this kind of setup, to ensure minimal connection setup latency for those connecting to their network from LISP sites.

While Map-Servers and Map-Resolvers implement different functionalities within the LISP mapping system, they can coexist on the same device. For example, MSPs offering both services, can deploy a single Map-Resolver/Map-Server in each PoP where they have a presence.

4. Proxy Tunnel Routers

4.1. P-ITR

Proxy Ingress Tunnel Routers (P-ITRs) are part of the non-LISP/LISP transition mechanism, allowing non-LISP sites to reach LISP sites. They announce via BGP certain EID prefixes (aggregated, whenever

possible) to attract traffic from non-LISP sites towards EIDs in the covered range. They do the mapping system lookup, and encapsulate received packets towards the appropriate ETR. Note that for the reverse path LISP sites can reach non-LISP sites simply by not encapsulating traffic. See [I-D.ietf-lisp-interworking] for a detailed description of P-ITR functionality.

The success of new protocols depends greatly on their ability to maintain backwards compatibility and inter-operate with the protocol(s) they intend to enhance or replace, and on the incentives to deploy the necessary new software or equipment. A LISP site needs an interworking mechanism to be reachable from non-LISP sites. A P-ITR can fulfill this role, enabling early adopters to see the benefits of LISP, similar to tunnel brokers helping the transition from IPv4 to IPv6. A site benefits from new LISP functionality (proportionally with existing global LISP deployment) when going LISP, so it has the incentives to deploy the necessary tunnel routers. In order to be reachable from non-LISP sites it has two options: keep announcing its prefix(es) with BGP (see next subsection), or have a P-ITR announce prefix(es) covering them.

If the goal of reducing the DFZ routing table size is to be reached, the second option is preferred. Moreover, the second option allows LISP-based ingress traffic engineering from all sites. However, the placement of P-ITRs greatly influences performance and deployment incentives. The following subsections present the LISP+BGP transition strategy and then possible P-ITR deployment scenarios. They use the loosely defined terms of "early transition phase", "late transition phase", and "LISP Internet phase", which refer to time periods when LISP sites are a minority, a majority, or represent all edge networks respectively.

4.1.1. LISP+BGP

For sites wishing to go LISP with their PI prefix the least disruptive way is to upgrade their border routers to support LISP, register the prefix into the LISP mapping system, but keep announcing it with BGP as well. This way LISP sites will reach them over LISP, while legacy sites will be unaffected by the change. The main disadvantage of this approach is that no decrease in the DFZ routing table size is achieved. Still, just increasing the number of LISP sites is an important gain, as an increasing LISP/non-LISP site ratio will slowly decrease the need for BGP-based traffic engineering that leads to prefix deaggregation. That, in turn, may lead to a decrease in the DFZ size in the late transition phase.

This scenario is not limited to sites that already have their prefixes announced with BGP. Newly allocated EID blocks could follow

this strategy as well during the early LISP deployment phase, depending on the cost/benefit analysis of the individual networks. Since this leads to an increase in the DFZ size, one of the following scenarios should be preferred for new allocations.

4.1.2. Mapping Service Provider P-ITR Service

In addition to publishing their clients' registered prefixes in the mapping system, MSPs with enough transit capacity can offer them P-ITR service as a separate service. This service is especially useful for new PI allocations, to sites without existing BGP infrastructure, that wish to avoid BGP altogether. The MSP announces the prefix into the DFZ, and the client benefits from ingress traffic engineering without prefix deaggregation. The downside of this scenario is path stretch, which may be greater than 1.

Routing all non-LISP ingress traffic through a third party which is not one of its ISPs is only feasible for sites with modest amounts of traffic (like those using the IPv6 tunnel broker services today), especially in the first stage of the transition to LISP, with a significant number of legacy sites. When the LISP/non-LISP site ratio becomes high enough, this approach can prove increasingly attractive.

Compared to LISP+BGP, this approach avoids DFZ bloat caused by prefix deaggregation for traffic engineering purposes, resulting in slower routing table increase in the case of new allocations and potential decrease for existing ones. Moreover, MSPs serving different clients with adjacent aggregable prefixes may lead to additional decrease, but quantifying this decrease is subject to future research study.

4.1.3. Tier 1 P-ITR Service

The ideal location for a P-ITR is on the traffic path, as close to non-LISP site as possible, to minimize or completely eliminate path stretch. However, this location is far away from the networks that most benefit from the P-ITR services (i.e., LISP sites, destinations of encapsulated traffic) and have the most incentives to deploy them. But the biggest challenge having P-ITRs close to the traffic source is the large number of devices and their wide geographical diversity required to have a good coverage, in addition to considerable transit capacity. Tier 1 service providers fulfill these requirements and have clear incentives to deploy P-ITRs: to attract more traffic from their customers. Since a large fraction is multihomed to different providers with more than one active link, they compete with the other providers for traffic.

To operate the P-ITR service, the ISP announces an aggregate of all

known EID prefixes (a mechanism will be needed to obtain this list) downstream to their customers with BGP. First, the performance concerns of the MSP P-ITR service described in the previous section are now addressed, as P-ITRs are on-path, eliminating path stretch (except when combined with LISP+BGP, see below). Second, thanks to the direction of the announcements, the DFZ routing table size is not affected.

The main downside of this approach is non-global coverage for the announced prefixes, caused by the downstream direction of the announcements. As a result, a LISP site will be only reachable from customers of service providers running P-ITRs, unless one of the previous approaches is used as well. Due to this issue, it is unlikely that existing BGP speakers migrating to LISP will withdraw their announcements to the DFZ, resulting in a combination of this approach with LISP+BGP. At the same time, smaller new LISP sites still depend on MSP for global reachability. The early transition phase thus will keep the status quo in the DFZ routing table size, but offers the benefits of increasingly better ingress traffic engineering to early adopters.

As the number of LISP destinations increases, traffic levels from those non-LISP, large multihomed clients who rely on BGP path length for provider selection (such as national/regional ISPs), start to shift towards the Tier 1 providing P-ITRs. The competition is then incentivised to deploy their own service, thus improving global P-ITR coverage. If all Tier 1 providers have P-ITR service, the LISP+BGP and MSP alternatives are not required for global reachability of LISP sites. Still, LISP+BGP users may still want to keep announcing their prefixes for security reasons (i.e., preventing hijacking). DFZ size evolution in this phase depends on that choice, and the aggregability of all LISP prefixes. As a result, it may decrease or stay at the same level.

For performance reasons, and to simplify P-ITR implementations, it is desirable to minimize the number of non-aggregable EID prefixes. In IPv6 this can be easily achieved if a large prefix block is reserved as LISP EID space [I-D.meyer-lisp-eid-block]. If the EID space is not fragmented, new LISP sites will not cause increase in the DFZ size, unless they do LISP+BGP.

To summarize, the main benefits of this scenario are stopping the increase and potentially decreasing the size of the DFZ routing tables, while keeping path stretch close to 1, with the cost of not having global coverage of one’s prefixes.

4.1.4. Migration Summary

The following table presents the expected effects of the different transition scenarios during a certain phase on the DFZ routing table size:

Phase	LISP+BGP	MSP	Tier 1
Early transition	no change	slowdown increase	no change
Late transition	may decrease	slowdown increase	may decrease
LISP Internet	considerable decrease		

It is expected that a combination of these scenarios will exist during the migration period, in particular existing sites choosing LISP+BGP, new small sites choosing MSP, and competition between Tier 1 providers bringing optimized service. If all Tier 1 ISPs have P-ITR service in place, the other scenarios can be deprecated, greatly reducing DFZ size.

4.2. P-ETR

In contrast to P-ITRs, P-ETRs are not required for the correct functioning of all LISP sites. There are two cases, where they can be of great help:

- LISP sites with unicast reverse path forwarding (uRPF) restrictions, and
- LISP sites without native IPv6 communicating with LISP nodes with IPv6-only locators.

In the first case, uRPF filtering is applied at their upstream PE router. When forwarding traffic to non-LISP sites, an ITR does not encapsulate packets, leaving the original IP headers intact. As a result, packets will have EIDs in their source address. Since we are discussing the transition period, we can assume that a prefix covering the EIDs belonging to the LISP site is advertised to the global routing tables by a P-ITR, and the PE router has a route towards it. However, the next hop will not be on the interface towards the CE router, so non-encapsulated packets will fail uRPF checks.

To avoid this filtering, the affected ITR encapsulates packets towards the locator of the P-ETR for non-LISP destinations. Now the source address of the packets, as seen by the PE router is the ITR's locator, which will not fail the uRPF check. The P-ETR then decapsulates and forwards the packets.

The second use case is IPv4-to-IPv6 transition. Service providers using older access network hardware, which only supports IPv4 can still offer IPv6 to their clients, by providing a CPE device running LISP, and P-ETR(s) for accessing IPv6-only non-LISP sites and LISP sites, with IPv6-only locators. Packets originating from the client LISP site for these destinations would be encapsulated towards the P-ETR's IPv4 locator. The P-ETR is in a native IPv6 network, decapsulating and forwarding packets. For non-LISP destination, the packet travels natively from the P-ETR. For LISP destinations with IPv6-only locators, the packet will go through a P-ITR, in order to reach its destination.

For more details on P-ETRs see the [I-D.ietf-lisp-interworking] draft.

P-ETRs can be deployed by ISPs wishing to offer value-added services to their customers. As is the case with P-ITRs, P-ETRs too may introduce path stretch. Because of this the ISP needs to consider the tradeoff of using several devices, close to the customers, to minimize it, or few devices, farther away from the customers, minimizing cost instead.

Since the deployment incentives for P-ITRs and P-ETRs are different, it is likely they will be deployed in separate devices, except for the CDN case, which may deploy both in a single device.

In all cases, the existence of a P-ETR involves another step in the configuration of a LISP router. CPE routers, which are typically configured by DHCP, stand to benefit most from P-ETRs. To enable autoconfiguration of the P-ETR locator, a DHCP option would be required.

As a security measure, access to P-ETRs should be limited to legitimate users by enforcing ACLs.

5. Security Considerations

Security implications of LISP deployments are to be discussed in separate documents. [I-D.saucez-lisp-security] gives an overview of LISP threat models, while securing mapping lookups is discussed in [I-D.maino-lisp-sec].

6. IANA Considerations

This memo includes no request to IANA.

7. Acknowledgements

Many thanks to Margaret Wasserman for her contribution to the IETF76 presentation that kickstarted this work. The authors would also like to thank Damien Saucez, Luigi Iannone, Joel Halpern, Vince Fuller, Dino Farinacci, Terry Manderson, Noel Chiappa, and everyone else who provided input.

8. References

8.1. Normative References

[I-D.ietf-lisp]
Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol (LISP)", draft-ietf-lisp-10 (work in progress), March 2011.

[I-D.ietf-lisp-alt]
Fuller, V., Farinacci, D., Meyer, D., and D. Lewis, "LISP Alternative Topology (LISP+ALT)", draft-ietf-lisp-alt-06 (work in progress), March 2011.

[I-D.ietf-lisp-interworking]
Lewis, D., Meyer, D., Farinacci, D., and V. Fuller, "Interworking LISP with IPv4 and IPv6", draft-ietf-lisp-interworking-01 (work in progress), August 2010.

[I-D.ietf-lisp-ms]
Fuller, V. and D. Farinacci, "LISP Map Server", draft-ietf-lisp-ms-07 (work in progress), March 2011.

[I-D.maino-lisp-sec]
Maino, F., Ermagan, V., Cabellos-Aparicio, A., Saucez, D., and O. Bonaventure, "LISP-Security (LISP-SEC)", draft-maino-lisp-sec-00 (work in progress), March 2011.

[I-D.saucez-lisp-security]
Saucez, D., Iannone, L., and O. Bonaventure, "LISP Security Threats", draft-saucez-lisp-security-03 (work in progress), March 2011.

8.2. Informative References

[I-D.lear-lisp-nerd]
Lear, E., "NERD: A Not-so-novel EID to RLOC Database", draft-lear-lisp-nerd-08 (work in progress), March 2010.

[I-D.meyer-lisp-eid-block]
Iannone, L., Lewis, D., Meyer, D., and V. Fuller, "LISP EID Block", draft-meyer-lisp-eid-block-02 (work in progress), March 2011.

[cache] Jung, J., Sit, E., Balakrishnan, H., and R. Morris, "DNS performance and the effectiveness of caching", 2002.

Authors' Addresses

Lorand Jakab
Technical University of Catalonia
C/Jordi Girona, s/n
BARCELONA 08034
Spain

Email: ljakab@ac.upc.edu

Albert Cabellos-Aparicio
Technical University of Catalonia
C/Jordi Girona, s/n
BARCELONA 08034
Spain

Email: acabello@ac.upc.edu

Florin Coras
Technical University of Catalonia
C/Jordi Girona, s/n
BARCELONA 08034
Spain

Email: fcoras@ac.upc.edu

Jordi Domingo-Pascual
Technical University of Catalonia
C/Jordi Girona, s/n
BARCELONA 08034
Spain

Email: jordi.domingo@ac.upc.edu

Darrel Lewis
Cisco Systems
170 Tasman Drive
San Jose, CA 95134
USA

Email: darlewis@cisco.com

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: September 5, 2011

F. Maino
V. Ermagan
Cisco Systems
A. Cabellos
Technical University of Catalonia
D. Saucez
O. Bonaventure
Universite catholique de Louvain
March 4, 2011

LISP-Security (LISP-SEC)
draft-maino-lisp-sec-00.txt

Abstract

This memo specifies LISP-SEC, a set of security mechanisms that provide origin authentication, integrity and anti-replay protection to LISP's EID-to-RLOC mapping data. LISP-SEC also enables verification of authorization on EID prefix claims.

Requirements Language

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

Status of this Memo

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

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

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

This Internet-Draft will expire on September 5, 2011.

Copyright Notice

Table of Contents

1. Introduction

The Locator/ID Separation Protocol [I-D.ietf-lisp] defines a set of functions for routers to exchange information used to map from non-routable Endpoint Identifiers (EIDs) to routable Routing Locators (RLOCs). If these EID-to-RLOC mappings, carried through Map-Reply messages, are transmitted without integrity protection, an adversary can manipulate them and hijack the communication, impersonate the requested EID or mount Denial of Service or Distributed Denial of Service attacks. Also, if the Map-Reply message is transported unauthenticated, an adversarial LISP entity can overclaim an EID-prefix and maliciously redirect traffic directed to a large number of hosts. A detailed description of "overclaiming" attack is provided in [I-D.saucez-lisp-security].

This memo specifies LISP-SEC, a set of sceurity mechanisms that provide origin authentication, integrity and anti-replay protection to LISP's EID-to-RLOC mapping data. LISP-SEC also enables verification of authorization on EID prefix claims, ensuring that the entity that provides the location for a given EID prefix is entitled to do so.

2. Definition of Terms

One-Time Key (OTK): An ephemeral randomly generated key that must be used for a single Map-Request/Map-Reply exchange.

Encapsulated Control Message (ECM): A LISP control message that is prepended with an additional LISP header. ECM is used by ITRs to send LISP control messages to a Map-Resolver, by Map-Resolvers to forward LISP control messages to a Map-Server, and by Map-Resolvers to forward LISP control messages to an ETR.

Authentication Data (AD): Metadata that is included either in a LISP ECM header or in a Map-Reply message to support confidentiality, integrity protection, and verification of EID prefix authorization.

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

3. LISP-SEC Threat Model

LISP-SEC addresses the control plane threats, described in

[I-D.saucez-lisp-security], that target EID-to-RLOC mappings, including manipulations of Map-Request and Map-Reply messages, and malicious xTR EID overclaiming. However LISP-SEC makes two main assumptions that are not part of [I-D.saucez-lisp-security]. First, the LISP Mapping System is expected to deliver Map-Request messages to their intended destinations as identified by the EID. Second, no Man-in-the-Middle (MiM) attack can be mounted within the LISP Mapping System.

Accordingly to the threat model described in [I-D.saucez-lisp-security] LISP-SEC assumes that any kind of attack, including MiM attacks, can be mounted in the access network, outside of the boundaries of the LISP mapping system. An on-path attacker, outside of the LISP mapping service system can, for instance, hijack mapping requests and replies, spoofing the identity of a LISP node. Another example of on-path attack, called over claiming attack, can be mounted by a malicious Egress Tunnel Router (ETR), by over claiming the EID prefixes for which it is authoritative. In this way the ETR can maliciously redirect traffic directed to a large number of hosts.

4. Protocol Operations

The goal of the security mechanisms defined in [I-D.ietf-lisp] is to prevent unauthorized insertion of mapping data, by providing origin authentication and integrity protection for the Map-Registration, and by using the nonce to detect unsolicited Map-Reply sent by off-path attackers.

LISP-SEC builds on top of the security mechanisms defined in [I-D.ietf-lisp] to address the threats described in Section 3 by leveraging the trust relationships existing among the LISP entities participating to the exchange of the Map-Request/Map-Reply messages. Those trust relationships are used to securely distribute a One-Time Key (OTK) that provides origin authentication, integrity and anti-replay protection to mapping protocol data, and that effectively prevent over claiming attacks. The processing of security parameters during the Map-Request/Map-Reply exchange is as follows:

- The OTK is generated and stored at the ITR, and securely transported to the Map-Server.

- The Map-Server uses the OTK to compute an HMAC that protects the integrity of the mapping data provided by the Map-Server to prevent overclaiming attacks. The Map-Server also derives a new OTK (OTK-ETR), by applying a Key Derivation Function (KDF) to the original OTK, that is passed to the ETR.

- The ETR uses the new OTK to compute an HMAC that protects the integrity of the Map-Reply sent to the ITR.

- Finally, the ITR uses the stored OTK to verify the integrity of the mapping data provided by both the Map-Server and the ETR, and to verify that no overclaiming attacks were mounted along the path between the Map-Server and the ITR.

Section 5 provides the detailed description of the LISP-SEC control messages and their processing, while the rest of this section describes the flow of protocol operations at each entity involved in the Map-Request/Map-Reply exchange:

- The ITR, upon transmitting a Map-Request message, generates and stores an OTK. This key is included into the Encapsulated Control Message (ECM) that contains the Map-Request sent to the Map-Resolver. To provide OTK confidentiality over the path between the ITR and its Map-Resolver, the OTK SHOULD be encrypted using a preconfigured key shared between the ITR and the Map-Resolver, similar to the key shared between the ETR and the Map-Server in order to secure ETR registration [I-D.ietf-lisp-ms].

- The Map-Resolver decapsulates the ECM message, decrypts the OTK, if needed, and forwards through the Mapping System the received Map-Request and the OTK, as part of a new ECM message. As described in Section 5.5, the LISP Mapping System delivers the ECM to the appropriate Map-Server, as identified by the EID destination address of the Map-Request.

- The Map-Server is configured with the location mappings and policy information for the ETR responsible for the destination EID address. Using this preconfigured information the Map-Server, after the decapsulation of the ECM message, finds the longest match EID prefix that covers the requested EID in the received Map-Request. The Map-Server adds this EID prefix, together with an HMAC computed using the OTK, to a new Encapsulated Control Message that contains the received Map-Request.

- The Map-Server derives a new OTK (OTK-ETR) by applying a Key Derivation Function (KDF) to the OTK. This new OTK is included in the Encapsulated Control Message sent to the ETR. To provide OTK confidentiality over the path between the Map-Server and the ETR, the new OTK should be encrypted using the key shared between the ETR and the Map-Server in order to secure ETR registration [I-D.ietf-lisp-ms].

- If the Map-Server is acting in proxy mode, as specified in [I-D.ietf-lisp], the ETR is not involved in the origination of the

Map-Reply. In this case the Map-Server originates the Map-Reply on behalf of the ETR as described below.

- The ETR, upon receiving the Encapsulated Map-Request from the Map-Server, decrypts the OTK-ETR, if needed, and originates a Map-Reply that contains the EID-to-RLOC mapping information as specified in [I-D.ietf-lisp].

- The ETR computes an HMAC over the original LISP Map-Reply, keyed with OTK-ETR to protect the integrity of the whole Map-Reply. The ETR also copies the EID prefix authorization data that the Map-Server included in the Encapsulated Map-Request into the Map-Reply message.

- The ITR, upon receiving the Map-Reply, uses the locally stored OTK to verify the integrity of the EID prefix authorization data included in the Map-Reply by the Map-Server. The ITR computes OTK-ETR by applying the same KDF used by the Map-Server, and verifies the integrity of the Map-Reply. If the integrity checks fail the Map-Reply MUST be discarded. Also, if the EID prefix claimed by the ETR in the Map-Reply is less specific than the EID prefix authorization data inserted by the Map-Server, the ITR MUST discard the Map-Reply.

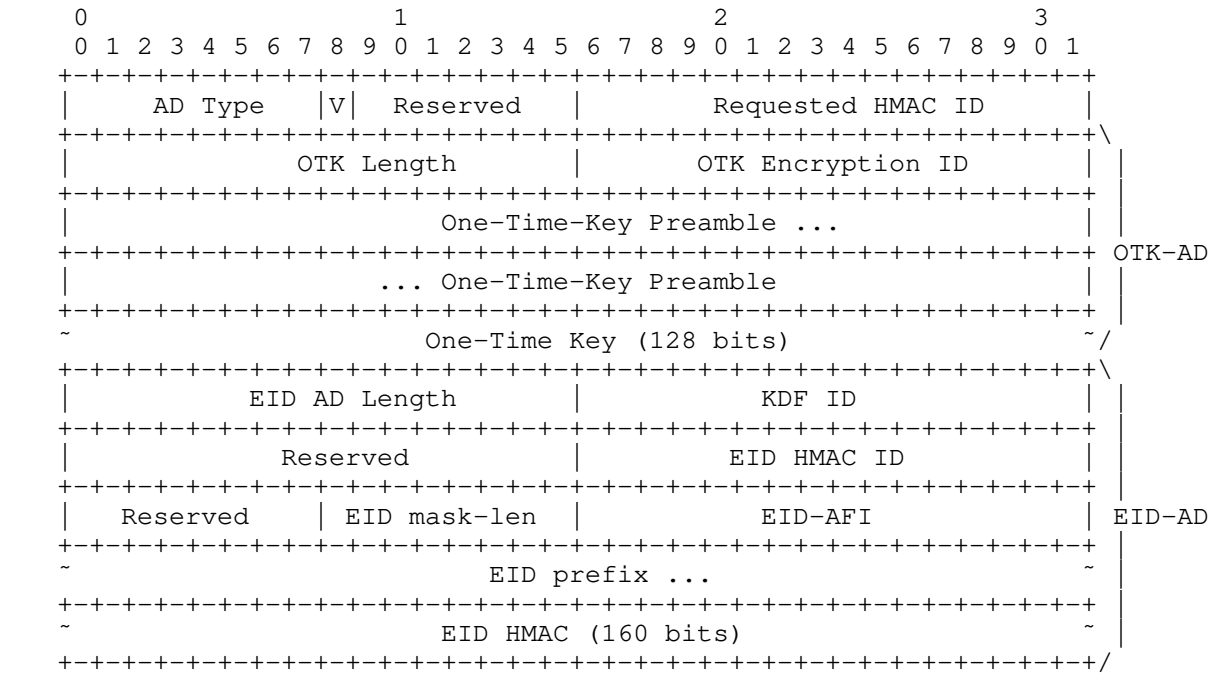
5. LISP-SEC Control Messages Details

LISP-SEC metadata associated with a Map-Request is transported within the Encapsulated Control Message that contains the Map-Request.

LISP-SEC metadata associated with the Map-Reply is transported within the Map-Reply itself.

5.1. Encapsulated Control Message LISP-SEC Extensions

LISP-SEC uses the ECM (Encapsulated Control Message) defined in [I-D.ietf-lisp] with Type set to 8, and S bit set to 1 to indicate that the LISP header includes Authentication Data (AD). The format of the LISP-SEC ECM Authentication Data is defined in the following figure. OTK-AD stands for One-Time Key Authentication Data and EID-AD stands for EID Authentication Data.



LISP-SEC ECM Authentication Data

AD Type: 1 (LISP-SEC Authentication Data)

V: Key Version bit. This bit is toggled when the sender switches to a new OTK wrapping key

Reserved: Set to 0 on transmission and ignored on receipt.

Requested HMAC ID: the HMAC algorithm requested by the ITR. See Section 5.3 for details.

OTK Length: The length (in bytes) of the OTK Authentication Data (OTK-AD), that contains the OTK Preamble and the OTK.

OTK Encryption ID: The identifier of the key wrapping algorithm used to encrypt the One-Time-Key. When a 128-bit OTK is sent unencrypted by the Map-Resolver, the OTK Encryption ID is set to NULL_KEY_WRAP_128. See Section 5.4 for more details.

One-Time-Key Preamble: set to 0 if the OTK is not encrypted. When the OTK is encrypted, this field may carry additional metadata resulting from the key wrapping operation. When a 128-bit OTK is sent unencrypted by Map-Resolver, the OTK Preamble is set to

0x0000000000000000 (64 bits). See Section 5.4 for details.

One-Time-Key: the OTK encrypted (or not) as specified by OTK Encryption ID. See Section 5.4 for details.

EID AD Length: length (in bytes) of the EID Authentication Data (EID-AD). The ITR MUST set EID AD Length to 32, as it only fills the KDF ID field, and all the remaining fields part of the EID-AD are not present.

KDF ID: Identifier of the Key Derivation Function used to derive OTK-ETR. The ITR SHOULD use this field to indicate the recommended KDF algorithm, according to local policy. The Map-Server can overwrite the KDF ID if it does not support the KDF ID recommended by the ITR. See Section 5.4 for more details.

Reserved: Set to 0 on transmission and ignored on receipt.

EID HMAC ID: Identifier of the HMAC algorithm used to protect the integrity of the EID prefix authorization fields. This field is filled by Map-Server that computed the EID prefix HMAC. See Section 5.4 for more details.

EID mask-len: Mask length for EID prefix.

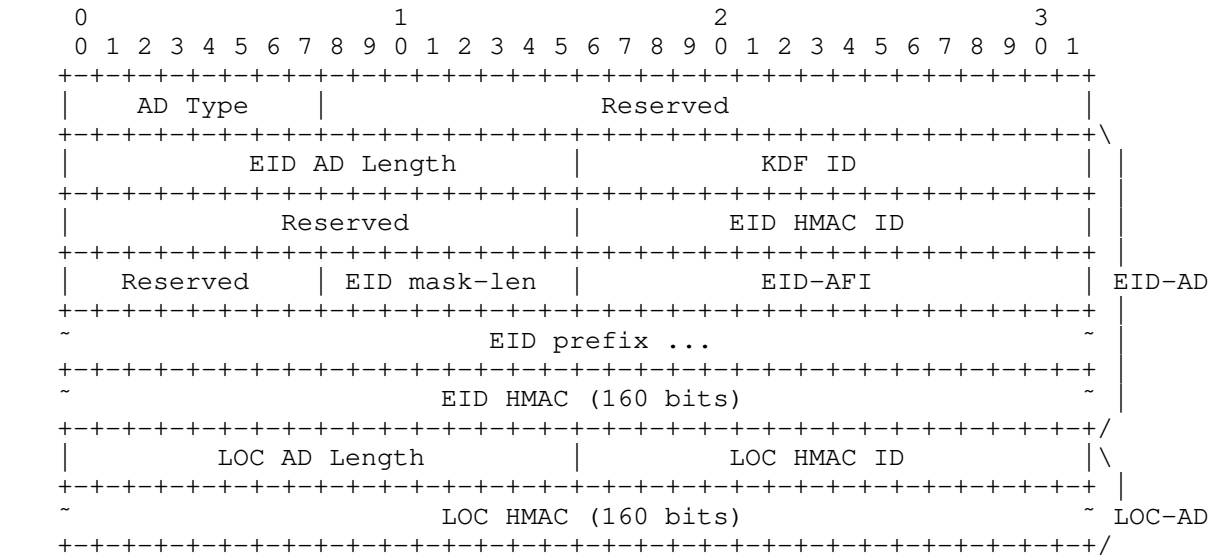
EID-AFI: Address family of EID-prefix according to [RFC5226]

EID prefix: The Map-Server uses this field to specify the EID prefix that the destination ETR is authoritative for, and is the longest match for the requested EID.

EID HMAC: HMAC of the EID prefix authorization fields that is computed and inserted by Map-Server. Before computing the HMAC operation the EID HMAC field MUST be set to 0. The HMAC covers the entire EID-AD.

5.2. Map-Reply LISP-SEC Extensions

LISP-SEC uses the Map-Reply defined in [I-D.ietf-lisp], with Type set to 2, and S bit set to 1 to indicate that the Map-Reply message includes Authentication Data (AD). The format of the LISP-SEC Map-Reply Authentication Data is defined in the following figure. LOC-AD stands for LOC Authentication Data.



LISP-SEC Map-Reply Authentication Data

AD Type: 1 (LISP-SEC Authentication Data)

EID AD Length: length (in bytes) of the EID-AD.

KDF ID: Identifier of the Key Derivation Function used to derive OTK-ETR. See Section 5.6 for more details.

Reserved: Set to 0 on transmission and ignored on receipt.

EID HMAC ID: Identifier of the HMAC algorithm used to protect the integrity of the EID prefix authorization fields. See Section 5.6 for more details.

EID mask-len: Mask length for EID prefix.

EID-AFI: Address family of EID-prefix according to [RFC5226].

EID prefix: This field contains the EID prefix that the destination ETR is authoritative for, and is the longest match for the requested EID.

EID HMAC: HMAC of the EID prefix authorization fields. Before computing the HMAC operation the EID HMAC field MUST be set to 0.

LOC AD Length: length (in bytes) of the Map-Reply Location Authentication Data (LOC-AD).

LOC HMAC ID: Identifier of the HMAC algorithm used to protect the integrity of the Map-reply Location Data.

LOC HMAC: HMAC of the Map-reply Location Data. The scope of the authentication covers the whole Map-Reply Payload (from Type to Mapping Protocol Data fields included). See Section 5.7 for more details.

5.3. ITR Processing

Upon creating a Map-Request, the ITR generates a random OTK that is stored locally, together with the nonce generated as specified in [I-D.ietf-lisp].

The Map-Request MUST be encapsulated in an ECM, with the S-bit set to 1, to indicate the presence of Authentication Data. If the ITR and the Map-Resolver are configured with a shared key, the OTK confidentiality SHOULD be protected by wrapping the OTK with the algorithm specified by the OTK Encryption ID field. See Section 5.4 for further details on OTK encryption.

The Requested HMAC ID field contains the suggested HMAC algorithm to be used by the Map-Server and the ETR to protect the integrity of the ECM Authentication data and of the Map-Reply.

The KDF ID field, specifies the suggested key derivation function to be used by the Map-Server to derive the OTK-ETR.

The EID AD length is set to 32, since the Authentication Data does not contain EID prefix Authentication Data, and the EID-AD contains only the KDF ID field.

In response to an encapsulated Map-Request that has the S-bit set, an ITR MUST receive a Map-Reply with the S-bit set, that includes an EID AD and a LOC AD. If the Map-Reply does not include both ADs, the ITR MUST discard it. In response to an encapsulated Map-Request with S-bit set to 0, the ITR expects a Map-Reply with S-bit set to 0, and the ITR SHOULD discard the Map-Reply if the S-bit is set.

Upon receiving a Map-Reply, the ITR must verify the integrity of both the EID-AD and the LOC-AD, and MUST discard the Map-Reply if one of the integrity checks fails.

The integrity of the EID-AD is verified using the locally stored OTK to re-compute the HMAC of the EID-AD using the Algorithm specified in the EID HMAC ID field. If the EID HMAC ID field does not match the Requested HMAC ID the ITR SHOULD discard the Map-Reply and send a new Map-Request with a different Requested HMAC ID field, according to

ITR's local policy. The ITR MUST set the EID HMAC ID field to 0 before computing the HMAC.

To verify the integrity of the LOC-AD, first the OTK-ETR is derived from the locally stored OTK using the algorithm specified in the KDF ID field. This is because the LOC AD is generated by the ETR using the OTK-ETR. If the KDF ID in the Map-Reply does not match the KDF ID requested in the Map-Request, the ITR SHOULD discard the Map-Reply, and send a new Map-Request with a different KDF ID, according to ITR's local policy. The derived OTK-ETR is then used to re-compute the HMAC of the LOC-AD using the Algorithm specified in the LOC HMAC ID field. If the LOC HMAC ID field does not match the Requested HMAC ID the ITR SHOULD discard the Map-Reply, and send a new Map-Request with a new Required HMAC ID according to ITR's local policy.

The Map-Reply is considered a valid Map-Reply only if: (1) both EID-AD and LOC-AD are valid, and (2) the EID prefixes in the Map-Reply records are equal to or more specific than the EID prefix in the EID-AD. After identifying the Map-Reply as valid, the ITR proceeds to adding the Map-Reply records to its EID-to-RLOC cache, as described in [I-D.ietf-lisp].

The ITR SHOULD send SMR triggered Map Requests over the mapping system in order to receive a secure Map-Reply. If an ITR accepts piggybacked Map-Replies, it SHOULD also send a Map-Request over the mapping system in order to securely verify the piggybacked Map-Reply.

5.4. Encrypting and Decrypting an OTK

If OTK confidentiality is required in the path between the Map-Server and the ETR, the OTK SHOULD be encrypted using the preconfigured key shared between the Map-Server and the ETR for the purpose of securing ETR registration [I-D.ietf-lisp-ms]. Similarly, if OTK confidentiality is required in the path between the ITR and the Map-Resolver, the OTK SHOULD be encrypted with a key shared between the ITR and the Map-Resolver.

The OTK is encrypted using the algorithm specified in the OTK Encryption ID field. When the AES Key Wrap algorithm is used to encrypt a 128-bit OTK, according to [RFC3339], the AES Key Wrap Initialization Value MUST be set to 0xA6A6A6A6A6A6A6A6 (64 bits). The output of the AES Key Wrap operation is 192-bit long. The most significant 64-bit are copied in the One-Time Key Preamble field, while the 128 less significant bits are copied in the One-Time Key field of the LISP-SEC Authentication Data.

When decrypting an encrypted OTK the receiver MUST verify that the

Initialization Value resulting from the AES Key Wrap decryption operation is equal to 0xA6A6A6A6A6A6A6A6. If this verification fails the receiver MUST discard the entire message.

When a 128-bit OTK is sent unencrypted the OTK Encryption ID is set to NULL_KEY_WRAP_128, and the OTK Preamble is set to 0x0000000000000000 (64 bits).

5.5. Map-Resolver Processing

Upon receiving an encapsulated Map-Request with the S-bit set, the Map-Resolver decapsulates the ECM message. The OTK, if encrypted, is decrypted as specified in Section 5.4.

The Map-Resolver, as specified in [I-D.ietf-lisp-ms], originates a new ECM header with the S-bit set, that contains the unencrypted OTK, as specified in Section 5.4, and the other data derived from the ECM Authentication Data of the received encapsulated Map-Request.

The Map-Resolver then forwards the received Map-Request, encapsulated in the new ECM header that includes the newly computed Authentication Data fields.

5.6. Map-Server Processing

Upon receiving an encapsulated Map-Request with the S-bit set, the Map-Server decapsulates the ECM and generates a new ECM Authentication Data. The Authentication Data includes the OTK-AD and the EID-AD, that contains EID prefix authorization information, that are ultimately sent to the requesting ITR.

The Map-Server updates the OTK-AD by deriving a new OTK (OTK-ETR) from the OTK received with the Map-Request. OTK-ETR is derived applying the key derivation function specified in the KDF ID field. If the algorithm specified in the KDF ID field is not supported, the Map-Server uses a different algorithm to derive the key and updates the KDF ID field accordingly.

The Map-Server and the ETR MUST be configured with a shared key for mapping registration according to [I-D.ietf-lisp-ms]. If OTK confidentiality is required, then the OTK-ETR SHOULD be encrypted, by wrapping the OTK-ETR with the algorithm specified by the OTK Encryption ID field as specified in Section 5.4.

The Map-Server includes in the EID AD the longest match registered EID prefix for the destination EID, and an HMAC of this EID prefix. The HMAC is keyed with the OTK in the ECM Authentication Data that is received from ITR, and the HMAC algorithm is chosen according to the

Requested HMAC ID field. If The Map-Server does not support this algorithm, the Map-Server uses a different algorithm and specifies it in the EID HMAC ID field. The scope of the HMAC operation covers the entire EID-AD, from the EID-AD Length field to the EID HMAC field, which must be set to 0 before the computation.

The Map-Server then forwards the updated ECM encapsulated Map-Request, that contains the OTK-AD, the EID-AD, and the received Map-Request to an authoritative ETR as specified in [I-D.ietf-lisp].

5.6.1. Map-Server Processing in Proxy mode

If the Map-Server is in proxy mode, it generates a Map-Reply, as specified in [I-D.ietf-lisp], with the S-bit set to 1. The Map-Reply includes the Authentication Data that contains the EID AD, computed as specified in Section 5.6, as well as the LOC-AD computed as specified in Section 5.7.

5.7. ETR Processing

Upon receiving an encapsulated Map-Request with the S-bit set, the ETR decapsulates the ECM message. The OTK field, if encrypted, is decrypted as specified in Section 5.4 to obtain the unencrypted OTK-ETR.

The ETR then generates a Map-Reply as specified in [I-D.ietf-lisp] and includes an Authentication Data that contains the EID-AD, as received in the encapsulated Map-Request, as well as the LOC-AD.

The EID-AD is copied from the Authentication Data of the received encapsulated Map-Request.

The LOC-AD contains the HMAC of the whole Map-Reply message, keyed with the OTK-ETR and computed using the HMAC algorithm specified in the Requested HMAC ID field of the received encapsulated Map-Request. If the ETR does not support the Requested HMAC ID, it uses a different algorithm and updates the LOC HMAC ID field accordingly. Finally the ETR sends the Map-Reply to the requesting ITR as specified in [I-D.ietf-lisp].

6. Security Considerations

6.1. Mapping System Security

The LISP-SEC threat model described in Section 3, assumes that the LISP Mapping System is working properly and eventually delivers Map-Request messages to a Map-Server that is authoritative for the

requested EID.

Security is not yet embedded in LISP+ALT but BGP route filtering SHOULD be deployed in the ALT infrastructure to enforce proper routing in the mapping system. The SIDR working group is currently addressing prefix and route advertisement authorization and authentication for BGP. While following SIDR recommendations in the global Internet will take time, applying these recommendations to the ALT, which relies on BGP, should be less complex, as ALT is currently small and with a limited number of operators. Ultimately, deploying the SIDR recommendations in ALT further ensures that the fore mentioned assumption is true.

It is also assumed that no man-in-the-middle attack can be carried out against the ALT router to ALT router tunnels, and that the information included into the Map-Requests, in particular the OTK, cannot be read by third-party entities. It should be noted that the integrity of the Map-Request in the ALT is protected by BGP authentication, and that in order to provide OTK confidentiality in the ALT mapping system the ALT router to ALT router tunnels MAY be deployed using GRE+IPSec.

6.2. Random Number Generation

The OTK MUST be generated by a properly seeded pseudo-random (or strong random) source. See [RFC4086] for advice on generating security-sensitive random data

7. IANA Considerations

7.1. HMAC functions

The following HMAC ID values are defined by this memo for use as Requested HMAC ID, EID HMAC ID, and LOC HMAC ID in the LISP-SEC Authentication Data:

Name	Number	Defined In
NONE	0	
AUTH-HMAC-SHA-1-160	1	[RFC2104]
AUTH-HMAC-SHA-256-128	2	[RFC4634]

values 2-65535 are reserved to IANA.

HMAC Functions

AUTH-HMAC-SHA-1-160 MUST be supported, AUTH-HMAC-SHA-256-128 should be supported.

7.2. Key Wrap Functions

The following OTK Encryption ID values are defined by this memo for use as OTK key wrap algorithms ID in the LISP-SEC Authentication Data:

Name	Number	Defined In
NULL-KEY-WRAP-128	1	
AES-KEY-WRAP-128	2	[RFC3394]

values 0 and 3-65535 are reserved to IANA.

Key Wrap Functions

NULL-KEY-WRAP-128, and AES-KEY-WRAP-128 MUST be supported.

NULL-KEY-WRAP-128 is used to carry an unencrypted 128-bit OTK, with a 64-bit preamble set to 0x0000000000000000 (64 bits).

7.3. Key Derivation Functions

The following KDF ID values are defined by this memo for use as KDF ID in the LISP-SEC Authentication Data:

Name	Number	Defined In
NONE	0	
HKDF-SHA1-128	1	[RFC5869]

values 2-65535 are reserved to IANA.

Key Derivation Functions

HKDF-SHA1-128 MUST be supported

8. Acknowledgements

The authors would like to acknowledge Pere Monclus, Dave Meyer, Dino Farinacci, Brian Weis, David McGrew, Darrel Lewis and Landon Curt Noll for their valuable suggestions provided during the preparation of this document.

9. Normative References

[I-D.ietf-lisp]
Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol (LISP)", draft-ietf-lisp-10 (work in progress), March 2011.

[I-D.ietf-lisp-interworking]
Lewis, D., Meyer, D., Farinacci, D., and V. Fuller, "Interworking LISP with IPv4 and IPv6", draft-ietf-lisp-interworking-01 (work in progress), August 2010.

[I-D.ietf-lisp-ms]
Fuller, V. and D. Farinacci, "LISP Map Server", draft-ietf-lisp-ms-06 (work in progress), October 2010.

[I-D.saucez-lisp-security]
Saucez, D., Iannone, L., and O. Bonaventure, "LISP Security Threats", draft-saucez-lisp-security-02 (work in progress), January 2011.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3394] Schaad, J. and R. Housley, "Advanced Encryption Standard (AES) Key Wrap Algorithm", RFC 3394, September 2002.

[RFC4086] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5869] Krawczyk, H. and P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)", RFC 5869, May 2010.

Authors' Addresses

Fabio Maino
Cisco Systems
170 Tasman Drive
San Jose, California 95134
USA

Email: fmaino@cisco.com

Vina Ermagan
Cisco Systems
170 Tasman Drive
San Jose, California 95134
USA

Email: vermagan@cisco.com

Albert Cabellos
Technical University of Catalonia
c/ Jordi Girona s/n
Barcelona, 08034
Spain

Email: acabello@ac.upc.edu

Damien Saucez
Universite catholique de Louvain
Place St. Barbe 2
Louvain-la-Neuve,
Belgium

Email: damien.saucez@uclouvain.be

Olivier Bonaventure
Universite catholique de Louvain
Place St. Barbe 2
Louvain-la-Neuve,
Belgium

Email: olivier.bonaventure@uclouvain.be

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 13, 2011

L. Iannone
Deutsche Telekom Laboratories
D. Lewis
D. Meyer
V. Fuller
Cisco Systems, Inc.
March 12, 2011

LISP EID Block
draft-meyer-lisp-eid-block-02.txt

Abstract

This is a direction to IANA to allocate a /16 IPv6 prefix for use with the Locator/ID Separation Protocol (LISP).

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on September 13, 2011.

Copyright Notice

Table of Contents

1. 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].

2. Introduction

This memo directs the IANA to allocate a /16 IPv6 prefix for use with the Locator/ID Separation Protocol (LISP - [I-D.ietf-lisp]), LISP Map Server ([I-D.ietf-lisp-ms]), LISP Alternative Topology (LISP+ALT - [I-D.ietf-lisp-alt]) (or other) mapping system, and LISP Interworking ([I-D.ietf-lisp-interworking]).

This block will be used as global Endpoint IDentifier (EID) space (Section 3).

3. Definition of Terms

LISP operates on two name spaces and introduces several new network elements. This section provides high-level definitions of the LISP name spaces and network elements.

Legacy Internet: The portion of the Internet which does not run LISP and does not participate in LISP+ALT or any other mapping system.

LISP site: A LISP site is a set of routers in an edge network that are under a single technical administration. LISP routers which reside in the edge network are the demarcation points to separate the edge network from the core network. See [I-D.ietf-lisp] for more details.

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. A packet that is emitted by a system contains EIDs in its headers and LISP headers are prepended only when the packet reaches an Ingress Tunnel Router (ITR) on the data path to the destination EID. The source EID is obtained via existing mechanisms used to set a host's "local" IP address. An EID is allocated to a host from an EID-prefix block associated with the site where the host is located. See [I-D.ietf-lisp] for more details.

EID-prefix: A a power-of-two block of EIDs which are allocated to a site by an address allocation authority. See [I-D.ietf-lisp] for more details.

EID-Prefix Aggregate: A set of EID-prefixes said to be aggregatable in the [RFC4632] sense. That is, an EID-Prefix aggregate is defined to be a single contiguous power-of-two EID-prefix block. Such a block is characterized by a prefix and a length. See [I-D.ietf-lisp] for more details.

Routing LOCator (RLOC): A RLOC is an IPv4 or IPv6 address of an egress tunnel router (ETR). A RLOC is the output of a EID-to-RLOC mapping lookup. An EID maps to one or more RLOCs. Typically, RLOCs are numbered from topologically-aggregatable blocks that are assigned to a site at each point to which it attaches to the global Internet; where the topology is defined by the connectivity of provider networks, RLOCs can be thought of as Provider Aggregatable (PA) addresses. See [I-D.ietf-lisp] for more details.

EID-to-RLOC Mapping: A binding between an EID-Prefix and the RLOC-set that can be used to reach the EID-Prefix. The general term "mapping" always refers to an EID-to-RLOC mapping. See [I-D.ietf-lisp] for more details.

Ingress Tunnel Router (ITR): An Ingress Tunnel Router (ITR) is a router which accepts receives IP packets from site end-systems on one side and sends LISP-encapsulated IP packets toward the Internet on the other side. The router treats the "inner" 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 globally-routable RLOCs in the source address field and the result of the mapping lookup in the destination address field. See [I-D.ietf-lisp] for more details.

Egress Tunnel Router (ETR): An Egress Tunnel Router (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. An ETR router 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. See [I-D.ietf-lisp] for more details.

Proxy ITR (PITR): A Proxy-ITR (PITR) acts like an ITR but does so on behalf of non-LISP sites which send packets to destinations at LISP sites. See [I-D.ietf-lisp-interworking] for more details.

Proxy ETR (PETR): A Proxy-ETR (PETR) acts like an ETR but does so on behalf of LISP sites which send packets to destinations at non-LISP sites. See [I-D.ietf-lisp-interworking] for more details.

Map Server (MS): A network infrastructure component which learns EID-to-RLOC mapping entries from an authoritative source (typically an ETR). A Map-Server publishes these mappings in the distributed mapping system. See [I-D.ietf-lisp-ms] for more details.

Map Resolver (MR): A network infrastructure component which accepts LISP Encapsulated Map-Requests, typically from an ITR, quickly determines whether or not the destination IP address is part of the EID namespace; if it is not, a Negative Map-Reply is immediately returned. Otherwise, the Map-Resolver finds the appropriate EID-to-RLOC mapping by consulting the distributed mapping database system. See [I-D.ietf-lisp-ms] for more details.

The LISP Alternative Logical Topology (ALT): The virtual overlay network made up of tunnels between LISP+ALT Routers. The Border Gateway Protocol (BGP) runs between ALT Routers and is used to carry reachability information for EID-prefixes. The ALT provides a way to forward Map-Requests toward the ETR that "owns" an EID-prefix. See [I-D.ietf-lisp-alt] for more details.

ALT Router: The device on which runs the ALT. The ALT is a static network built using tunnels between ALT Routers. These routers are deployed in a roughly-hierarchical mesh in which routers at each level in the topology are responsible for aggregating EID-Prefixes learned from those logically "below" them and advertising summary prefixes to those logically "above" them. Prefix learning and propagation between ALT Routers is done using BGP. When an ALT Router receives an ALT Datagram, it looks up the destination EID in its forwarding table (composed of EID-Prefix routes it learned from neighboring ALT Routers) and forwards it to the logical next-hop on the overlay network. The primary function of LISP+ALT routers is to provide a lightweight forwarding infrastructure for LISP control-plane messages (Map-Request and Map-Reply), and to transport data packets when the packet has the same destination address in both the inner (encapsulating) destination and outer destination addresses ((i.e., a Data Probe packet). See [I-D.ietf-lisp-alt] for more details.

4. Security Considerations

This document does not introduces new security threats in the LISP architecture.

5. Acknowledgments

Marla Azinger, Chris Morrow, Peter Schoenmaker all made insightful comments on early versions of this draft.

6. IANA Considerations

This document instructs the IANA to allocate a /16 IPv6 prefix for use as the global LISP EID space.

7. Normative References

[I-D.ietf-lisp]
Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol (LISP)", draft-ietf-lisp-10 (work in progress), March 2011.

[I-D.ietf-lisp-alt]
Fuller, V., Farinacci, D., Meyer, D., and D. Lewis, "LISP Alternative Topology (LISP+ALT)", draft-ietf-lisp-alt-06 (work in progress), March 2011.

[I-D.ietf-lisp-interworking]
Lewis, D., Meyer, D., Farinacci, D., and V. Fuller, "Interworking LISP with IPv4 and IPv6", draft-ietf-lisp-interworking-02 (work in progress), March 2011.

[I-D.ietf-lisp-ms]
Fuller, V. and D. Farinacci, "LISP Map Server", draft-ietf-lisp-ms-07 (work in progress), March 2011.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, August 2006.

Authors' Addresses

Luigi Iannone
Deutsche Telekom Laboratories

Email: luigi@net.t-labs.tu-berlin.de

Darrel Lewis
Cisco Systems, Inc.

Email: darlewis@cisco.com

David Meyer
Cisco Systems, Inc.

Email: dmm@cisco.com

Vince Fuller
Cisco Systems, Inc.

Email: vaf@cisco.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 13, 2011

D. Saucez
Universite catholique de Louvain
L. Iannone
TU Berlin - Deutsche Telekom Laboratories AG
O. Bonaventure
Universite catholique de Louvain

March 12, 2011

LISP Security Threats
draft-saucez-lisp-security-03.txt

Abstract

This draft analyzes some of the threats against the security of the Locator/Identifier Separation Protocol and proposes a set of recommendations to mitigate some of the identified security risks.

Table of Contents

1. 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].

2. Introduction

The Locator/ID Separation Protocol (LISP) is defined in [I-D.ietf-lisp]. The present document aims at identifying threats in the current LISP specification. We also propose some recommendations on mechanisms that could improve the security of LISP against off-path attackers. This document builds upon [I-D.bagnulo-lisp-threat].

This document is split in two parts. The first discusses the LISP data-plane and the second the LISP control-plane.

The LISP data-plane consists of LISP packet encapsulation, decapsulation, and forwarding and includes the EID-to-RLOC Cache and EID-to-RLOC Database data structures used to perform these operations.

The LISP control-plane consists in the mapping distribution system, which can be one of the mapping distribution systems proposed so far (e.g., [I-D.ietf-lisp], [I-D.ietf-lisp-alt], [I-D.ietf-lisp-ms], [I-D.meyer-lisp-cons], and [I-D.lear-lisp-nerd]), and the Map-Request, Map-Reply, Map-Register messages.

This document does not consider all the possible uses of LISP as discussed in [I-D.ietf-lisp]. In the current version, the document focuses on LISP unicast, including as well LISP Interworking, and briefly considers the ALT mapping system described in [I-D.ietf-lisp-alt]. Later versions of this document will include a deeper analysis of the ALT mapping system, as well as the analysis of the security issues in multicast LISP ([I-D.ietf-lisp-multicast]), interworking between LISP and the legacy IPv4 and IPv6 Internet ([I-D.ietf-lisp-interworking]), and LISP-MS ([I-D.ietf-lisp-ms]).

Furthermore, here we assume a generic IP service and do not discuss the difference from a security viewpoint between using IPv4 or IPv6.

3. Definition of Terms

The present document does not introduce any new term, compared to the main LISP specification. For a complete list of terms please refer to [I-D.ietf-lisp].

4. On-path Attackers

On-path attackers are attackers that are able to capture and modify all the packets exchanged between an ITR and an ETR. To cope with such an attacker, cryptographic techniques such as those used by IPSec are required. We do not consider that LISP has to cope with such attackers.

Mobile IP has also considered time-shifted attacks from on-path attackers. A time-shifted attack is an attack where the attacker is temporarily on the path between two communicating hosts. While it is on-path, the attacker sends specially crafted packets or modifies packets exchanged by the communicating hosts in order to disturb the packet flow (e.g., by performing a man in the middle attack). An important issue for time-shifted attacks is the duration of the attack once the attacker has left the path between the two communicating hosts. We do not consider time-shifted attacks in this document.

5. Off-Path Attackers: Reference Environment

Throughout this document we consider the reference environment shown in the figure below. There are two hosts attached to LISP routers: HA and HB. HA is attached to the two LISP xTRs LR1 and LR2, which are attached to two different ISPs. HB is attached to the two LISP xTRs LR3 and LR4. HA and HB are the EIDs of the two hosts. LR1, LR2, LR3, and LR4 are the RLOCs of the xTRs.

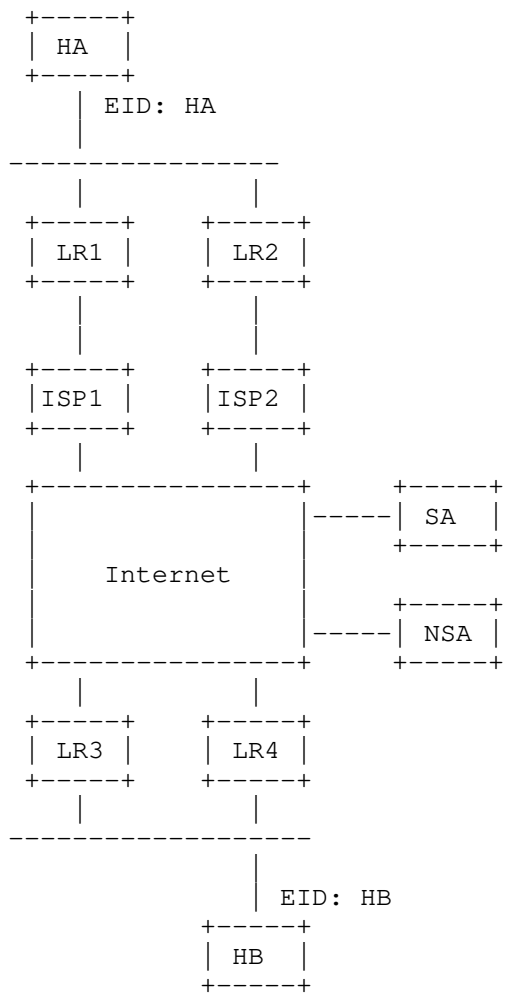


Figure 1: Reference Network

We consider two off-path attackers with different capabilities:

SA is an off-path attacker that is able to send spoofed packets, i.e., packets with a different source IP address than its assigned IP address.

NSA is an off-path attacker that is only able to send packets whose source address is its assigned IP address.

It should be noted that with LISP, packet spoofing is slightly different than in the current Internet. Generally the term "spoofed

packet" indicates a packet containing a source IP address which is not the one of the actual originator of the packet. Since LISP uses encapsulation, the spoofed address can be in the outer header as well as in the inner header, this translates in two types of spoofing:

EID Spoofing: the originator of the packet puts in it a spoofed EID. The packet will be normally encapsulated by the ITR of the site.

RLOC Spoofing: the originator of the packet generates directly a LISP-encapsulated packet with a spoofed source RLOC.

Note that the two types of spoofing are not mutually exclusive, rather all combinations are possible and can be used to perform different kind of attacks.

In our reference environment, both SA and NSA attackers are capable of sending LISP encapsulated data packets and LISP control packets. This means that SA is able to perform both RLOC and EID spoofing while NSA can only perform EID spoofing. They may also send other types of IP packets such as ICMP messages. We assume that both attackers can query the LISP mapping system to obtain the mappings for both HA and HB.

6. Data-Plane Threats

This section discusses threats and attacks related to the LISP data-plane. More precisely, we discuss the operations of encapsulation, decapsulation, and forwarding as well as the content of the EID-to-RLOC Cache and EID-to-RLOC Database as specified in the original LISP document ([I-D.ietf-lisp]).

We start considering the two main data structures of LISP, namely the EID-to-RLOC Database and the EID-to-RLOC Cache. Then, we look at the data plane attacks that can be performed by a spoofing off-path attacker (SA) and discuss how they can be mitigated by the LISP xTRs. In this analysis, we assume that the LR1 and LR2 (resp. LR3 and LR4) xTRs maintain a EID-to-RLOC Cache that contains the required mapping entries to allow HA and HB to exchange packets.

6.1. EID-to-RLOC Database Threats

The EID-to-RLOC Database on each xTR maintains the set of mappings related to the EID-Prefixes that are "behind" the xTR. Where "behind" means that at least one of the xTR's globally-visible IP addresses is a RLOC for those EID-Prefixes.

As described in [I-D.ietf-lisp], the EID-to-RLOC Database content is determined by configuration. This means that the only way to attack this data structure is by gaining privileged access to the xTR. As such, it is out of the scope of LISP to propose any mechanism to protect routers and, hence, it is no further analyzed in this document.

6.2. EID-to-RLOC Cache Threats

A key component of the overall LISP architecture is the EID-to-RLOC Cache. The EID-to-RLOC Cache is the data structure that stores the bindings between EID and RLOC (namely the "mappings") to be used later on. Attacks against this data structure can happen either when the mappings are first installed in the cache (see also Section 7) or by corrupting (poisoning) the mappings already present in the cache.

6.2.1. EID-to-RLOC Cache poisoning

The content of the EID-to-RLOC Cache can be poisoned by spoofing LISP encapsulated packets. Example of EID-to-RLOC Cache poisoning are:

Fake mapping: The cache contains entirely fake mappings that do not originate from an authoritative mapping server. This can be achieved either through gleaning as described in Section 7.3 or by attacking the control-plane as described in Section 7.

EID Poisoning: The EID-Prefix in a specific mapping is not owned by the originator of the entry. Similarly to the previous case, this can be achieved either through gleaning as described in Section 7.3 or by attacking the control-plane as described in Section 7.

EID redirection/RLOC poisoning: The EID-Prefix in the mapping is not bound to (located by) the set of RLOCs present in the mapping. This can result in packets being redirected elsewhere, eavesdropped, or even blackholed. Note that not necessarily all RLOCs are fake/spoofed. The attack works also if only part of the RLOCs, the highest priority ones, are compromised. Again, this can be achieved either through the gleaning as described in Section 7.3 or by attacking the control-plane as described in Section 7.

Reachability poisoning: The reachability information stored in the mapping could be poisoned, redirecting the packets to a subset of the RLOCs (or even stopping it if locator status bits are all set to 0). If reachability information is not verified through the control-plane this attack can be simply achieved by sending a spoofed packet with swapped or all locator status

bits reset. The same result can be obtained by attacking the control-plane as described in Section 7. Depending on how the RLOC reachability information is stored on the router, the attack can impact only one mapping or all the mappings that share the same RLOC.

Traffic Engineering information poisoning: The LISP protocol defines two attributes associated to each RLOC in order to perform inbound Traffic Engineering: namely priority and weight. By injecting fake TE attributes, the attacker is able to break load balancing policies and concentrate all the traffic on a single RLOC or put more load on a RLOC than what is expected, creating congestion. It is even possible to block the traffic if all the priorities are set to 255. Corrupting the TE attributes can be achieved by attacking the control-plane as described in Section 7.

Mapping TTL poisoning: The LISP protocol associates a Time-To-Live to each mapping that, once expired, allows to delete a mapping from the EID-to-RLOC Cache (or forces a Map-Request/Map-Reply exchange to refresh it if still needed). By injecting fake TTL values, an attacker can either shrink the EID-to-RLOC Cache (using very short TTL), thus creating an excess of cache miss causing a DoS on the mapping system, or it can increase the size of the cache by putting very high TTL values, up to a cache overflow (see Section 6.2.2). Corrupting the TTL can be achieved by attacking the control-plane as described in Section 7. Long TTL can be use in fake mappings to increase an attack duration.

Instance ID poisoning: The LISP protocol allows to use a 24-bit identifier to select the forwarding table to use on the decapsulating ETR to forward the decapsulated packet. By spoofing this attribute the attacker is able to redirect or blackhole inbound traffic. Corrupting the Instance ID attribute can be achieved by attacking the control-plane as described in Section 7.

Map-Version poisoning: The LISP protocol allows to associate a version number to mappings ([I-D.ietf-lisp-map-versioning]). The LISP header can transport source and destination map-versions, describing which version of the mapping have been used to select the source and the destination RLOCs of the LISP encapsulated packet. By spoofing this attribute the attacker is able to trigger Map-Request on the receiving ETR. Corrupting the Map-Version attribute can be achieved either by attacking the control-plane as described in Section 7 or by using spoofed packets as described in Section 6.4.2.

If the above listed attacks succeed, the attacker has the means of controlling the traffic.

6.2.2. EID-to-RLOC Cache overflow

Depending on how the EID-to-RLOC Cache is managed (e.g., LRU vs. LFU) and depending on its size, an attacker can try to fill the cache with fake mappings. Once the cache is full, some mappings will be replaced by new fake ones, causing traffic disruption.

This can be achieved either through the gleaning as described in Section 7.3 or by attacking the control-plane as described in Section 7.

Another way to generate a EID-to-RLOC Cache overflow is by injecting mapping with a fake and very large TTL value. In this case the cache will keep a large amount of mappings ending with a completely full cache. This type of attack can also be performed through the control-plane.

6.3. Attacks not leveraging on the LISP header

We first consider an attacker that sends packets without exploiting the LISP header, i.e., with the N, L, E, V, and I bits reset ([I-D.ietf-lisp]).

To inject a packet in the HA-HB flow, a spoofing off-path attacker (SA) can send a LISP encapsulated packet whose source is set to LR1 or LR2 and destination LR3 or LR4. The packet will reach HB as if the packet was sent by host HA. This is not different from today's Internet where a spoofing off-path attacker may inject data packets in any flow. Several existing techniques can be used by hosts to prevent such attacks from affecting established flows, e.g., [RFC4301] and [I-D.ietf-tcpm-tcp-security] .

On the other hand, a non-spoofing off-path attacker (NSA) can only send a packet whose source address is set to its assigned IP address. The destination address of the encapsulated packet can be LR3 or LR4. When the LISP ETR that serves HB receives the encapsulated packet, it can consult its EID-to-RLOC Cache and verify that NSA is not a valid source address for LISP encapsulated packets containing a packet sent by HA. This verification is only possible if the ETR already has a valid mapping for HA. Otherwise, and to avoid such data packet injection attacks, the LISP ETR should reject the packet and possibly query the mapping system to obtain a mapping for the encapsulated source EID (HA).

6.4. Attacks leveraging on the LISP header

The latest LISP draft [I-D.ietf-lisp] defines several flags that modify the interpretation of the LISP header in data packets. In this section, we discuss how an off-path attacker could exploit this LISP header.

6.4.1. Attacks using the Locator Status Bits

When the L bit is set to 1, it indicates that the second 32-bits longword of the LISP header contains the Locator Status Bits. In this field, each bit position reflects the status of one of the RLOCs mapped to the source EID found in the encapsulated packet. In particular, a packet with the L bit set and all Locator Status Bits set to zero indicates that none of the locators of the encapsulated source EID are reachable. The reaction of a LISP ETR that receives such a packet is not clearly described in [I-D.ietf-lisp].

A spoofing off-path attacker (SA) can send a data packet with the L bit set to 1, all Locator Status Bits set to zero, a spoofed source RLOC (e.g. LR1), destination LR3, and containing an encapsulated packet whose source is HA. If LR3 blindly trust the Locator Status Bits of the received packet it will set LR1 and LR2 as unreachable, possibly disrupting ongoing communication.

Locator Status Bits can be blindly trusted only in secure environments. In the general unsecured Internet environment, the safest practice for xTRs is to confirm the reachability change through the mapping system. In the above example, LR3 should note that something as changed in the Locator Status Bits and query the mapping system in order to confirm status of the RLOCs of the source EID.

A similar attack could occur by setting only one Locator Status Bit to 1, e.g., the one that corresponds to the source RLOC of the packet.

If a non-spoofing off-path attacker (NSA) sends a data packet with the L bit set to 1 and all Locator Status Bits set to zero, this packet will contain the source address of the attacker. Similarly as in Section 6.3, if the xTR accepts the packet without checking the EID-to-RLOC Cache for a mapping that binds the source EID and the source RLOC of the received packet, then the same observation like for the the spoofing attacker (SA) apply.

Otherwise, if the xTR does make the check through the EID-to-RLOC Cache, it should reject the packet because its source address is not one of the addresses listed as RLOCs for the source EID.

Nevertheless, in this case a Map-Request should be sent, which can be used to perform Denial of Service attacks. Indeed an attacker can frequently change the Locator Status Bits in order to trigger a large amount of Map-Requests. Rate limitation, as described in [I-D.ietf-lisp], does not allow to send high number of such a request, resulting in the attacker saturating the rate with these spoofed packets.

6.4.2. Attacks using the Map-Version bit

The Map-Version bit is used to indicate whether the low-order 24 bits of the first 32 bits word of the LISP header contain an Source and Destination Map-Version. When a LISP ETR receives a LISP encapsulated packet with the Map-Version bit set to 1, the following actions are taken:

- It compares the Destination Map-Version found in the header with the current version of its own mapping, in the EID-to-RLOC Database, for the destination EID found in the encapsulated packet. If the received Destination Map-Version is smaller (i.e., older) than the current version, the ETR should apply the SMR procedure described in [I-D.ietf-lisp] and send a Map-Request with the SMR bit set.

- If a mapping exists in the EID-to-RLOC Cache for the source EID, then it compares the Map-Version of that entry with the Source Map-Version found in the header of the packet. If the stored mapping is older (i.e., the Map-Version is smaller) than the source version of the LISP encapsulated packet, the xTR should send a Map-Request for the source EID.

A spoofing off-path attacker (SA) could use the Map-Version bit to force an ETR to send Map-Request messages. The attacker can retrieve the current source and destination Map-Version for both HA and HB. Based on this information, it can send a spoofed packet with an older Source Map-Version or Destination Map-Version. If the size of the Map-Request message is larger than the size of the smallest LISP-encapsulated packet that could trigger such a message, this could lead to amplification attacks (see Section 7.1). Fortunately, [I-D.ietf-lisp] recommends to rate limit the Map-Request messages that are sent by an xTR. This prevents the amplification attack, but there is a risk of Denial of Service attack if an attacker sends packets with Source and Destination Map-Versions that frequently change. In this case, the ETR could consume all its rate by sending Map-Request messages in response to these spoofed packets.

A non-spoofing off-path attacker (NSA) cannot success in such an attack if the destination xTR rejects the LISP encapsulated packets

that are not sent by one of the RLOCs mapped to the included source EID. If it is not the case, the attacker can be able to perform attacks concerning the Destination Map Version number as for the spoofing off-path attacker (SA).

6.4.3. Attacks using the Nonce-Present and the Echo-Nonce bits

The Nonce-Present and Echo-Nonce bits are used when verifying the reachability of a remote ETR. Assume that LR3 wants to verify that LR1 receives the packets that it sends. LR3 can set the Echo-Nonce and the Nonce-Present bits in LISP data encapsulated packets and include a random nonce in these packets. Upon reception of this packet, LR1 will store the nonce sent by LR3 and echo it when it returns LISP encapsulated data packets to LR3.

A spoofing off-path attacker (SA) could interfere with this reachability test by sending two different types of packets:

1. LISP data encapsulated packets with the Nonce-Present bit set and a random nonce and the appropriate source and destination RLOCs.

2. LISP data encapsulated packets with the Nonce-Present and the Echo-Nonce bits both set and the appropriate source and destination RLOCs. These packets will force the receiving ETR to store the received nonce and echo it in the LISP encapsulated packets that it sends.

The first type of packet should not cause any major problem to ITRs. As the reachability test uses a 24 bits nonce, it is unlikely that an off-path attacker could send a packet that causes an ITR to believe that the ETR it is testing is reachable while in reality it is not reachable.

The second type of packet could be exploited to create a Denial of Service attack against the nonce-based reachability test. Consider a spoofing off-path attacker (SA) that sends a continuous flow of spoofed LISP data encapsulated packets that contain the Nonce-Present and the Echo-Nonce bit and each packet contains a different random nonce. The ETR that receives such packets will continuously change the nonce that it returns to the remote ITR. If the remote ITR starts a nonce-reachability test, this test may fail because the ETR has received a spoofed LISP data encapsulated packet with a different random nonce and never echoes the real nonce. In this case the ITR will consider the ETR not reachable. The success of this test will of course depend on the ratio between the amount of packets sent by the legitimate ITR and the spoofing off-path attacker (SA).

Packets sent by a non-spoofing off-path attacker (NSA) can cause

similar problem if no check is done with the EID-to-RLOC Cache (see Section 6.3 for the EID-to-RLOC Cache check). Otherwise, if the check is performed the packets will be rejected by the ETR that receives them and cannot cause problems.

7. Control Plane Threats

In this section, we discuss the different types of attacks that can occur when an off-path attacker sends control plane packets. We focus on the packets that are sent directly to the ETR and do not analyze the particularities of a LISP mapping system. The ALT mapping system is discussed in Section 10.

7.1. Attacks with Map-Request messages

An off-path attacker could send Map-Request packets to a victim ETR. In theory, a Map-Request packet is only used to solicit an answer and as such it should not lead to security problems. However, the LISP specification [I-D.ietf-lisp] contains several particularities that could be exploited by an off-path attacker.

The first possible exploitation is the P bit. The P bit is used to probe the reachability of remote ETRs in the control plane. In our reference environment, LR3 could probe the reachability of LR1 by sending a Map-Request with the P bit set. LR1 would reply by sending a Map-Reply message with the P bit set and the same nonce as in the Map-Request message.

A spoofing off-path attacker (SA) could use the P bit to force a victim ETR to send a Map-Reply to the spoofed source address of the Map-Request message. As the Map-Reply can be larger than the Map-Request message, there is a risk of amplification attack. Considering only IPv6 addresses, a Map-Request can be as small as 40 bytes, considering one single ITR address and no Mapping Protocol Data. The Map-Reply instead has a size of O(12 + (R * (28 + N * 24))) bytes, where N is the maximum number of RLOCs in a mapping and R the maximum number of records in a Map-Reply. Since up to 255 RLOCs can be associated to an EID-Prefix and 255 records can be stored in a Map-Reply, the maximum size of a Map-Reply is thus above 1 MB showing a size factor of up to 39,193 between the message sent by the attacker and the message sent by the ETR. These numbers are however theoretical values not considering transport layer limitations and it is more likely that the reply will contain only on record with at most a dozen of locators, giving an amplification factor around 8.

Any ISP with a large number of potential RLOCs for a given EID-Prefix

should carefully ponder the best trade-off between the number of RLOCs through which it wants that the EID is reachable and the consequences that an amplification attack can produce.

It should be noted that the maximum rate of Map-Reply messages should apply to all Map-Replies and also be associated to each destination that receives Map-Reply messages. Otherwise, a possible amplification attack could be launched by a spoofing off-path attacker (SA) as follows. Consider an attacker SA and and EID-Prefix p/P and a victim ITR. To amplify a Denial of Service attack against the victim ITR, SA could send spoofed Map-Request messages whose source EID addresses are all the addresses inside p/P and source RLOC address is the victim ITR. Upon reception of these Map-Request messages, the ETR would send large Map-Reply messages for each of the addresses inside p/P back to the victim ITR.

If a non-spoofing off-path attacker (NSA) sends a Map-Request with the P bit set, it will receive a Map-Reply with the P bit set. This does not raise security issues besides the usual risk of overloading a victim ETR by sending too many Map-Request messages.

The Map-Request message may also contain the SMR bit. Upon reception of a Map-Request message with the SMR bit, an ETR must return to the source of the Map-Request message a Map-Request message to retrieve the corresponding mapping. This raises similar problems as the P bit discussed above except that as the Map-Request messages are smaller than Map-Reply messages, the risk of amplification attacks is reduced. This is not true anymore if the ETR append to the Map-Request messages its own Map-Records. This mechanism is meant to reduce the delay in mapping distribution since mapping information is provided in the Map-Request message.

Furthermore, appending Map-Records to Map-Request messages represents a major security risk since an off-path attacker could generate a (spoofed or not) Map-Request message and include in the Map-Reply portion of the message mapping for EID prefixes that it does not serve. This could lead to various types of redirection and denial of service attacks. An xTR should not process the Map-Records information that it receives in a Map-Request message.

7.2. Attacks with Map-Reply messages

In this section we analyze the attacks that could occur when an off-path attacker sends directly Map-Reply messages to ETRs without using one of the proposed LISP mapping systems.

There are two different types of Map-Reply messages:

Positive Map-Reply: This messages contain a Map-Record binding an EID-Prefix to one or more RLOCs.

Negative Map-Reply: This messages contain a Map-Record for an EID-Prefix with an empty locator-set and specifying an action, which may be either Drop, Natively forward, or Send Map-Request.

Positive Map-Reply messages are used to map EID-Prefixes onto RLOCs. Negative Map-Reply messages are used to support PTR and interconnect the LISP Internet with the legacy Internet.

Most of the security of the Map-Reply messages depend on the 64 bits nonce that is included in a Map-Request and returned in the Map-Reply. An ETR must never accept a Map-Request message whose nonce does not match one of the pending Map-Request messages. If an ETR does not accept Map-Reply messages with an invalid nonce, the risk of attack is very small given the size of the nonce (64 bits).

Note, however, that the nonce only confirms that the Map-Reply was sent by the ETR that received the Map-Request. It does not validate the content of the Map-Reply message.

7.3. Gleaning Attacks

A third type of attack involves the gleaning mechanism proposed in [I-D.ietf-lisp] and discussed in [Saucez09]. In order to reduce the time required to obtain a mapping, [I-D.ietf-lisp] allows an ITR to learn a mapping from the LISP data encapsulated packets and the Map-Request packets that it receives. LISP data encapsulated packet contains a source RLOC, destination RLOC, source EID and destination EID. When a ITR receives a data encapsulated packet coming from a source EID for which it does not already know a mapping, it may insert the mapping between the source RLOC and the source EID in its EID-to-RLOC Cache. Gleaning can also be used when an ITR receives a Map-Request as the Map-Request also contains a source EID address and a source RLOC. Once a gleaned entry has been added to the cache, the LISP ITR sends a Map-Request to retrieve the mapping for the gleaned EID from the mapping system. [I-D.ietf-lisp] recommends to store the gleaned entries for only a few seconds.

The first risk of gleaning is the ability to temporarily hijack an identity. Consider an off-path attacker that wants to temporarily hijack host HA's identity and send packets to host HB with host HA's identity. If the xTRs that serve host HB do not store a mapping for host HA, a non-spoofing off-path attacker (NSA) could send a LISP encapsulated data packet to LR3 or LR4. The ETR will store the gleaned entry and use it to return the packets sent by host HB to the

attacker. In parallel, the ETR will send a Map-Request to retrieve the mapping for HA. During a few seconds or until the reception of the Map-Reply, host HB will exchange packets with the attacker that has hijacked HA's identity. Note that the attacker could in parallel send lots of Map-Requests or lots of LISP data encapsulated packets with random sources to force the xTR that is responsible for host HA to send lots of Map-Request messages in order to force it to exceed its rate limit for control plane messages. This could further delay the arrival of the Map-Reply message on the requesting ETR.

Gleaning also introduces the possibility of a man-in-the-middle attack. Consider an off-path attacker that knows that hosts HA and HB that reside in different sites will exchange information at time t. An off-path attacker could use this knowledge to launch a man-in-the-middle attack if the xTRs that serve the two hosts do not have mapping for the other EID. For this, the attacker sends to LR1 (resp. LR3) a LISP data encapsulated packet whose source RLOC is its IP address and contains an IP packet whose source is set to HB (resp. HA). The attacker chooses a packet that will not trigger an answer, for example the last part of a fragmented packet. Upon reception of these packets, LR1 and LR3 install gleaned entries that point to the attacker. As explained above, the attacker could, at the same time, send lots of packets to LR1 and LR3 to force them to exhaust their control plane rate limit. This will extend the duration of the gleaned entry. If host HA establishes a flow with host HB at that time, the packets that they exchange will first pass through the attacker.

In both cases, the attack only lasts for a few seconds (unless the attacker is able to exhaust the rate limitation). However it should be noted that today a large amount of packets may be exchanged during even a small fraction of time.

8. Threats concerning Interworking

[I-D.ietf-lisp-interworking] defines two network elements to allow LISP and non-LISP sites to communicate, namely the Proxy-ITR and the Proxy-ETR. The Proxy-ITR encapsulates traffic from non-LISP sites in order to forward it toward LISP sites, while the Proxy-ETR decapsulates traffic arriving from LISP sites in order to forward it toward non-LISP sites. For these elements some of the attack based on the LISP specific header are not possible, for the simple reason that some of the fields cannot be used due to the unidirectional nature of the traffic.

The Proxy-ITR has functionalities similar to the ITR, however, its main purpose is to encapsulate packets arriving from the DFZ in order

to reach LISP sites. This means that it is no bound to any particular EID-Prefix, hence no mapping exists and no mapping can be configured in the EID-to-RLOC Database. This means that the Proxy-ITR element itself is not able, to check whether or not the arriving traffic has the right to be encapsulated or not. To limit such an issue it is recommended to use the current practice based on firewalls and ACLs on the machine running the Proxy-ITR service. On the other side, the Proxy-ITR is meant to encapsulate only packets that are destined to one of the LISP sites it is serving. This is the case for instance for a service provider selling Proxy-ITR services. For this purpose a static EID-to-RLOC Cache can be configured in order to encapsulate only valid packets. In case of a cache-miss no Map-Request needs to be sent and the packet can be silently dropped.

The Proxy-ETR has functionalities similar to the ETR, however, its main purpose is to inject un-encapsulated packet in the DFZ in order to reach non-LISP-Sites. This means that since there is no specific EID-Prefix downstream, it has no EID-to-RLOC Database that can be used to check whether or not the destination EID is part of its domain. In order to avoid for the Proxy-ETR to be used as relay in a DoS attack it is preferable to configure the EID-to-RLOC Cache with static entries used to check if an encapsulated packet coming from a specific RLOC and having a specific source EID is actually allowed to transit through the Proxy-ETR. This is also important for services provider selling Proxy-ETR service to actually process only packets arriving from its customers. However, in case of cache-miss no Map-Request needs to be sent, rather the packet can be silently dropped since it is not originating from a valid site. The same drop policy should be used for packets with an invalid source RLOC or a valid source RLOC but an invalid EID.

9. Threats with Malicious xTRs

In this section, we discuss the threats that could be caused by malicious xTRs. We consider the reference environment below where EL1 is a malicious or compromised xTR. This malicious xTR serves a set of hosts that includes HC. The other xTR and hosts in this network play the same role as in the reference environment described in Section 5.

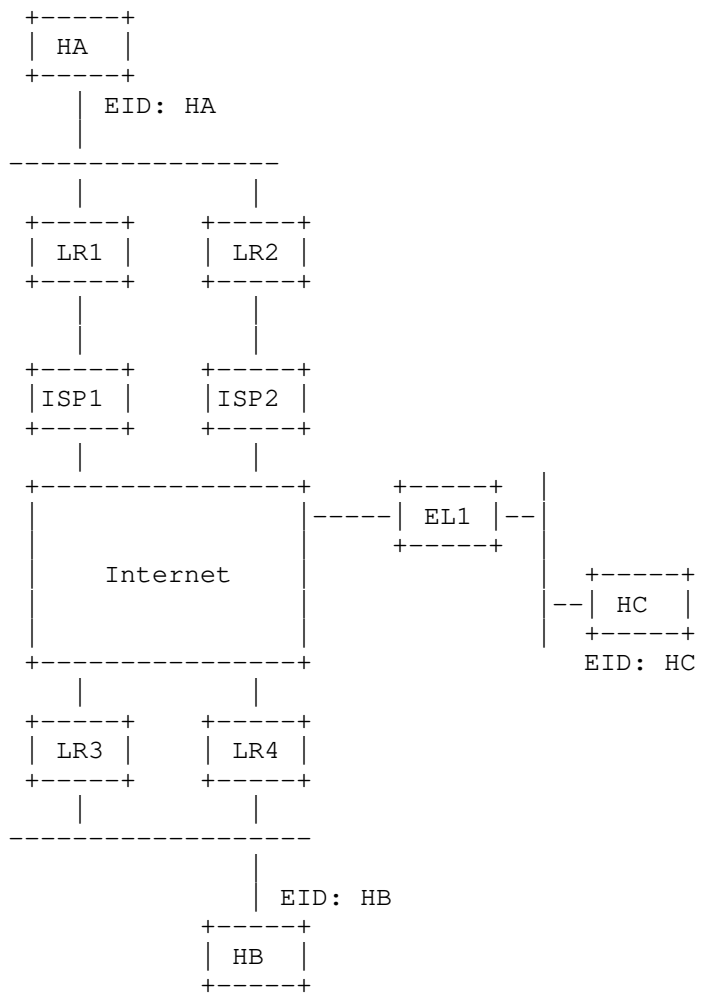


Figure 2: Malicious xTRs' Reference Environment

Malicious xTRs are probably the most serious threat to the LISP control plane from a security viewpoint. To understand the problem, let us consider the following scenario. Host HC and HB exchange packets with host HA. As all these hosts reside in LISP sites, LR1 and LR2 store mappings for HB and HC. Thus, these xTRs may need to exchange LISP control plane packets with EL1, e.g., to perform reachability tests or to refresh expired mappings (e.g., if HC's mapping has a small TTL).

A first threat against the LISP control plane is when EL1 replies to a legitimate Map-Request message sent by LR1 or LR2 with a Map-Reply

message that contains an EID-Prefix that is larger than the prefix owned by the site attached to EL1. This could allow EL1 to attract packets destined to other EIDs than the EIDs that are attached to EL1.

Another possible attack is a Denial of Service attack by sending a Negative Map-Reply message for a coarser prefix without any locator and with the Drop action. Such a Negative Map-Reply indicates that the ETR that receives it should discard all packets. The current LISP specification briefly discusses this problem [I-D.ietf-lisp], but the proposed solutions does not solve the problem.

Another concern with malicious xTRs is the possibility of Denial of Service attacks. A first attack is the flooding attack that was described in [I-D.bagnulo-lisp-threat]. This attack allows a malicious xTR to redirect traffic to a victim. The malicious xTR first defines a mapping for HC with two RLOCs: its own RLOC (EL1) and the RLOC of the victim (e.g., LR3). The victim's RLOC is set as unreachable in the mapping. HC starts a large download from host HA. Once the download starts, the malicious xTR updates its Locator Status Bits, changes the mapping's version number or sets the SMR bit such that LR1 updates its EID-to-RLOC Cache to send all packets destined to HC to the victim's RLOC. Instead of downloading from HA, the attacker could also send packets that trigger a response (e.g., ICMP, TCP SYN, DNS request, ...) to HA. HA would then send its response and its xTR would forward it to the victim's RLOC.

An important point to note about this flooding attack is that it reveals a potential problem in the LISP architecture. A LISP ITR relies on the received mapping and possible reachability information to select the RLOC of the ETR that it uses to reach a given EID or block of EIDs. However, if the ITR made a mistake, e.g., due to configuration, implementation or other types of errors and has chosen a RLOC that does not serve the destination EID, there is no easy way for the LISP ETR to inform the ITR of its mistake. A possible solution could be to force a ETR to perform a reachability test with the selected ITR as soon as it selects it. This will be analyzed in the next version of this document.

10. Security of the ALT Mapping System

One of the assumptions in [I-D.ietf-lisp] is that the mapping system is more secure than sending Map-Request and Map-Reply messages directly. We analyze this assumption in this section by analyzing the security of the ALT mapping system.

The ALT mapping system is basically a manually configured overlay of

GRE tunnels between ALT routers. BGP sessions are established between ALT routers that are connected through such a tunnel. An ALT router advertises the EID prefixes that it serves over its BGP sessions with neighboring ALT routers and the EID-Prefixes that it has learned from neighboring ALT routers.

The ALT mapping system is in fact a discovery system that allows any ALT router to discover the ALT router that is responsible for a given EID-Prefix. To obtain a mapping from the ALT system, an ITR sends a packet containing a Map-Request on the overlay. This Map-Request is sent inside a packet whose destination is the requested EID. The Map-Request is routed on the overlay until it reaches the ALT router that advertised initially the prefix that contains the requested EID. This ALT router then replies directly by sending a Map-Reply to the RLOC of the requesting ITR.

The security of the ALT mapping system depends on many factors, including:

- The security of the intermediate ALT routers.

- The validity of the BGP advertisements sent on the ALT overlay.

Unfortunately, experience with BGP on the global Internet has shown that BGP is subject to various types of misconfiguration problems and security attacks. The SIDR working group is developing a more secure inter-domain routing architecture to solve this problem ([I-D.ietf-sidr-arch]).

The security of the intermediate ALT routers is another concern. A malicious intermediate ALT router could manipulate the received BGP advertisements and also answer to received Map-Requests without forwarding them to their final destination on the overlay. This could lead to various types of redirection attacks. Note that in contrast with a regular IP router that could also manipulate in transit packets, when a malicious or compromised ALT router replies to a Map-Request, it can redirect legitimate traffic for a long period of time by sending an invalid Map-Reply message. Thus, the impact of a malicious ALT router could be much more severe than a malicious router in today's Internet.

11. Suggested Recommendations

To mitigate the impact of attacks against LISP, the following recommendations should be followed.

First, the use of some form of filtering can help in avoid or at

least mitigate some types of attacks.

- On ETRs, packets should be decapsulated only if the destination EID is effectively part of the EID-Prefix downstream the ETR. Further, still on ETRs, packets should be decapsulated only if a mapping for the source EID is present in the EID-to-RLOC Cache and has been obtained through the mapping system (not gleaned).

- On ITRs, packets should be encapsulated only if the source EID is effectively part of the EID-Prefix downstream the ITR. Further, still on ITRs, packets should be encapsulated only if a mapping obtained from the mapping system is present in the EID-to-RLOC Cache (no Data-Probing).

Note that this filtering, since complete mappings need to be installed in both ITRs and ETRs, can introduce a higher connection setup latency and hence potentially more packets drops due to the lack of mappings in the EID-to-RLOC Cache.

While the gleaning mechanism allows to start encapsulating packets to a certain EID in parallel with the Map-Request to obtain a mapping when a new flow is established, it creates important security risks since it allows attackers to perform identity hijacks. Although the duration of these identity hijacks is limited (except the case of rate limitation exhaustion), their impact can be severe. A first option would be to disable gleaning until the security concerns are solved. A second option would be to strictly limit the number of packets that can be forwarded via a gleaned entry. Overall the benefits of gleaning, i.e., avoiding the loss of the first packet of a flow, seems very small compared to the associated security risks. Furthermore, measurements performed in data centers show that today's Internet often operate with packet loss ratio of 1 or 2 percentage ([Chu]). These packet loss ratio are probably already orders of magnitude larger than the improvement provided by the gleaning mechanism.

With the increasing deployment of spoofing prevention techniques such as [RFC3704] or SAVI [SAVI], it can be expected that attackers will become less capable of sending packets with a spoofed source address. To prevent packet injection attacks from non-spoofing attackers (NSA), ETRs should always verify that the source RLOC of each received LISP data encapsulated packet corresponds to one of the RLOCs listed in the mappings for the source EID found in the inner packet. An alternative could be to use existing IPSec techniques [RFC4301]and when necessary including perhaps [RFC5386] to establish an authenticated tunnel between the ITR and the ETR.

[I-D.ietf-lisp] recommends to rate limit the control messages that

are sent by a xTR. This limit is important to deal with denial of service attacks. However, a strict limit, e.g., implemented with a token bucket, on all the Map-Request and Map-Reply messages sent by a xTR is not sufficient. A xTR should distinguish between different types of control plane packets:

1. The Map-Request messages that it sends to refresh expired mapping information.

2. The Map-Request messages that it sends to obtain mapping information because one of the served hosts tried to contact an external EID.

3. The Map-Request messages that it sends as reachability probes.

4. The Map-Reply messages that it sends as response to reachability probes.

5. The Map-Request messages that it sends to support gleaning.

These control plane messages are used for different purposes. Fixing a global rate limit for all control plane messages increases the risk of Denial of Service attacks if a single type of control plane message can exceed the configured limit. This risk could be mitigated by either specifying a rate for each of the five types of control plane messages. Another option could be to define a maximum rate for all control plane messages, and prioritize the control plane messages according to the list above (with the highest priority for message type 1).

In [I-D.ietf-lisp], there is no mechanism that allows a xTR to verify the validity of the content a Map-Reply message that it receives. Besides the attacks discussed earlier in the document, a time-shifted attack where an attacker is able to modify the content of a Map-Reply message but then needs to move off-path could also create redirection attacks. The nonce only allows a xTR to verify that a Map-Reply responds to a previously sent Map-Request message. The LISP Working Group should explore solutions that allow to verify the validity and integrity of bindings between EID-Prefixes and their RLOCS (e.g., [I-D.saucez-lisp-mapping-security] and [I-D.maino-lisp-sec]). Having such kind of mechanism would allow ITRs to ignore non-verified mappings, thus increasing security.

LISP Working Group should consider developing secure mechanisms to allow an ETR to indicate to an ITR that it does not serve a particular EID or block of EIDs in order to mitigate the flooding attacks.

Finally, there is also the risk of Denial of Service attack against the EID-to-RLOC Cache. We have discussed these attacks when considering external attackers with, e.g., the gleaning mechanism and in Section 6.2. If an ITR has a limited EID-to-RLOC Cache, a malicious or compromised host residing in the site that it serves could generate packets to random destinations to force the ITR to issue a large number of Map-Requests whose answers could fill its cache. Faced with such misbehaving hosts, LISP ITR should be able to limit the percent of Map-Requests that it sends for a given source EID.

12. Document Status and Plans

In this document, we have analyzed some of the security threats that affect the Locator/Identifier Separation Protocol (LISP). We have focused our analysis on unicast traffic and considered both the LISP data and control planes, and provided some recommendations to improve the security of LISP.

Revisions of this document will document the security threats of other parts of the LISP architecture, including but not limited to:

- Instance ID attribute.
- LISP Multicast.
- LISP Map-Server.

13. IANA Considerations

This document makes no request to IANA.

14. Security Considerations

Security considerations are the core of this document and do not need to be further discussed in this section.

15. Acknowledgments

The flooding attack and the reference environment were first described in Marcelo Bagnulo's draft [I-D.bagnulo-lisp-threat].

This work has been partially supported by the INFSO-ICT-216372 TRILOGY Project (www.trilogy-project.org).

16. References

16.1. Normative References

[I-D.ietf-lisp]
Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol (LISP)", draft-ietf-lisp-10 (work in progress), March 2011.

[I-D.ietf-lisp-alt]
Fuller, V., Farinacci, D., Meyer, D., and D. Lewis, "LISP Alternative Topology (LISP+ALT)", draft-ietf-lisp-alt-06 (work in progress), March 2011.

[I-D.ietf-lisp-interworking]
Lewis, D., Meyer, D., Farinacci, D., and V. Fuller, "Interworking LISP with IPv4 and IPv6", draft-ietf-lisp-interworking-02 (work in progress), March 2011.

[I-D.ietf-lisp-map-versioning]
Iannone, L., Saucez, D., and O. Bonaventure, "LISP Map-Versioning", draft-ietf-lisp-map-versioning-01 (work in progress), March 2011.

[I-D.ietf-lisp-ms]
Fuller, V. and D. Farinacci, "LISP Map Server", draft-ietf-lisp-ms-07 (work in progress), March 2011.

[I-D.ietf-lisp-multicast]
Farinacci, D., Meyer, D., Zwiebel, J., and S. Venaas, "LISP for Multicast Environments", draft-ietf-lisp-multicast-04 (work in progress), October 2010.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

16.2. Informative References

[Chu] Jerry Chu, H., "Tuning TCP Parameters for the 21st Century", 75th IETF, Stockholm, July 2009, <http://tools.ietf.org/wg/savi/>.

[I-D.bagnulo-lisp-threat]
Bagnulo, M., "Preliminary LISP Threat Analysis", draft-bagnulo-lisp-threat-01 (work in progress), July 2007.

[I-D.ietf-sidr-arch]
Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", draft-ietf-sidr-arch-12 (work in progress), February 2011.

[I-D.ietf-tcpm-tcp-security]
Gont, F., "Security Assessment of the Transmission Control Protocol (TCP)", draft-ietf-tcpm-tcp-security-02 (work in progress), January 2011.

[I-D.lear-lisp-nerd]
Lear, E., "NERD: A Not-so-novel EID to RLOC Database", draft-lear-lisp-nerd-08 (work in progress), March 2010.

[I-D.maino-lisp-sec]
Maino, F., Ermagan, V., Cabellos-Aparicio, A., Saucez, D., and O. Bonaventure, "LISP-Security (LISP-SEC)", draft-maino-lisp-sec-00 (work in progress), March 2011.

[I-D.meyer-lisp-cons]
Brim, S., "LISP-CONS: A Content distribution Overlay Network Service for LISP", draft-meyer-lisp-cons-04 (work in progress), April 2008.

[I-D.saucez-lisp-mapping-security]
Saucez, D. and O. Bonaventure, "Securing LISP Mapping replies", draft-saucez-lisp-mapping-security-00 (work in progress), February 2011.

[RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, March 2004.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC5386] Williams, N. and M. Richardson, "Better-Than-Nothing Security: An Unauthenticated Mode of IPsec", RFC 5386, November 2008.

[SAVI] IETF, "Source Address Validation Improvements Working Group", <http://tools.ietf.org/wg/savi/>.

[Saucez09]
Saucez, D. and L. Iannone, "How to mitigate the effect of scans on mapping systems", Submitted to the Trilogy Summer School on Future Internet.

Authors' Addresses

Damien Saucez
Universite catholique de Louvain
Place St. Barbe 2
Louvain la Neuve
Belgium

Email: damien.saucez@uclouvain.be

Luigi Iannone
TU Berlin - Deutsche Telekom Laboratories AG
Ernst-Reuter Platz 7
Berlin
Germany

Email: luigi@net.t-labs.tu-berlin.de

Olivier Bonaventure
Universite catholique de Louvain
Place St. Barbe 2
Louvain la Neuve
Belgium

Email: olivier.bonaventure@uclouvain.be