RFC 9299 An Architectural Introduction to the Locator/ID Separation Protocol (LISP)

Albert Cabellos, Damien Saucez

Stream: Internet Engineering Task Force (IETF)
RFC: 9299
Category: Informational
Published: October 2022
ISSN: 2070-1721
Authors: A. Cabellos
Universitat Politecnica de Catalunya
D. Saucez, Ed.
Inria

RFC 9299
An Architectural Introduction to the Locator/ID Separation Protocol (LISP)

Abstract

This document describes the architecture of the Locator/ID Separation Protocol (LISP), making it easier to read the rest of the LISP specifications and providing a basis for discussion about the details of the LISP protocols. This document is used for introductory purposes; more details can be found in the protocol specifications, RFCs 9300 and 9301.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9299.

Copyright Notice

Table of Contents

1. Introduction

This document introduces the Locator/ID Separation Protocol (LISP) architecture [RFC9300] [RFC9301], its main operational mechanisms, and its design rationale. Fundamentally, LISP is built following a well-known architectural idea: decoupling the overloaded semantics of IP addresses. As pointed out by Noel Chiappa [RFC4984], currently, IP addresses identify both the topological location of a network attachment point as well as the node's identity. However, nodes and routing have fundamentally different requirements. On one hand, routing systems require that addresses be aggregatable and have topological meaning; on the other hand, nodes must be identified independently of their current location [RFC4984].

LISP creates two separate namespaces, Endpoint Identifiers (EIDs) and Routing Locators (RLOCs). Both are syntactically identical to the current IPv4 and IPv6 addresses. However, EIDs are used to uniquely identify nodes irrespective of their topological location and are typically routed intra-domain. RLOCs are assigned topologically to network attachment points and are typically routed inter-domain. With LISP, the edge of the Internet (where the nodes are connected) and the core (where inter-domain routing occurs) can be logically separated. LISP-capable routers interconnect the two logical spaces. LISP also introduces a database, called the Mapping System, to store and retrieve mappings between identity and location. LISP-capable routers exchange packets over the Internet core by encapsulating them to the appropriate location.

In summary:

- RLOCs have meaning only in the underlay network, that is, the underlying core routing system.
- EIDs have meaning only in the overlay network, which is the encapsulation relationship between LISP-capable routers.
- The LISP edge maps EIDs to RLOCs.
- Within the underlay network, RLOCs have both Locator and identifier semantics.

- An EID within a LISP site carries both identifier and Locator semantics to other nodes within that site.
- An EID within a LISP site carries identifier and limited Locator semantics to nodes at other LISP sites (i.e., enough Locator information to tell that the EID is external to the site).

The relationship described above is not unique to LISP, and it is common to other overlay technologies.

The initial motivation in the LISP effort is to be found in the routing scalability problem [RFC4984], where, if LISP were to be completely deployed, the Internet core is populated with RLOCs while Traffic Engineering (TE) mechanisms are pushed to the Mapping System. In such a scenario, RLOCs are quasi-static (i.e., low churn), hence making the routing system scalable [Quoitin], while EIDs can roam anywhere with no churn to the underlying global routing system. [RFC7215] discusses the impact of LISP on the global routing system during the transition period. However, the separation between location and identity that LISP offers makes it suitable for use in additional scenarios, such as TE, multihoming, and mobility among others.

This document describes the LISP architecture and its main operational mechanisms as well as its design rationale. It is important to note that this document does not specify or complement LISP. The interested reader should refer to the main LISP specifications (see [RFC9300] and [RFC9301]), as well as the complementary documents (i.e., [RFC6831], [RFC6832], [RFC9302], [RFC6835], [RFC6836], and [RFC7052]) for the protocol specifications along with the LISP deployment guidelines [RFC7215].

2. Definitions of Terms

Endpoint Identifier (EID): Addresses used to uniquely identify nodes irrespective of their topological location. Typically routed intra-domain.

Routing Locator (RLOC): Addresses assigned topologically to network attachment points. Typically routed inter-domain.

Ingress Tunnel Router (ITR): A LISP-capable router that encapsulates packets from a LISP site towards the core network.

Egress Tunnel Router (ETR): A LISP-capable router that decapsulates packets from the core of the network towards a LISP site.

xTR: A router that implements both ITR and ETR functionalities.

Map-Request: A LISP signaling message used to request an EID-to-RLOC mapping.

Map-Reply: A LISP signaling message sent in response to a Map-Request that contains a resolved EID-to-RLOC mapping.

Map-Register: A LISP signaling message used to register an EID-to-RLOC mapping.

Map-Notify: A LISP signaling message sent in response of a Map-Register to acknowledge the correct reception of an EID-to-RLOC mapping.

This document describes the LISP architecture and does not introduce any new terms. The reader is referred to [RFC9300], [RFC9301], [RFC6831], [RFC6832], [RFC9302], [RFC6835], [RFC6836], [RFC7052], and [RFC7215] for the complete definition of terms.

3. LISP Architecture

This section presents the LISP architecture. It first details the design principles of LISP, and then it proceeds to describe its main aspects: data plane, control plane, and internetworking mechanisms.

3.1. Design Principles

The LISP architecture is built on top of four basic design principles:

Locator/Identifier split: Decoupling the overloaded semantics of current IP addresses allows devices to have identity-based addresses that are separate from topologically meaningful addresses. By allowing only the topologically meaningful addresses to be exposed to the Internet core, those topologically meaningful addresses can be aggregated to support substantial scaling. Individual devices are assigned identity-based addresses that are not used for forwarding in the Internet core.

Overlay architecture: This architecture overlays route packets over the current Internet, allowing deployment of new protocols without changing the current infrastructure; hence, this results in a low deployment cost.

Decoupled data plane and control plane: Separating the data plane from the control plane allows them to scale independently and use different architectural approaches. This is important given that they typically have different requirements and allows for other data planes to be added. Even though the data plane and the control plane are decoupled, they are not completely isolated, because the LISP data plane may trigger control plane activity.

Incremental deployability: This principle ensures that the protocol interoperates with the legacy Internet while providing some of the targeted benefits to early adopters.

3.2. Overview of the Architecture

LISP architecturally splits the core from the edge of the Internet by creating two separate namespaces: Endpoint Identifiers (EIDs) and Routing Locators (RLOCs). The edge consists of LISP sites (e.g., an Autonomous System) that use EID addresses. EIDs are IPv4 or IPv6 addresses that uniquely identify communication end hosts and are assigned and configured by the same mechanisms that exist at the time of this writing. EIDs do not contain inter-domain topological information, and because of this, EIDs are usually routable at the edge (within LISP sites) but not in the core; see Section 3.5 for discussion of LISP site internetworking with non-LISP sites and domains in the Internet.

LISP sites (at the edge) are connected to the interconnecting core of the Internet by means of LISP-capable routers (e.g., border routers). LISP sites are connected across the interconnecting core of the Internet using tunnels between the LISP-capable routers. When packets originated from a LISP site are flowing towards the core network, they ingress into an encapsulated tunnel via an Ingress Tunnel Router (ITR). When packets flow from the core network to a LISP site, they egress from an encapsulated tunnel to an Egress Tunnel Router (ETR). An xTR is a router that can perform both ITR and ETR operations. In this context, ITRs encapsulate packets, while ETRs decapsulate them; hence, LISP operates as an overlay on top of the current Internet core.

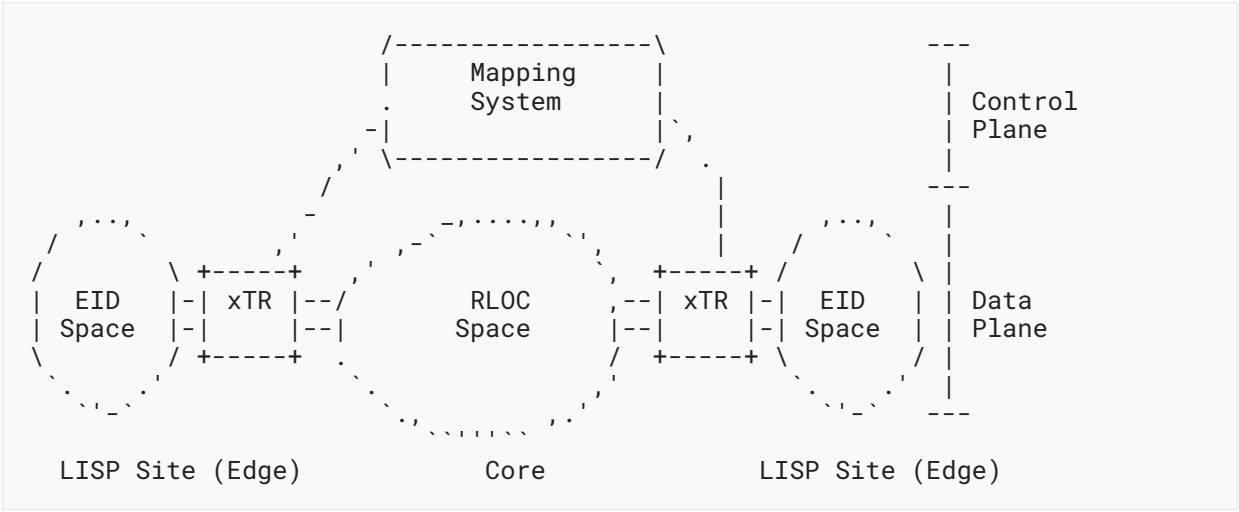

Figure 1: A Schema of the LISP Architecture

With LISP, the core uses RLOCs. An RLOC is an IPv4 or IPv6 address assigned to a core-facing network interface of an ITR or ETR.

A database that is typically distributed, called the Mapping System, stores mappings between EIDs and RLOCs. Such mappings relate the identity of the devices attached to LISP sites (EIDs) to the set of RLOCs configured at the LISP-capable routers servicing the site. Furthermore, the mappings also include TE policies and can be configured to achieve multihoming and load balancing. The LISP Mapping System is conceptually similar to the DNS, where it is organized as a distributed multi-organization network database. With LISP, ETRs register mappings, while ITRs retrieve them.

Finally, the LISP architecture emphasizes incremental deployment. Given that LISP represents an overlay to the current Internet architecture, end hosts, as well as intra-domain and inter-domain routers, remain unchanged. The only required changes to the existing infrastructure are to routers connecting the EID space with the RLOC space. Additionally, LISP requires the deployment of an independent Mapping System; such a distributed database is a new network entity.

The following describes a simplified packet flow sequence between two nodes that are attached to LISP sites. Please note that typical LISP-capable routers are xTRs (both ITR and ETR). Client HostA wants to send a packet to server HostB.

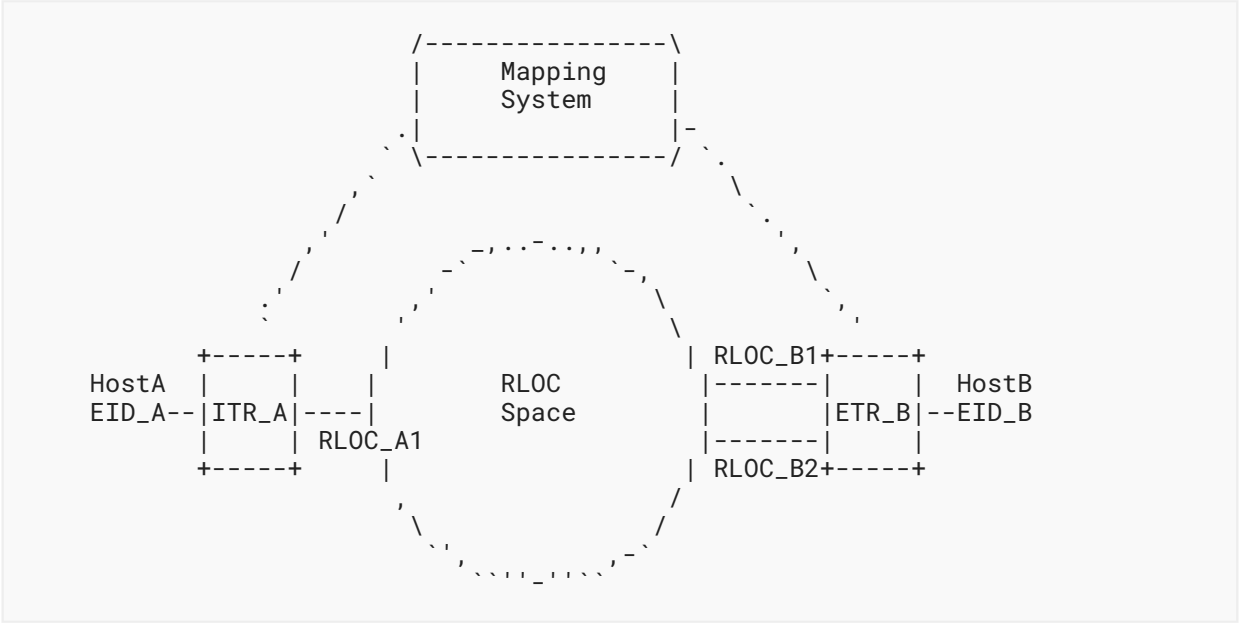

Figure 2: Packet Flow Sequence in LISP

1. HostA retrieves the EID_B of HostB, typically querying the DNS and obtaining an A or AAAA record. Then, it generates an IP packet as in the Internet. The packet has source address EID_A and destination address EID_B.
2. The packet is forwarded towards ITR_A in the LISP site using standard intra-domain mechanisms.
3. ITR_A, upon receiving the packet, queries the Mapping System to retrieve the Locator of ETR_B that is servicing HostB's EID_B. In order to do so, it uses a LISP control message called Map-Request. The message contains EID_B as the lookup key. In turn, it receives another LISP control message called Map-Reply. The message contains two Locators: RLOC_B1 and RLOC_B2. It also contains TE policies: priority and weight per Locator. Note that a Map-Reply can contain more Locators if needed. ITR_A can cache the mapping in local storage to speed up forwarding of subsequent packets.
4. ITR_A encapsulates the packet towards RLOC_B1 (chosen according to the priorities/weights specified in the mapping). The packet contains two IP headers. The outer header has RLOC_A1 as source and RLOC_B1 as destination. The inner original header has EID_A as source and EID_B as destination. Furthermore, ITR_A adds a LISP header. More details about LISP encapsulation can be found in Section 3.3.1.
5. The encapsulated packet is forwarded over the interconnecting core as a normal IP packet, making the EID invisible from the core.

6. Upon reception of the encapsulated packet by ETR_B, it decapsulates the packet and forwards it to HostB.

3.3. Data Plane

This section provides a high-level description of the LISP data plane, which is specified in detail in [RFC9300]. The LISP data plane is responsible for encapsulating and decapsulating data packets and caching the appropriate forwarding state. It includes two main entities, the ITR and the ETR. Both are LISP-capable routers that connect the EID with the RLOC space (ITR) and vice versa (ETR).

3.3.1. LISP Encapsulation

ITRs encapsulate data packets towards ETRs. LISP data packets are encapsulated using UDP (port 4341). The source port is usually selected by the ITR using a 5-tuple hash of the inner header (so as to be consistent in case of multipath solutions, such as ECMP [RFC2992]) and ignored on reception. LISP data packets are often encapsulated in UDP packets that include a zero checksum [RFC6935] [RFC6936] that may not be verified when it is received, because LISP data packets typically include an inner transport protocol header with a non-zero checksum. The use of UDP zero checksums over IPv6 for all tunneling protocols like LISP is subject to the applicability statement in [RFC6936]. If LISP data packets are encapsulated in UDP packets with non-zero checksums, the outer UDP checksums are verified when the UDP packets are received, as part of normal UDP processing.

LISP-encapsulated packets also include a LISP header (after the UDP header and before the original IP header). The LISP header is prepended by ITRs and stripped by ETRs. It carries reachability information (see more details in Section 4.2) and the 'Instance ID' field. The 'Instance ID' field is used to distinguish traffic to/from different tenant address spaces at the LISP site, and this use of the Instance ID may use overlapped but logically separated EID addressing.

Overall, LISP works on 4 headers: the inner header the source constructed and the 3 headers a LISP encapsulator prepends ("outer" to "inner"):

1. Outer IP header containing RLOCs as source and destination addresses. This header is originated by ITRs and stripped by ETRs.
2. UDP header (port 4341), usually with zero checksum. This header is originated by ITRs and stripped by ETRs.
3. LISP header that contains various forwarding-plane features (such as reachability) and an 'Instance ID' field. This header is originated by ITRs and stripped by ETRs.
4. Inner IP header containing EIDs as source and destination addresses. This header is created by the source end host and is left unchanged by the LISP data plane processing on the ITR and ETR.

Finally, in some scenarios, re-encapsulating and/or recursive tunnels are useful to choose a specified path in the underlay network, for instance, to avoid congestion or failure. Re-encapsulating tunnels are consecutive LISP tunnels and occur when a decapsulator (an ETR action) removes a LISP header and then acts as an encapsulator (an ITR action) to prepend

another one. On the other hand, recursive tunnels are nested tunnels and are implemented by using multiple LISP encapsulations on a packet. Such functions are implemented by Re-encapsulating Tunnel Routers (RTRs). An RTR can be thought of as a router that first acts as an ETR by decapsulating packets and then as an ITR by encapsulating them towards another Locator; more information can be found in [RFC9300] and [RFC9301].

3.3.2. LISP Forwarding State

In the LISP architecture, ITRs keep just enough information to route traffic flowing through them. In other words, ITRs only need to retrieve from the LISP Mapping System mappings between EID-Prefixes (blocks of EIDs) and RLOCs that are used to encapsulate packets. Such mappings are stored in a local cache called the LISP Map-Cache for subsequent packets addressed to the same EID-Prefix. Note that in the case of overlapping EID-Prefixes, after a request, the ITR may receive a set of mappings covering the requested EID-Prefix and all more-specific EID-Prefixes (cf., Section 5.5 of [RFC9301]). Mappings include a Time to Live (TTL) (set by the ETR). More details about the Map-Cache management can be found in Section 4.1.

3.4. Control Plane

The LISP control plane, specified in [RFC9301], provides a standard interface to register and request mappings. The LISP Mapping System is a database that stores such mappings. The following sub-sections first describe the mappings, then the standard interface to the Mapping System, and finally its architecture.

3.4.1. LISP Mappings

Each mapping includes the bindings between EID-Prefix(es) and a set of RLOCs as well as TE policies, in the form of priorities and weights for the RLOCs. Priorities allow the ETR to configure active/backup policies, while weights are used to load-balance traffic among the RLOCs (on a per-flow basis).

Typical mappings in LISP bind EIDs in the form of IP prefixes with a set of RLOCs, also in the form of IP addresses. IPv4 and IPv6 addresses are encoded using the appropriate Address Family Identifier (AFI) [RFC8060]. However, LISP can also support more general address encoding by means of the ongoing effort around the LISP Canonical Address Format (LCAF) [RFC8060].

With such a general syntax for address encoding in place, LISP aims to provide flexibility to current and future applications. For instance, LCAFs could support Media Access Control (MAC) addresses, geocoordinates, ASCII names, and application-specific data.

3.4.2. Mapping System Interface

LISP defines a standard interface between data and control planes. The interface is specified in [RFC9301] and defines two entities:

Map-Server:

A network infrastructure component that learns mappings from ETRs and publishes them into the LISP Mapping System. Typically, Map-Servers are not authoritative to reply to queries; hence, they forward them to the ETR. However, they can also operate in proxy-mode, where the ETRs delegate replying to queries to Map-Servers. This setup is useful when the ETR has limited resources (e.g., CPU or power).

Map-Resolver: A network infrastructure component that interfaces ITRs with the Mapping System by proxying queries and, in some cases, responses.

The interface defines four LISP control messages that are sent as UDP datagrams (port 4342):

Map-Register: This message is used by ETRs to register mappings in the Mapping System, and it is authenticated using a shared key between the ETR and the Map-Server.

Map-Notify: When requested by the ETR, this message is sent by the Map-Server in response to a Map-Register to acknowledge the correct reception of the mapping and convey the latest Map-Server state on the EID-to-RLOC mapping. In some cases, a Map-Notify can be sent to the previous RLOCs when an EID is registered by a new set of RLOCs.

Map-Request: This message is used by ITRs or Map-Resolvers to resolve the mapping of a given EID.

Map-Reply: This message is sent by Map-Servers or ETRs in response to a Map-Request and contains the resolved mapping. Please note that a Map-Reply may contain a negative reply if, for example, the queried EID is not part of the LISP EID space. In such cases, the ITR typically forwards the traffic as is (non-encapsulated) to the public Internet. This behavior is defined to support incremental deployment of LISP.

3.4.3. Mapping System

LISP architecturally decouples control and data planes by means of a standard interface. This interface glues the data plane -- routers responsible for forwarding data packets -- with the LISP Mapping System -- a database responsible for storing mappings.

With this separation in place, the data and control planes can use different architectures if needed and scale independently. Typically, the data plane is optimized to route packets according to hierarchical IP addresses. However, the control plane may have different requirements, for instance, and by taking advantage of the LCAFs, the Mapping System may be used to store nonhierarchical keys (such as MAC addresses), requiring different architectural approaches for scalability. Another important difference between the LISP control and data planes is that, and as a result of the local mapping cache available at the ITR, the Mapping System does not need to operate at line-rate.

Many of the existing mechanisms to create distributed systems have been explored and considered for the Mapping System architecture: graph-based databases in the form of LISP Alternative Logical Topology (LISP-ALT) [RFC6836], hierarchical databases in the form of the LISP Delegated Database Tree (LISP-DDT) [RFC8111], monolithic databases in the form of the LISP Not-so-novel EID-to-RLOC Database (LISP-NERD) [RFC6837], flat databases in the form of the LISP

Distributed Hash Table (LISP-DHT) [LISP-SHDHT] [Mathy], and a multicast-based database [LISP-EMACS]. Furthermore, it is worth noting that, in some scenarios, such as private deployments, the Mapping System can operate as logically centralized. In such cases, it is typically composed of a single Map-Server/Map-Resolver.

The following sub-sections focus on the two Mapping Systems that have been implemented and deployed (LISP-ALT and LISP-DDT).

3.4.3.1. LISP-ALT

LISP-ALT [RFC6836] was the first Mapping System proposed, developed, and deployed on the LISP pilot network. It is based on a distributed BGP overlay in which Map-Servers and Map-Resolvers participate. The nodes connect to their peers through static tunnels. Each Map-Server involved in the ALT topology advertises the EID-Prefixes registered by the serviced ETRs, making the EID routable on the ALT topology.

When an ITR needs a mapping, it sends a Map-Request to a Map-Resolver that, using the ALT topology, forwards the Map-Request towards the Map-Server responsible for the mapping. Upon reception, the Map-Server forwards the request to the ETR, which in turn replies directly to the ITR.

3.4.3.2. LISP-DDT

LISP-DDT [RFC8111] is conceptually similar to the DNS, a hierarchical directory whose internal structure mirrors the hierarchical nature of the EID address space. The DDT hierarchy is composed of DDT nodes forming a tree structure; the leafs of the tree are Map-Servers. On top of the structure, there is the DDT root node, which is a particular instance of a DDT node, that matches the entire address space. As in the case of DNS, DDT supports multiple redundant DDT nodes and/or DDT roots. Finally, Map-Resolvers are the clients of the DDT hierarchy and can query the DDT root and/or other DDT nodes.

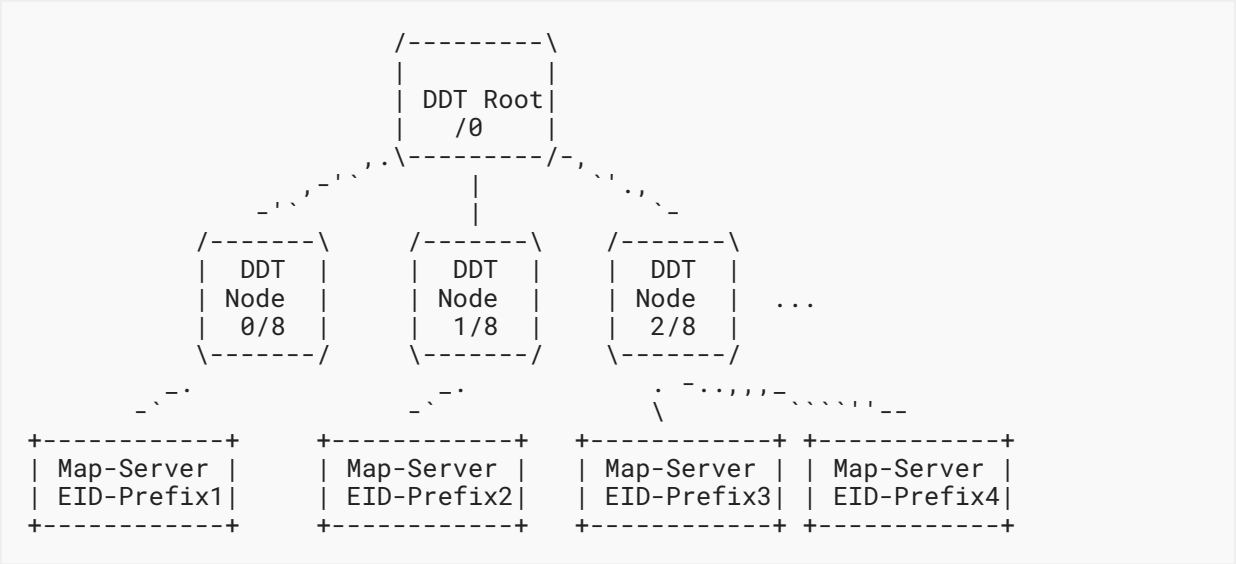

Figure 3: A Schematic Representation of the DDT Tree Structure

Please note that the prefixes and the structure depicted in the figure above should only be considered as an example.

The DDT structure does not actually index EID-Prefixes; rather, it indexes Extended EID-Prefixes (XEID-Prefixes). An XEID-Prefix is just the concatenation of the following fields (from most significant bit to less significant bits): Database-ID, Instance ID, Address Family Identifier, and the actual EID-Prefix. The Database-ID is provided for possible future requirements of higher levels in the hierarchy and to enable the creation of multiple and separate database trees.

In order to resolve a query, LISP-DDT operates in a similar way to the DNS but only supports iterative lookups. DDT clients (usually Map-Resolvers) generate Map-Requests to the DDT root node. In response, they receive a newly introduced LISP control message: a Map-Referral. A Map-Referral provides the list of RLOCs of the set of DDT nodes matching a configured XEID delegation. That is, the information contained in the Map-Referral points to the child of the queried DDT node that has more specific information about the queried XEID-Prefix. This process is repeated until the DDT client walks the tree structure (downwards) and discovers the Map-Server servicing the queried XEID. At this point, the client sends a Map-Request and receives a Map-Reply containing the mappings. It is important to note that DDT clients can also cache the information contained in Map-Referrals; that is, they cache the DDT structure. This is used to reduce the time required to retrieve mappings [Jakab].

The DDT Mapping System relies on manual configuration. That is, Map-Resolvers are configured with the set of available DDT root nodes, while DDT nodes are configured with the appropriate XEID delegations. Configuration changes in the DDT nodes are only required when the tree structure changes itself, but it doesn't depend on EID dynamics (RLOC allocation or TE policy changes).

3.5. Internetworking Mechanisms

EIDs are typically identical to either IPv4 or IPv6 addresses, and they are stored in the LISP Mapping System. However, they are usually not announced in the routing system beyond the local LISP domain. As a result, LISP requires an internetworking mechanism to allow LISP sites to speak with non-LISP sites and vice versa. LISP internetworking mechanisms are specified in [RFC6832].

LISP defines two entities to provide internetworking:

Proxy Ingress Tunnel Router (PITR): PITRs provide connectivity from the legacy Internet to LISP sites. PITRs announce in the global routing system blocks of EID-Prefixes (aggregating when possible) to attract traffic. For each incoming packet from a source not in a LISP site (a non-EID), the PITR LISP-encapsulates it towards the RLOC(s) of the appropriate LISP site. The impact of PITRs on the routing table size of the Default-Free Zone (DFZ) is, in the worst case, similar to the case in which LISP is not deployed. EID-Prefixes will be aggregated as much as possible, both by the PITR and by the global routing system.

Proxy Egress Tunnel Router (PETR): PETRs provide connectivity from LISP sites to the legacy Internet. In some scenarios, LISP sites may be unable to send encapsulated packets with a local EID address as a source to the legacy Internet, for instance, when Unicast Reverse Path Forwarding (uRPF) is used by Provider Edge routers or when an intermediate network between a LISP site and a non-LISP site does not support the desired version of IP (IPv4 or IPv6). In both cases, the PETR overcomes such limitations by encapsulating packets over the network. There is no specified provision for the distribution of PETR RLOC addresses to the ITRs.

Additionally, LISP also defines mechanisms to operate with private EIDs [RFC1918] by means of LISP-NAT [RFC6832]. In this case, the xTR replaces a private EID source address with a routable one. At the time of this writing, work is ongoing to define NAT-traversal capabilities, that is, xTRs behind a NAT using non-routable RLOCs.

PITRs, PETRs, and LISP-NAT enable incremental deployment of LISP by providing significant flexibility in the placement of the boundaries between the LISP and non-LISP portions of the network and making it easy to change those boundaries over time.

4. LISP Operational Mechanisms

This section details the main operational mechanisms defined in LISP.

4.1. Cache Management

LISP's decoupled control and data planes, where mappings are stored in the control plane and used for forwarding in the data plane, require a local cache in ITRs to reduce signaling overhead (Map-Request/Map-Reply) and increase forwarding speed. The local cache available at the ITRs,

called Map-Cache, is used by the router to LISP-encapsulate packets. The Map-Cache is indexed by (Instance ID, EID-Prefix) and contains basically the set of RLOCs with the associated TE policies (priorities and weights).

The Map-Cache, as with any other cache, requires cache coherence mechanisms to maintain up-to-date information. LISP defines three main mechanisms for cache coherence:

Record Time To Live (TTL): Each mapping record contains a TTL set by the ETR. Upon expiration of the TTL, the ITR can't use the mapping until it is refreshed by sending a new Map-Request.

Solicit-Map-Request (SMR): SMR is an explicit mechanism to update mapping information. In particular, a special type of Map-Request can be sent on demand by ETRs to request refreshing a mapping. Upon reception of an SMR message, the ITR must refresh the bindings by sending a Map-Request to the Mapping System. Further uses of SMRs are documented in [RFC9301].

Map-Versioning: This optional mechanism piggybacks, in the LISP header of data packets, the version number of the mappings used by an xTR. This way, when an xTR receives a LISP-encapsulated packet from a remote xTR, it can check whether its own Map-Cache or the one of the remote xTR is outdated. If its Map-Cache is outdated, it sends a Map-Request for the remote EID so as to obtain the newest mappings. On the contrary, if it detects that the remote xTR Map-Cache is outdated, it sends an SMR to notify it that a new mapping is available. Further details are available in [RFC9302].

Finally, it is worth noting that, in some cases, an entry in the Map-Cache can be proactively refreshed using the mechanisms described in the section below.

4.2. RLOC Reachability

In most cases, LISP operates with a pull-based Mapping System (e.g., DDT). This results in an edge-to-edge pull architecture. In such a scenario, the network state is stored in the control plane while the data plane pulls it on demand. This has consequences concerning the propagation of xTRs' reachability/liveness information, since pull architectures require explicit mechanisms to propagate this information. As a result, LISP defines a set of mechanisms to inform ITRs and PITRs about the reachability of the cached RLOCs:

Locator-Status-Bits (LSBs): Using LSBs is a passive technique. The 'LSB' field is carried by data packets in the LISP header and can be set by ETRs to specify which RLOCs of the ETR site are up/down. This information can be used by the ITRs as a hint about the reachability to perform additional checks. Also note that LSBs do not provide path reachability status; they only provide hints about the status of RLOCs. As such, they must not be used over the public Internet and should be coupled with Map-Versioning to prevent race conditions where LSBs are interpreted as referring to different RLOCs than intended.

Echo-Nonce: This is also a passive technique that can only operate effectively when data flows bidirectionally between two communicating xTRs. Basically, an ITR piggybacks a random number (called a nonce) in LISP data packets. If the path and the probed Locator are up, the ETR will piggyback the same random number on the next data packet; if this is not the case, the ITR can set the Locator as unreachable. When traffic flow is unidirectional or when the ETR receiving the traffic is not the same as the ITR that transmits it back, additional mechanisms are required. The Echo-Nonce mechanism must be used in trusted environments only, not over the public Internet.

RLOC-Probing: This is an active probing algorithm where ITRs send probes to specific Locators. This effectively probes both the Locator and the path. In particular, this is done by sending a Map-Request (with certain flags activated) on the data plane (RLOC space) and then waiting for a Map-Reply (also sent on the data plane). The active nature of RLOC-Probing provides an effective mechanism for determining reachability and, in case of failure, switching to a different Locator. Furthermore, the mechanism also provides useful RTT estimates of the delay of the path that can be used by other network algorithms.

It is worth noting that RLOC-Probing and the Echo-Nonce can work together. Specifically, if a nonce is not echoed, an ITR cannot determine which path direction has failed. In this scenario, an ITR can use RLOC-Probing.

Additionally, LISP also recommends inferring the reachability of Locators by using information provided by the underlay, particularly:

ICMP signaling: The LISP underlay -- the current Internet -- uses ICMP to signal unreachability (among other things). LISP can take advantage of this, and the reception of an ICMP Network Unreachable or ICMP Host Unreachable message can be seen as a hint that a Locator might be unreachable. This should lead to performing additional checks.

Underlay routing: Both BGP and IGP carry reachability information. LISP-capable routers that have access to underlay routing information can use it to determine if a given Locator or path is reachable.

4.3. ETR Synchronization

All the ETRs that are authoritative to a particular EID-Prefix must announce the same mapping to the requesters. This means that ETRs must be aware of the status of the RLOCs of the remaining ETRs. This is known as ETR synchronization.

At the time of this writing, LISP does not specify a mechanism to achieve ETR synchronization. Although many well-known techniques could be applied to solve this issue, it is still under research. As a result, operators must rely on coherent manual configuration.

4.4. MTU Handling

Since LISP encapsulates packets, it requires dealing with packets that exceed the MTU of the path between the ITR and the ETR. Specifically, LISP defines two mechanisms:

Stateless: With this mechanism, the effective MTU is assumed from the ITR's perspective. If a payload packet is too big for the effective MTU and can be fragmented, the payload packet is fragmented on the ITR, such that reassembly is performed at the destination host.

Stateful: With this mechanism, ITRs keep track of the MTU of the paths towards the destination Locators by parsing the ICMP Too Big packets sent by intermediate routers. ITRs will send ICMP Too Big messages to inform the sources about the effective MTU. Additionally, ITRs can use mechanisms such as Path MTU Discovery (PMTUD) [RFC1191] or Packetization Layer Path MTU Discovery (PLPMTUD) [RFC4821] to keep track of the MTU towards the Locators.

In both cases, if the packet cannot be fragmented (IPv4 with DF=1 or IPv6), then the ITR drops it and replies with an ICMP Too Big message to the source.

5. Mobility

The separation between Locators and identifiers in LISP is suitable for TE purposes where LISP sites can change their attachment points to the Internet (i.e., RLOCs) without impacting endpoints or the Internet core. In this context, the border routers operate the xTR functionality, and endpoints are not aware of the existence of LISP. This functionality is similar to Network Mobility [RFC3963]. However, this mode of operation does not allow seamless mobility of endpoints between different LISP sites, as the EID address might not be routable in a visited site. Nevertheless, LISP can be used to enable seamless IP mobility when LISP is directly implemented in the endpoint or when the endpoint roams to an attached xTR. Each endpoint is then an xTR, and the EID address is the one presented to the network stack used by applications while the RLOC is the address gathered from the network when it is visited. This functionality is similar to Mobile IP ([RFC5944] and [RFC6275]).

Whenever a device changes its RLOC, the xTR updates the RLOC of its local mapping and registers it to its Map-Server, typically with a low TTL value (1 min). To avoid the need for a home gateway, the ITR also indicates the RLOC change to all remote devices that have ongoing communications with the device that moved. The combination of both methods ensures the scalability of the system, as signaling is strictly limited to the Map-Server and to hosts with which communications are ongoing. In the mobility case, the EID-Prefix can be as small as a full /32 or /128 (IPv4 or IPv6, respectively), depending on the specific use case (e.g., subnet mobility vs. single VM/Mobile node mobility).

The decoupled identity and location provided by LISP allow it to operate with other Layer 2 and Layer 3 mobility solutions.

6. Multicast

LISP also supports transporting IP multicast packets sent from the EID space. The required operational changes to the multicast protocols are documented in [RFC6831].

In such scenarios, LISP may create multicast state both at the core and at the sites (both source and receiver). When signaling is used to create multicast state at the sites, LISP routers encapsulate PIM Join/Prune messages from receiver to source sites as unicast packets. At the core, ETRs build a new PIM Join/Prune message addressed to the RLOC of the ITR servicing the source. A simplified sequence is shown below.

1. An end host willing to join a multicast channel sends an IGMP report. Multicast PIM routers at the LISP site propagate PIM Join/Prune messages (S-EID, G) towards the ETR.
2. The Join message flows to the ETR. Upon reception, the ETR builds two Join messages. The first one unicast LISP-encapsulates the original Join message towards the RLOC of the ITR servicing the source. This message creates (S-EID, G) multicast state at the source site. The second Join message contains, as a destination address, the RLOC of the ITR servicing the source (S-RLOC, G) and creates multicast state at the core.
3. Multicast data packets originated by the source (S-EID, G) flow from the source to the ITR. The ITR LISP-encapsulates the multicast packets. The outer header includes its own RLOC as the source (S-RLOC) and the original multicast group address (G) as the destination. Please note that multicast group addresses are logical and are not resolved by the Mapping System. Then, the multicast packets are transmitted through the core towards the receiving ETRs, which decapsulate the packets and forward them using the receiver site's multicast state.

Please note that the inner and outer multicast addresses are generally different, except in specific cases where the underlay provider implements tight control on the overlay. LISP specifications already support all PIM modes [RFC6831]. Additionally, LISP can also support non-PIM mechanisms in order to maintain multicast state.

When multicast sources and receivers are active at LISP sites and the core network between the sites does not provide multicast support, a signal-free mechanism can be used to create an overlay that will allow multicast traffic to flow between sites and connect the multicast trees at the different sites [RFC8378]. Registrations from the different receiver sites will be merged in the Mapping System to assemble a multicast replication list inclusive of all RLOCs that lead to receivers for a particular multicast group or multicast channel. The replication list for each specific multicast entry is maintained as a database mapping entry in the LISP Mapping System.

7. Use Cases

7.1. Traffic Engineering

A LISP site can strictly impose via which ETRs the traffic must enter the LISP site network even though the path followed to reach the ETR is not under the control of the LISP site. This fine control is implemented with the mappings. When a remote site is willing to send traffic to a LISP site, it retrieves the mapping associated with the destination EID via the Mapping System. The mapping is sent directly by an authoritative ETR of the EID and is not altered by any intermediate network.

A mapping associates a list of RLOCs with an EID-Prefix. Each RLOC corresponds to an interface of an ETR (or set of ETRs) that is able to correctly forward packets to EIDs in the prefix. Each RLOC is tagged with a priority and a weight in the mapping. The priority is used to indicate which RLOCs should be preferred for sending packets (the least preferred ones being provided for backup purposes). The weight permits balancing the load between the RLOCs with the same priority, in proportion to the weight value.

As mappings are directly issued by the authoritative ETR of the EID and are not altered when transmitted to the remote site, it offers highly flexible incoming inter-domain TE and even makes it possible for a site to support a different mapping policy for each remote site.

7.2. LISP for IPv6 Co-existence

LISP encapsulations allow transporting packets using EIDs from a given address family (e.g., IPv6) with packets from other address families (e.g., IPv4). The absence of correlation between the address families of RLOCs and EIDs makes LISP a candidate to allow, e.g., IPv6 to be deployed when all of the core network may not have IPv6 enabled.

For example, two IPv6-only data centers could be interconnected via the legacy IPv4 Internet. If their border routers are LISP capable, sending packets between the data centers is done without any form of translation, as the original IPv6 packets (in the EID space) will be LISP encapsulated and transmitted over the IPv4 legacy Internet via IPv4 RLOCs.

7.3. LISP for Virtual Private Networks

It is common to operate several virtual networks over the same physical infrastructure. In such virtual private networks, determining to which virtual network a packet belongs is essential; tags or labels are used for that purpose. When using LISP, the distinction can be made with the 'Instance ID' field. When an ITR encapsulates a packet from a particular virtual network (e.g., known via Virtual Routing and Forwarding (VRF) or the VLAN), it tags the encapsulated packet with the Instance ID corresponding to the virtual network of the packet. When an ETR receives a packet tagged with an Instance ID, it uses the Instance ID to determine how to treat the packet.

The main usage of LISP for virtual private networks does not introduce additional requirements on the underlying network, as long as it runs IP.

7.4. LISP for Virtual Machine Mobility in Data Centers

A way to enable seamless virtual machine (VM) mobility in the data center is to conceive the data center backbone as the RLOC space and the subnet where servers are hosted as forming the EID space. A LISP router is placed at the border between the backbone and each subnet. When a VM is moved to another subnet, it can keep (temporarily) the address it had before the move so as to continue without a transport-layer connection reset. When an xTR detects a source address received on a subnet to be an address not assigned to the subnet, it registers the address to the Mapping System.

To inform the other LISP routers that the machine moved and where, and then to avoid detours via the initial subnetwork, mechanisms such as the Solicit-Map-Request messages are used.

8. Security Considerations

This section describes the security considerations associated with LISP.

In a push Mapping System, the state necessary to forward packets is learned independently of the traffic itself. However, with a pull architecture, the system becomes reactive, and data plane events (e.g., the arrival of a packet with an unknown destination address) may trigger control plane events. This on-demand learning of mappings provides many advantages, as discussed above, but may also affect the way security is enforced.

Usually, the data plane is implemented in the fast path of routers to provide high-performance forwarding capabilities, while the control plane features are implemented in the slow path to offer high flexibility, and a performance gap of several orders of magnitude can be observed between the slow and fast paths. As a consequence, the way to notify the control plane of data plane events must be considered carefully so as not to overload the slow path, and rate limiting should be used as specified in [RFC9300] and [RFC9301].

Care must also be taken not to overload the Mapping System (i.e., the control plane infrastructure), as the operations to be performed by the Mapping System may be more complex than those on the data plane. For that reason, [RFC9300] and [RFC9301] recommend rate limiting the sending of messages to the Mapping System.

To improve resiliency and reduce the overall number of messages exchanged, LISP makes it possible to leak certain information, such as the reachability of Locators, directly into data plane packets. In environments that are not fully trusted, like the open Internet, control information gleaned from data plane packets must not be used or must be verified before using it.

Mappings are the centerpiece of LISP, and all precautions must be taken to prevent malicious entities from manipulating or misusing them. Using trustable Map-Servers that strictly respect [RFC9301] and the authentication mechanism proposed by LISP-SEC [RFC9303] reduces the risk of attacks on mapping integrity. In more critical environments, secure measures may be needed. The way security is implemented for a given Mapping System strongly depends on the architecture of the Mapping System itself and the threat model assumed for the deployment. Thus, Mapping System security has to be discussed in the relevant documents proposing the Mapping System architecture.

As with any other tunneling mechanism, middleboxes on the path between an ITR (or PITR) and an ETR (or PETR) must implement mechanisms to strip the LISP encapsulation to correctly inspect the content of LISP-encapsulated packets.

Like other map-and-encap mechanisms, LISP enables triangular routing (i.e., packets of a flow cross different border routers, depending on their direction). This means that intermediate boxes may have an incomplete view of the traffic they inspect or manipulate. Moreover, LISP-encapsulated packets are routed based on the outer IP address (i.e., the RLOC) and can be delivered to an ETR that is not responsible for the destination EID of the packet or even delivered

to a network element that is not an ETR. Mitigation consists of applying appropriate filtering techniques on the network elements that can potentially receive unexpected LISP-encapsulated packets.

More details about security implications of LISP are discussed in [RFC7835].

9. IANA Considerations

This document has no IANA actions.

10. References

10.1. Normative References

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

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

[RFC2992] Hopps, C., "Analysis of an Equal-Cost Multi-Path Algorithm", RFC 2992, DOI 10.17487/RFC2992, November 2000, <https://www.rfc-editor.org/info/rfc2992>.

[RFC3963] Devarapalli, V., Wakikawa, R., Petrescu, A., and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol", RFC 3963, DOI 10.17487/RFC3963, January 2005, <https://www.rfc-editor.org/info/rfc3963>.

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

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

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

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

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

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

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

[RFC6836] Fuller, V., Farinacci, D., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol Alternative Logical Topology (LISP+ALT)", RFC 6836, DOI 10.17487/RFC6836, January 2013, <https://www.rfc-editor.org/info/rfc6836>.

[RFC6837] Lear, E., "NERD: A Not-so-novel Endpoint ID (EID) to Routing Locator (RLOC) Database", RFC 6837, DOI 10.17487/RFC6837, January 2013, <https://www.rfc-editor.org/info/rfc6837>.

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

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

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

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

[RFC7835] Saucez, D., Iannone, L., and O. Bonaventure, "Locator/ID Separation Protocol (LISP) Threat Analysis", RFC 7835, DOI 10.17487/RFC7835, April 2016, <https://www.rfc-editor.org/info/rfc7835>.

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

[RFC8111] Fuller, V., Lewis, D., Ermagan, V., Jain, A., and A. Smirnov, "Locator/ID Separation Protocol Delegated Database Tree (LISP-DDT)", RFC 8111, DOI 10.17487/RFC8111, May 2017, <https://www.rfc-editor.org/info/rfc8111>.

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

[RFC9300] Farinacci, D., Fuller, V., Meyer, D., Lewis, D., and A. Cabellos, Ed., "The Locator/ID Separation Protocol (LISP)", RFC 9300, DOI 10.17487/RFC9300, October 2022, <https://www.rfc-editor.org/info/rfc9300>.

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

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

[RFC9303] Maino, F., Ermagan, V., Cabellos, A., and D. Saucez, "Locator/ID Separation Protocol Security (LISP-SEC)", RFC 9303, DOI 10.17487/RFC9303, October 2022, <https://www.rfc-editor.org/info/rfc9303>.

10.2. Informative References

[Jakab] Jakab, L., Cabellos-Aparicio, A., Coras, F., Saucez, D., and O. Bonaventure, "LISP-TREE: A DNS Hierarchy to Support the LISP Mapping System", IEEE Journal on Selected Areas in Communications, vol. 28, no. 8, pp. 1332-1343, DOI 10.1109/JSAC.2010.101011, October 2010, <https://ieeexplore.ieee.org/document/5586446>.

[LISP-EMACS] Brim, S., Farinacci, D., Meyer, D., and J. Curran, "EID Mappings Multicast Across Cooperating Systems for LISP", Work in Progress, Internet-Draft, draft-curran-lisp-emacs-00, 9 November 2007, <https://www.ietf.org/archive/id/draft-curran-lisp-emacs-00.txt>.

[LISP-SHDHT] Cheng, L. and M. Sun, "LISP Single-Hop DHT Mapping Overlay", Work in Progress, Internet-Draft, draft-cheng-lisp-shdht-04, 15 July 2013, <https://www.ietf.org/archive/id/draft-cheng-lisp-shdht-04.txt>.

[Mathy] Mathy, L. and L. Iannone, "LISP-DHT: Towards a DHT to map identifiers onto locators", CoNEXT '08: Proceedings of the 2008 ACM CoNEXT Conference, ReArch '08 - Re-Architecting the Internet, DOI 10.1145/1544012.1544073, December 2008, <https://dl.acm.org/doi/10.1145/1544012.1544073>.

[Quoitin] Quoitin, B., Iannone, L., de Launois, C., and O. Bonaventure, "Evaluating the Benefits of the Locator/Identifier Separation", Proceedings of 2nd ACM/IEEE International Workshop on Mobility in the Evolving Internet Architecture, DOI 10.1145/1366919.1366926, August 2007, <https://dl.acm.org/doi/10.1145/1366919.1366926>.

Appendix A. A Brief History of Location/Identity Separation

The LISP architecture for separation of location and identity resulted from the discussions of this topic at the Amsterdam IAB Routing and Addressing Workshop, which took place in October 2006 [RFC4984].

A small group of like-minded personnel spontaneously formed immediately after that workshop to work on an idea that came out of informal discussions at the workshop and on various mailing lists. The first Internet-Draft on LISP appeared in January 2007.

Trial implementations started at that time, with initial trial deployments underway since June 2007; the results of early experience have been fed back into the design in a continuous, ongoing process over several years. At this point, LISP represents a moderately mature system, having undergone a long, organic series of changes and updates.

LISP transitioned from an IRTF activity to an IETF WG in March 2009. After numerous revisions, the basic specifications moved to becoming RFCs at the start of 2013; work to expand, improve, and find new uses for it continues (and undoubtedly will for a long time to come). The LISP WG was rechartered in 2018 to continue work on the LISP base protocol and produce Standards Track documents.

A.1. Old LISP Models

LISP, as initially conceived, had a number of potential operating modes, named 'models'. Although they are not used anymore, one occasionally sees mention of them, so they are briefly described here.

LISP 1: EIDs all appear in the normal routing and forwarding tables of the network (i.e., they are 'routable'). This property is used to load EID-to-RLOC mappings via bootstrapping operations. Packets are sent with the EID as the destination in the outer wrapper; when an ETR sees such a packet, it sends a Map-Reply to the source ITR, giving the full mapping.

LISP 1.5: LISP 1.5 is similar to LISP 1, but the routability of EIDs happens on a separate network.

LISP 2: EIDs are not routable; EID-to-RLOC mappings are available from the DNS.

LISP 3: EIDs are not routable and have to be looked up in a new EID-to-RLOC mapping database (in the initial concept, a system using Distributed Hash Tables). Two variants were possible: a 'push' system in which all mappings were distributed to all ITRs and a 'pull' system in which ITRs load the mappings when they need them.

Acknowledgments

This document was initiated by Noel Chiappa, and much of the core philosophy came from him. The authors acknowledge the important contributions he has made to this work and thank him for his past efforts.

The authors would also like to thank Dino Farinacci, Fabio Maino, Luigi Iannone, Sharon Barkai, Isidoros Kouvelas, Christian Cassar, Florin Coras, Marc Binderberger, Alberto Rodriguez-Natal, Ronald Bonica, Chad Hintz, Robert Raszuk, Joel M. Halpern, Darrel Lewis, and David Black.

Authors' Addresses

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

Damien Saucez (EDITOR)
Inria
2004 route des Lucioles - BP 93
Sophia Antipolis
France
Email: damien.saucez@inria.fr