LISP: An Architectural Solution to Multi-homing, Traffic Engineering, and Internet Route Scaling

Dave Meyer & Dino Farinacci

LISP Designers:
Dave Meyer, Vince Fuller, Darrel Lewis, Andrew Partan, John Zwiebel, Scott Brim, Noel Chiappa & Dino Farinacci
Separating (or adding) an Address

Changing the semantics of the IP address


IPv4: 209.131.36.158.10.0.0.1

If PI, get new locator
If PA, get new ID
Why the Separation?

- **Level of Indirection** allows us to:
  - Keep either ID or Location **fixed** while **changing** the other
  - Create **separate namespaces** which can have different allocation properties

- **By keeping IDs fixed**
  - Assign fixed addresses that never change to hosts and routers at a site

- **You can change Locators**
  - Now the sites can change providers
  - Now the hosts can move
LISP creates a **Level of indirection** with two namespaces: **EID** and **RLOC**

- **EID (Endpoint Identifier)** is the IP address of a host – just as it is today
- **RLOC (Routing Locator)** is the IP address of the LISP router for the host
- **EID-to-RLOC mapping** is the distributed architecture that maps EIDs to RLOCs

- Network-based solution
- No host changes
- Minimal configuration
- Incrementally deployable
- Support for mobility
- Address Family agnostic
Some Brief Definitions

- **IDs or EIDs**
  - End-site addresses for hosts and routers at the site
  - They go in DNS records
  - Generally not globally routed on underlying infrastructure
  - New namespace

- **RLOCs or Locators**
  - Infrastructure addresses for LISP routers and ISP routers
  - Hosts do not know about them
  - They are globally routed and aggregated along the Internet connectivity topology
  - Existing namespace
Multi-Level Addressing

Provider A
10.0.0.0/8

Provider B
11.0.0.0/8

RLOCs used in the core

EIDs are inside of sites
What is LISP?

- Locator/ID Separation Protocol
- Ground rules for LISP
  - Network-based solution
  - No changes to hosts whatsoever
  - No new addressing changes to site devices
  - Very few configuration file changes
  - Imperative to be incrementally deployable
  - Address family agnostic
IP encapsulation scheme

- Decouples host **IDENTITY** and **LOCATION**
- Dynamic **IDENTITY**-to-**LOCATION** mapping resolution
- Address Family agnostic day-one

**Minimal Deployment Impact**

- No changes to end systems or core
- Minimal changes to edge devices

**Incrementally deployable**

- LISP/LISP and non-LISP/LISP considered day-one
LISP Overview

LISP Map Lookup is analogous to a DNS lookup

- DNS resolves IP addresses for URLs

- LISP resolves locators for queried identities

---

LISP router

[ where is 2610:D0:110C:1::3 ]

[ location is 128.107.81.169 ]

LISP Mapping System

LISP Identity-to-location Map Resolution

---

host

[ who is lisp.cisco.com ]

[ 153.16.5.29, 2610:D0:110C:1::3 ]

DNS Server

DNS URL Resolution

---

LISP Overview – 7
What is LISP?

“Jack-Up” or “Map-n-Encap”
<table>
<thead>
<tr>
<th></th>
<th>Version</th>
<th>IHL</th>
<th>Type of Service</th>
<th>Total Length</th>
<th>Identification</th>
<th>Flags</th>
<th>Fragment Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>OH</td>
<td>Time to Live</td>
<td>Protocol = 17</td>
<td>Header Checksum</td>
<td>Source Routing Locator</td>
<td>Destination Routing Locator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UDP</td>
<td>Source Port</td>
<td>Dest Port (4341)</td>
<td>UDP length</td>
<td>UDP Checksum</td>
<td>Locator Reach Bits</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LISP</td>
<td>Nonce</td>
<td>Source EID</td>
<td>Destination EID</td>
<td>-----------------------</td>
<td>-----------------------</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### LISP Architecture

**Nanog 45 - Jan 2009**

Slide 14
LISP for IPv6 Transition

Legend:
- EIDs -> Green
- Locators -> Red
What is LISP?

• Data plane
  - Design for encapsulation and tunnel router placement
  - Design for locator reachability
  - Data-triggered mapping service

• Control plane
  - Design for a scalable mapping service
  - Examples are: CONS, NERD, ALT, EMACS
LISP Network Elements

- **Ingress Tunnel Router (ITR)**
  - Finds EID to RLOC mapping
  - Encapsulates to Locators at source site
- **Egress Tunnel Router (ETR)**
  - Owns EID to RLOC mapping
  - Decapsulates at destination site
- **xTR**
  - Term used when not referring to directionality
  - Basically a LISP router
Unicast Packet Forwarding

PI EID-prefix 1.0.0.0/8

- S1
- S2

1.0.0.1 \rightarrow 2.0.0.2

11.0.0.1 \rightarrow 12.0.0.2

DNS entry:
D.abc.com A 2.0.0.2

Legend:
EIDs \rightarrow Green
Locators \rightarrow Red

EID-prefix: 2.0.0.0/8
Locator-set:
- 12.0.0.2, priority: 1, weight: 50 (D1)
- 13.0.0.2, priority: 1, weight: 50 (D2)

Policy controlled by destination site
When the xTR has no Mapping

- Need a scalable EID to Locator mapping lookup mechanism
- Network based solutions
  - Have query/reply latency
  - Can have packet loss characteristics
  - Or, have a full table like BGP does
- How does one design a scalable Mapping Service?
Mapping Database Designs

• You need a “map” before you can “encap”
• We have designed several mapping database protocols
  - CONS, NERD, EMACS, ALT
  - Tradeoff push versus pull benefit/cost
  - Needs to be scalable to $10^{10}$ entries
• ALT has the most promise
  - We are deploying ALT
What is LISP+ALT?

- EID namespace is used at the site
- RLOC namespace is used in the Internet core
- Mappings need to be authoritative and reside at site ETRs
- Advertise EID-prefixes in BGP on an alternate topology of GRE tunnels
- ITRs get mappings by routing Map-Requests on ALT topology
- ETRs respond with Map-Replies
How LISP+ALT Works

Legend:
- EIDs -> Green
- Locators -> Red
- GRE Tunnel
- Low Opex
- Physical link
- Data Packet
- Map-Request
- Map-Reply

EID-prefix 240.0.0.0/24

240.0.0.1 -> 240.1.1.1

11.0.0.1 -> 240.1.1.1

240.0.0.1 -> 240.1.1.1

11.0.0.1 -> 240.1.1.1

240.0.0.1 -> 240.1.1.1

240.0.0.1 -> 240.1.1.1

EID-prefix 240.1.0.0/16

1.1.1.1 -> 240.1.1.1

240.0.0.1 -> 240.1.1.1

11.0.0.1 -> 240.1.1.1

11.0.0.1 -> 240.1.1.1

240.0.0.1 -> 240.1.1.1

240.0.0.1 -> 240.1.1.1

240.0.0.1 -> 240.1.1.1

240.0.0.1 -> 240.1.1.1

EID-prefix 240.1.1.0/24

240.0.0.1 -> 240.1.1.1

EID-prefix 240.1.2.0/24

240.0.0.1 -> 240.1.1.1

EID-prefix 240.2.1.0/24

12.0.0.1

11.0.0.1

2.2.2.2

3.3.3.3

11.0.0.1

1.1.1.1

11.0.0.1

11.0.0.1

11.0.0.1

11.0.0.1

11.0.0.1
Locator Reachability

EID-prefix: 2.0.0.0/8

Legend:
EIDs -> Green
Locators -> Red

Locator-set:
- 12.0.0.2, priority: 1, weight: 50 (D1) -> ordinal 0
- 13.0.0.2, priority: 1, weight: 50 (D2) -> ordinal 1

Loc-reach-bits: 0x0000 0003

b'xxxx xxxx' 7654 3210

LISP Architecture
LISP Interworking

- LISP will not be widely deployed day-1
- Need a way for LISP-capable sites to communicate with rest of Internet
- Two basic Techniques
  - LISP Network Address Translators (LISP-NAT)
  - Proxy Tunnel Routers (PTRs)
**LISP Interworking**

- These combinations must be supported
  - Non-LISP site to non-LISP site
    - Today’s Internet
  - LISP site to LISP site
    - Encapsulation over IPv4 makes this work
    - IPv4-over-IPv4 or IPv6-over-IPv4
  - LISP-R site to non-LISP site
    - When LISP site has PI or PA routable addresses
  - LISP-NR site to non-LISP site
    - When LISP site has PI or PA non-routable addresses
Interworking using LISP-NAT

Legend:
LISP Sites -> Green (and EIDs)
non-LISP Sites -> Red (and RLOCs)

xTR

Local/Uncoordinated Solution
Interworking using PTRs

1. 65.1.1.1 -> 1.1.1.1
2. 65.9.1.1 -> 66.1.1.1
3. 1.1.1.1 -> 65.1.1.1

R-prefix 65.1.0.0/16
R-prefix 65.2.0.0/16
R-prefix 65.3.0.0/16

NR-prefix 1.1.0.0/16
NR-prefix 1.2.0.0/16
NR-prefix 1.3.0.0/16

Infrastructure Solution

Legend:
LISP Sites -> Green (and EIDs)
non-LISP Sites -> Red (and RLOCs)

xTR

LISP Architecture
Nanog 45 - Jan 2009
Slide 31
LISP Use Cases
VM-Mobility

Needs:
- VM-Mobility across subnets
- Move detection, dynamic EID-to-RLOC mappings, traffic redirection

LISP Solution:
- OTV + LISP to extend subnets
- LISP for VM-moves across subnets

Benefits:
- Integrated Mobility
- Direct Path (no triangulation)
- Connections maintained across moves
- No routing re-convergence
- No DNS updates required
- Global Scalability (cloud bursting)
- IPv4/IPv6 Support
- ARP elimination

Applicability:
- VM OS agnostic
- Services Creation (disaster recovery, cloud burst, etc.)
LISP Use Cases
LISP-MN

Needs:
- Mobile devices roaming across any access media without connection reset
- Mobile device keeps the same IP address forever

LISP Solution:
- LISP level or indirection separates endpoints and locators
- Network-based; no host changes, minimal network changes
- Scalable, host-level registration ($10^{10}$)

Benefits:
- MNs can roam and stay connected
- MNs can be servers
- MNs roam without DNS changes
- MNs can use multiple interfaces
- Packets have “stretch-1” reducing latency

Applicability:
- IPv4 and IPv6
- Android and Linux
- Open
Prototype Implementation

- **cisco** has a LISP prototype implementation
- **Supports:**
  
  - draft-farinacci-lisp-11.txt
  - draft-fuller-lisp-alt-03.txt
  - draft-lewis-lisp-interworking-02.txt

- **Software switching only**
- **Supports LISP for both IPv4 and IPv6**
  - ITR, ETR, and PTR
  - LISP-NAT for IPv4 only
Internet Drafts

draft-farinacci-lisp-11.txt
draft-farinacci-lisp-multicast-01.txt
draft-fuller-lisp-alt-03.txt
draft-lewis-lisp-interworking-02.txt
draft-meyer-lisp-eid-block-01.txt
draft-meyer-loc-id-implications-01.txt

draft-mathy-lisp-dht-00.txt
draft-ianonne-openlisp-implementation-02.txt
draft-brim-lisp-analysis-00.txt

draft-meyer-lisp-cons-04.txt
draft-lear-lisp-nerd-04.txt
draft-curran-lisp-emacs-00.txt
References

• **Public mailing list:**
  lisp@ietf.org

• **More info at:**
  http://www.lisp4.net
  http://www.lisp6.net