Network Working Group
Internet-Draft
Intended status: Informational
Expires: October 4, 2015

A. Cabellos
UPC-BarcelonaTech
D. Saucez (Ed.)
INRIA

April 02, 2015

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

**draft-ietf-lisp-introduction-13.txt**

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 RFC6830, the protocol specification.

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 4, 2015.

Table of Contents



## 1. Introduction

This document introduces the Locator/ID Separation Protocol (LISP) [RFC6830] architecture, its main operational mechanisms and its design rationale. Fundamentally, LISP is built following a well-known architectural idea: decoupling the IP address overloaded semantics. Indeed and as pointed out by Noel Chiappa [RFC4984],

currently IP addresses both identify the topological location of a network attachment point as well as the node's identity. However, nodes and routing have fundamentally different requirements, routing systems require that addresses are aggregatable and have topological meaning, while nodes require to be identified independently of their current location [RFC4984].

LISP creates two separate namespaces, EIDs (End-host IDentifiers) and RLOCs (Routing LOCators), both are syntactically identical to the current IPv4 and IPv6 addresses. 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 and interconnected by LISP-capable routers. 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 but 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 mechanisms are pushed to the Mapping System. In

such 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 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 Traffic Engineering (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 the LISP protocol. The interested reader should refer to the main LISP specifications [RFC6830] and the complementary documents [RFC6831], [RFC6832], [RFC6833], [RFC6834], [RFC6835], [RFC6836], [RFC7052] for the protocol specifications along with the LISP deployment guidelines [RFC7215].

## 2. Definition of Terms

Endpoint IDentifier (EID): EIDs are addresses used to uniquely identify nodes irrespective of their topological location and are typically routed intra-domain.

Routing LOcator (RLOC): RLOCs are addresses assigned topologically to network attachment points and 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 term. The reader is referred to [RFC6830], [RFC6831], [RFC6832], [RFC6833], [RFC6834], [RFC6835], [RFC6836], [RFC7052], [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: By decoupling the overloaded semantics of the current IP addresses the Internet core can be assigned identity meaningful addresses and hence, can use aggregation to scale. Devices are assigned with relatively opaque topologically meaningful addresses that are independent of their topological location.

- Overlay architecture: Overlays route packets over the current Internet, allowing deployment of new protocols without changing the current infrastructure hence, resulting into a low deployment cost.

- Decoupled data 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. While decoupled, data and control-plane 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 splits architecturally 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) or in the non-LISP Internet; see Section 3.5 for discussion of LISP site internetworking with non-LISP sites and domains in the Internet.

LISP sites (at the edge of the Internet) are connected to the core of the Internet by means of LISP-capable routers (e.g., border routers). LISP sites are connected across the 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 which 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.

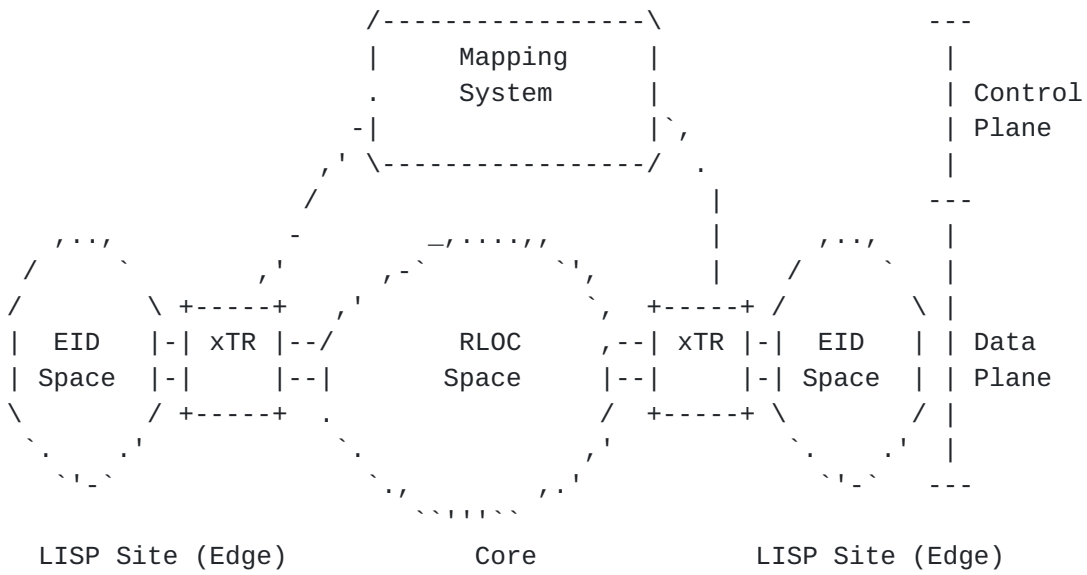


Figure 1.- A schema of the LISP Architecture

With LISP, the core uses RLOCs, an RLOC is an IPv4 or IPv6 address assigned to an Internet-facing network interface of an ITR or ETR. Typically RLOCs are numbered from topologically aggregatable blocks assigned to a site at each point to which it attaches to the global Internet, the topology is defined by the connectivity of networks.

A database which 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 traffic engineering 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, endhosts as well as intra and inter-domain routers remain unchanged, and the only required changes to the existing infrastructure are to routers connecting the EID with the RLOC space. Additionally, LISP requires the deployment of an independent Mapping System, such 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.

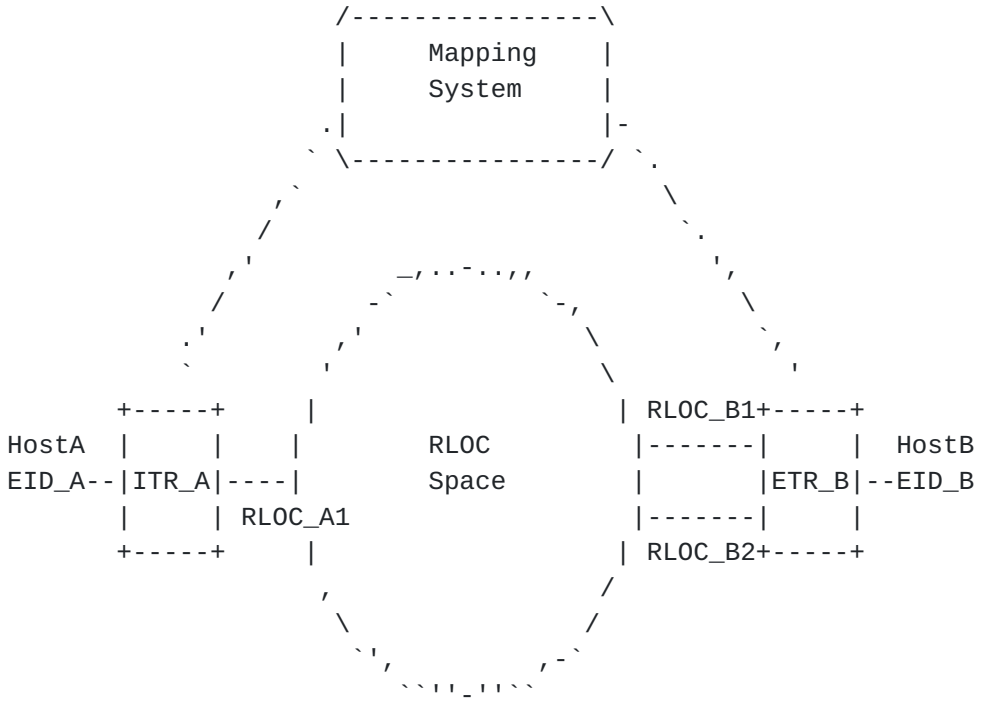


Figure 2.- Packet flow sequence in LISP

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 routed 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 along with traffic engineering policies: priority and weight per locator. Note that a Map-Reply can contain more locators if needed. ITR_A also stores the mapping in a local cache 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 by the Internet core as a normal IP packet, making the EID invisible from the Internet 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 [RFC6830]. 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 to

be consistent in case of multi-path 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 is not verified when it is received, because LISP data packets typically include an inner transport protocol header with a non-zero checksum. By omitting the additional outer UDP encapsulation checksum, xTRs can forward packets more efficiently. 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 striped 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 that 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) 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 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 encapsultor (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 Reencapsulating 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 at [RFC6830].

### 3.3.2. LISP Forwarding State

In the LISP architecture, ITRs keep just enough information to route traffic flowing through them. Meaning that, ITRs 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 Map-Cache for subsequent packets addressed to the same EID prefix. Note that, in case of overlapping EID-prefixes, following a single request, the ITR may receive a set of mappings, covering the requested EID-prefix and all more-specifics (cf., Section 6.1.5 [RFC6830]). 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 [RFC6833], provides a standard interface to register and request mappings. The LISP Mapping System is a database that stores such mappings. The following first describes 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 set of RLOCs as well as traffic engineering 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 IPs. IPv4 and IPv6 addresses are encoded using the appropriate Address Family Identifier (AFI) [RFC3232]. However LISP can also support more general address encoding by means of the ongoing effort around the LISP Canonical Address Format (LCAF) [I-D.ietf-lisp-lcaf].

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 MAC addresses, geo-coordinates, 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 [RFC6833] 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 and 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 (i.e., 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 which 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 natively (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-plane 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-plane 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 non-hierarchical 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 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+ALT [RFC6836], hierarchical databases in the form of LISP-DDT [I-D.ietf-lisp-ddt], monolithic databases in the form of LISP-NERD [RFC6837], flat databases in the form of LISP-DHT [I-D.cheng-lisp-shdht],[Mathy] and, a multicast-based database [I-D.curran-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 focuses on the two mapping systems that have been implemented and deployed (LISP-ALT and LISP+DDT).

#### 3.4.3.1. LISP+ALT

The LISP Alternative Topology (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 participated by Map-Servers and Map-Resolvers. 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 that in turn, replies directly to the ITR using the native Internet core.

#### 3.4.3.2. LISP-DDT

LISP-DDT [I-D.ietf-lisp-ddt] 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 [DDT-ROOT], which is a particular instance of a DDT node and 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 either the DDT root and/or other DDT nodes.

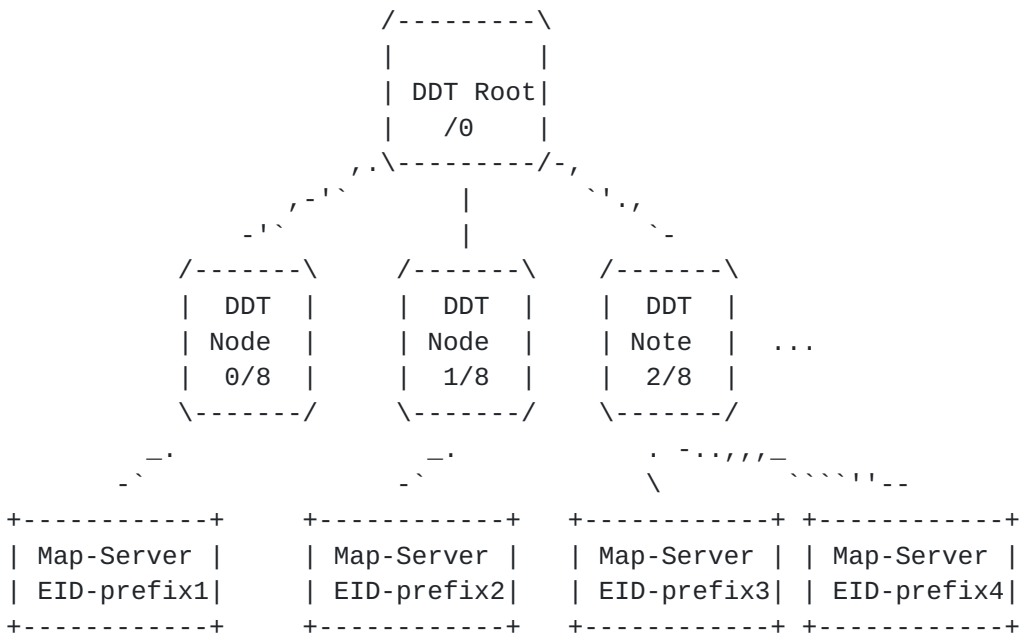


Figure 3.- A schematic representation of the DDT tree structure, please note that the prefixes and the structure depicted should be only considered as an example.

The DDT structure does not actually index EID-prefixes but eXtended EID-prefixes (XEID). An XEID-prefix is just the concatenation of the following fields (from most significant bit to less significant bit): 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 mapping retrieving latency[Jakab].

The DDT Mapping System relies on manual configuration. That is Map-Resolvers are manually configured with the set of available DDT root nodes while DDT nodes are manually 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 traffic engineering 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 Internet global routing system. 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 in 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-plane, where mappings are stored in the control-plane and used for forwarding in the data plane, requires 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 traffic engineering policies (priorities and weights).

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

Time-To-Live (TTL): Each mapping 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. Typical values for TTL defined by LISP are 24 hours.

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 a SMR message, the ITR must refresh the bindings by sending a Map-Request to the Mapping System. Further uses of SMRs are documented in [RFC6830].

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 to obtain the newest

mappings. On the contrary, if it detects that the remote xTR Map-Cache is outdated, it sends a SMR to notify it that a new mapping is available.

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 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 (LSB): LSB is a passive technique, the LSB field is carried by data-packets in the LISP header and can be set by a 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 LSB does not provide path reachability status, only hints on the status of RLOCs.

Echo-nonce: This is also a passive technique, that can only operate effectively when data flows bi-directionally between two communicating xTRs. Basically, an ITR piggybacks a random number (called 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.

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 waiting in return a Map-Reply, also sent on the data-plane. The active nature of RLOC-probing provides an effective mechanism to determine 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 Echo-nonce can work together. Specifically if a nonce is not echoed, an ITR could RLOC-probe to determine if the path is up when it cannot tell the difference between a failed bidirectional path or the return path is not used (a unidirectional path).

Additionally, LISP also recommends inferring reachability of locators by using information provided by the underlay, in particular:

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

Underlay routing: Both BGP and IBGP carry reachability information, LISP-capable routers that have access to underlay routing information can use it to determine if a given locator or path are 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 PMTUD [RFC1191] or 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 a ICMP Too Big message to the source.

## 5. Mobility

The separation between locators and identifiers in LISP is suitable for traffic engineering purpose 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 the device changes of RLOC, the xTR updates the RLOC of its local mapping and registers it to its Map-Server, typically with a low TTL value (1min). To avoid the need of 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 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 allows 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 operational changes required 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 unicast encapsulate PIM Join/Prune messages from receiver to source sites. At the core, ETRs build a new PIM Join/Prune message addressed to the RLOC of the ITR servicing the source. An 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 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 outter 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 address are logical and are not resolved by the mapping system. Then the multicast packet is transmitted through the core towards the receiving ETRs that decapsulates the packets and sends them using the receiver's site multicast state.

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

## 7. Use Cases

### 7.1. Traffic Engineering

A LISP site can strictly impose via which ETRs the traffic must enter the 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 to 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 to 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 indicates which RLOCs should be preferred to send packets (the least preferred ones being provided for backup purpose). The weight permits to balance the load between the RLOCs with the same priority, proportionally to the weight value.

As mappings are directly issued by the authoritative ETR of the EID and are not altered while transmitted to the remote site, it offers highly flexible incoming inter-domain traffic engineering with even the possibility for a site to support a different mapping policy for each remote site. routing policies.

### 7.2. LISP for IPv6 Co-existence

LISP encapsulations allows to transport 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 family 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 center is done without any form of translation as the native IPv6 packets (in the EID space) will be LISP encapsulated and transmitted over the IPv4 legacy Internet by the mean of 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, it is essential to distinguish which virtual network a packet belongs and 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 the VRF or 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 is running IP.

### 7.4. LISP for Virtual Machine Mobility in Data Centers

A way to enable seamless virtual machine mobility in data center is to conceive the datacenter 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 virtual machine is moved to another subnet, it can keep (temporarily) the address it had before the move so 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 to the LISP protocol.

While in a push mapping system, the state necessary to forward packets is learned independently of the traffic itself, with a pull architecture, the system becomes reactive and data-plane events (e.g., the arrival of a packet for an unknown destination) 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 order of magnitude can be observed between the slow and the fast paths. As a consequence, the way data-plane events are notified to the control-plane must be thought carefully so to not overload the slow path and rate limiting should be used as specified in [RFC6830].

Care must also be taken so to not 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 [RFC6830] recommends to rate limit the sending of messages to the mapping system.

To improve resiliency and reduce the overall number of messages exchanged, LISP offers the possibility to leak information, such as reachabilty of locators, directly into data plane packets. In

environments that are not fully trusted, control information gleaned from data-plane packets should be verified before using them.

Mappings are the centrepiece of LISP and all precautions must be taken to avoid them to be manipulated or misused by malicious entities. Using trustable Map-Servers that strictly respect [RFC6833] and the lightweight authentication mechanism proposed by LISP-Sec [I-D.ietf-lisp-sec] reduces the risk of attacks to the 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, the 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 incomplete view on 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 of the destination EID of the packet or even to a network element that is not an ETR. The mitigation consists in applying appropriate filtering techniques on the network elements that can potentially receive un-expected LISP-encapsulated packets

More details about security implications of LISP are discussed in [I-D.ietf-lisp-threats].

## 9. IANA Considerations

This memo includes no request to IANA.

## 10. Acknowledgements

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, David Black as well as every people acknowledged in [RFC6830].

## 11. References

### 11.1. Normative References

[I-D.ietf-lisp-ddt]
Fuller, V., Lewis, D., Ermagan, V., and A. Jain, "LISP Delegated Database Tree", draft-ietf-lisp-ddt-02 (work in progress), October 2014.

[I-D.ietf-lisp-lcaf]
Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format (LCAF)", draft-ietf-lisp-lcaf-07 (work in progress), December 2014.

[I-D.ietf-lisp-sec]
Maino, F., Ermagan, V., Cabellos-Aparicio, A., and D. Saucez, "LISP-Security (LISP-SEC)", draft-ietf-lisp-sec-07 (work in progress), October 2014.

[I-D.ietf-lisp-threats]
Saucez, D., Iannone, L., and O. Bonaventure, "LISP Threats Analysis", draft-ietf-lisp-threats-12 (work in progress), March 2015.

[RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, November 1990.

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

[RFC2992] Hopps, C., "Analysis of an Equal-Cost Multi-Path Algorithm", RFC 2992, November 2000.

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

[RFC3963] Devarapalli, V., Wakikawa, R., Petrescu, A., and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol", RFC 3963, January 2005.

[RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, March 2007.

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

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

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

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, January 2013.

[RFC6831] Farinacci, D., Meyer, D., Zwiebel, J., and S. Venaas, "The Locator/ID Separation Protocol (LISP) for Multicast Environments", RFC 6831, 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.

[RFC6834] Iannone, L., Saucez, D., and O. Bonaventure, "Locator/ID Separation Protocol (LISP) Map-Versioning", RFC 6834, January 2013.

[RFC6835] Farinacci, D. and D. Meyer, "The Locator/ID Separation Protocol Internet Groper (LIG)", RFC 6835, January 2013.

[RFC6836] Fuller, V., Farinacci, D., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol Alternative Logical Topology (LISP+ALT)", RFC 6836, January 2013.

[RFC6837] Lear, E., "NERD: A Not-so-novel Endpoint ID (EID) to Routing Locator (RLOC) Database", RFC 6837, January 2013.

[RFC6935] Eubanks, M., Chimento, P., and M. Westerlund, "IPv6 and UDP Checksums for Tunneled Packets", RFC 6935, April 2013.

[RFC6936] Fairhurst, G. and M. Westerlund, "Applicability Statement for the Use of IPv6 UDP Datagrams with Zero Checksums", RFC 6936, April 2013.

[RFC7052] Schudel, G., Jain, A., and V. Moreno, "Locator/ID Separation Protocol (LISP) MIB", RFC 7052, October 2013.

[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, April 2014.

### 11.2. Informative References

[DDT-ROOT]
LISP DDT ROOT, , "http://ddt-root.org/", August 2013.

[I-D.cheng-lisp-shdht]
Cheng, L. and J. Wang, "LISP Single-Hop DHT Mapping Overlay", draft-cheng-lisp-shdht-04 (work in progress), July 2013.

[I-D.curran-lisp-emacs]
Brim, S., Farinacci, D., Meyer, D., and J. Curran, "EID Mappings Multicast Across Cooperating Systems for LISP", draft-curran-lisp-emacs-00 (work in progress), November 2007.

[Jakab] Jakab, L., Cabellos, A., 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", October 2010.

[Mathy] Mathy, L., Iannone, L., and O. Bonaventure, "LISP-DHT: Towards a DHT to map identifiers onto locators. The ACM ReArch, Re-Architecting the Internet. Madrid (Spain)", December 2008.

[Quoitin] Quoitin, B., Iannone, L., Launois, C., and O. Bonaventure, ""Evaluating the Benefits of the Locator/Identifier Separation" in Proceedings of 2Nd ACM/IEEE International Workshop on Mobility in the Evolving Internet Architecture", 2007.

## 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. LISP at this point 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, and after numerous revisions, the basic specifications moved to becoming RFCs at the start of 2013 (although work to expand and improve it, and find new uses for it, continues, and undoubtly will for a long time to come).

## A.1. Old LISP Models

LISP, as initially conceived, had a number of potential operating modes, named 'models'. Although they are no 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 'bootstrap' operation, by using this to load EID->RLOC mappings. Packets were sent with the EID as the destination in the outer wrapper; when an ETR saw such a packet, it would send a Map-Reply to the source ITR, giving the full mapping.

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

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

LISP 3: EIDs are not routable; and have to be looked up in in a new EID->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 they need, as needed.

Authors' Addresses

Albert Cabellos
UPC-BarcelonaTech
c/ Jordi Girona 1-3
Barcelona, Catalonia 08034
Spain

Email: acabello@ac.upc.edu

Damien Saucez (Ed.)
INRIA
2004 route des Lucioles BP 93
Sophia Antipolis Cedex 06902
France

Email: damien.saucez@inria.fr