# OpenLISP: An Open Source Implementation of the Locator/ID Separation Protocol* †

Damien Saucez
Université catholique de Louvain
damien.saucez@uclouvain.be

Luigi Iannone
TU Berlin / Deutsche Telekom Laboratories
luigi@net.t-labs.tu-berlin.de

Olivier Bonaventure
Université catholique de Louvain
olivier.bonaventure@uclouvain.be

## ABSTRACT

The network research community has recently started to work on the design of an alternate Internet Architecture aiming at solving some scalability issues that the current Internet is facing. The Locator/ID separation paradigm seems to well fit the requirements for this new Internet Architecture. Among the various solutions, LISP (Locator/ID Separation Protocol), proposed by Cisco, has gained attention due to the fact that it is incrementally deployable.

In the present paper we give a short overview on OpenLISP, an open-source implementation of LISP. Beside LISP's basic specifications, OpenLISP provides a new socket-based API, namely the *Mapping Sockets*, which makes OpenLISP an ideal experimentation platform for LISP, but also other related protocols.

## Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Network communications; C.2.6 [**Internetworking**]: Routers; C.2.5 [**Local and Wide-Area Network**]: Internet

## General Terms

Algorithms, Management, Design.

## Keywords

LISP, Internet Architecture, Routing, Addressing.

## 1. INTRODUCTION AND MOTIVATION

The idea of improving the Internet Architecture with some form of separation between the identity of end-systems and their location in the Internet topology is not new [1]. But the Locator/ID separation paradigm has recently gained momentum due to the increasing scalability issues with the current Internet Architecture [2]. The Locator/ID separation paradigm has several implications, concerning the necessity to map IDs into locators, storing and distributing these mappings, and perform tunneling or address translation operations in order to forward packets in the core Internet.

Among the different Locator/ID split proposals, the Locator/ID Separation Protocol (LISP), proposed by Farinacci et al. [3], and based on a map-and-encap approach, has the main advantage of being incremental deployable on border routers of edge networks, thus limiting the number of systems that need to be upgraded.

OpenLISP, our open-source implementation of LISP, aims at providing an open and flexible platform for experimentation [4]. To this end, with OpenLISP we went further than the LISP specifications. LISP has a detailed description of the encapsulation and decapsulation operations, the forwarding operation, and offers several options as mapping system. Nevertheless there is no specification of an API to allow the mapping system to interact with the forwarding engine. In OpenLISP we proposed and implemented a new socket based solution in order to overcome this issue: the *Mapping Sockets*. Mapping sockets make OpenLISP an open and flexible solution, where different approaches for the locator/ID separation paradigm can be experimented, even if not strictly related to LISP. To the best of our knowledge, OpenLISP is the only existing effort in developing an open source Locator/ID separation approach. The development and the experimentation done with OpenLISP had also an impact on the original LISP specifications, allowing to correct some original design shortcomes and improve some engineering solutions [4].

## 2. LISP IN A NUTSHELL

LISP is meant to be deployed on border routers whose upstream IP address is used as Routing LOCator (RLOC) for the end-systems of the local domain. End-systems still communicate using legacy IP addresses, which in the LISP terminology are called Endpoint IDentifiers (EIDs). EIDs and RLOCs are both IP addresses, however, while EIDs have only a local scope, thus not routable outside the local domain, RLOCs are only used for inter-domain routing and cannot be used as endpoint identifiers for host-to-host connections. EIDs can be associated to a set of RLOCs, since a domain can be multi-homed, *i.e.*, having several border routers. LISP tunnels the packets in the core Internet, using an IP-over-UDP approach, from one of the RLOCs of the source EID to one of the RLOCs of the destination EID. To perform such a tunneling, LISP needs to know when to encapsulate or decapsulate a packet and what to put exactly in the header. For this purpose, LISP uses two data structures: the LISP Database and the LISP Cache. The LISP Database is used to select the source RLOC for outgoing packets and to determine whether an incoming packet needs to be decapsulated. It consists of all EID-Prefix-to-RLOC mappings that are "owned locally". A LISP router owns a mapping if its upstream interface (toward the provider), is in the set of RLOCs associated to the EID-Prefix used as

†This work has been partially supported by Cisco and the European Commission within the the INFSO-ICT-216372 TRILOGY and ECODE projects.

*A similar Demo has been presented in the demo session of the IEEE INFOCOM 2009 Conference.

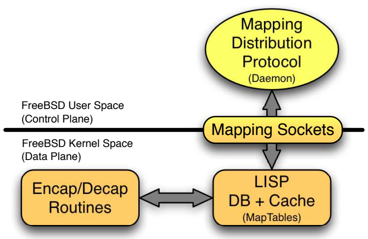


**Figure 1: OpenLISP Architecture.**

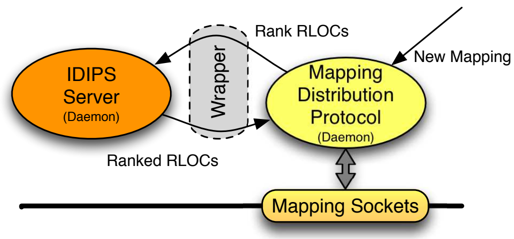


**Figure 2: Eample of interaction of OpenLISP and IDIPS.**

addressing space downstream (*i.e.*, inside the local domain). The LISP Cache is used to select the destination RLOC for outgoing packets and contains mappings for EID-Prefixes that are not local. The LISP Cache is fed in an on-demand fashion. When a packet generates a cache-miss, the mapping system is queried to retrieve the missing mapping. Mapping systems provide a lookup infrastructure usualy with a Mapping Distribution Protocol. There are several of such protocols proposed insofar. A list as well as a comparison can be found in [5].

There is a fair amount of activity on LISP in both the IRTF and the IETF. Cisco, in collaboration with other companies and research institutes, has already deployed its implementation on a testbed (*http://www.lisp4.net*) scattered around the world, using the LISP-ALT mapping system [6].

# 3. OVERVIEW OF OPENLISP

OpenLISP, whose high-level architecture is depicted in Fig. 1, is our implementation of LISP in the FreeBSD operating system. The forwarding engine of OpenLISP, which includes functions for encapsulation and decapsulation, has been implemented directly in the kernel, along with both LISP's cache and database.

Concerning the mapping system, OpenLISP does not provide any specific Mapping Distribution Protocol. The motivation for this choice is that our aim was to develop a flexible and extensible platform providing support for future experimentation of both new and existing Mapping Distribution Protocols. Nonetheless, we provided OpenLISP with some simple tools, namely `map` and `mapstat`, to have access and to control OpenLISP from a shell terminal.

The interaction between user space and kernel space is possible thanks to the new socket API that we developed in OpenLISP, namely the *Mapping Sockets*. Mapping sockets allow Mapping Distribution Protocols (or tools like `map` and `mapstat`) running in the user space to send messages to the kernel space in order to perform operations and modify kernel's data structure and receive confirmation messages. Moreover, mapping sockets also offer signaling functionality the other way around, allowing the kernel to notify daemons in user space about events related to LISP (*e.g.*, cache-miss).

# 4. IDIPS USE CASE

Besides the development and the evaluation of mapping distribution systems, OpenLISP can be used for other purposes. When an identifier is reachable via several routing locators, a LISP router that needs to encapsulate packets towards this identifier can select any of these locators. The remote LISP routers may indicate preferences among the locators to control the incoming packets. This is done by using the priority field that LISP associate to each RLOC. However, it can be expected that in many cases a site will associate the same priority to its different locators. In this case, the encapsulating router will be free to select its preferred locator to reach this identifier.

Selecting the best path towards each remote LISP router is difficult for a router. For this reason, ISP Driven Informed Path Selection (IDIPS) has been proposed [7]. IDIPS is a service that ranks paths provided by a client. Integrating IDIPS with OpenLisp is straightforward with the mapping socket. Neither IDIPS nor LISP need to be explicitely aware of each other. As shown in Fig. 2, a wrapper installed on the OpenLISP machine can make the link between LISP and IDIPS. When a mapping is installed in the LISP cache, its list of locators is sent to the wrapper. The wrapper then sends a request to the IDIPS server to rank the locators. When the wrapper receives the answer from the IDIPS, it translates the ranking into OpenLISP list of RLOCs with priorities and updates the priorities in the LISP cache via the mapping socket. The new choice of RLOCs is then available for the next encapsulation.

# 5. CONCLUSION

Work is ongoing to integrate the Cisco testbed with OpenLISP boxes. The advantage of OpenLISP is that it is open for development and experimenting new mapping systems, as well as traffic engineering solutions like IDIPS. OpenLISP is freely available and can be downloaded from:
*http://inl.info.ucl.ac.be/softwares/openlisp*

# 6. REFERENCES


[1] R. Hiden, "New Scheme for Internet Routing and Addressing (ENCAPS) for IPNG," IETF, RFC 1955.

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

[3] D. Farinacci, V. Fuller, D. Oran, and D. Meyer, "Locator/ID Separation Protocol (LISP)," IETF, Internet Draft *draft-ietf-lisp-01.txt*, May 2009.

[4] L. Iannone, D. Saucez, and O. Bonaventure, "OpenLISP Implementation Report," IETF *draft-iannone-openlisp-implementation-01.txt*, July 2008.

[5] L. Mathy and L. Iannone, "LISP-DHT: Towards a DHT to map identifiers onto locators," in *Proc. of ReArch'08 - Re-Architecting the Internet.*, Dec. 2008.

[6] D. Farinacci, V. Fuller, and D. Meyer, "LISP Alternative Topology (LISP-ALT)," IETF, Internet Draft *draft-ietf-lisp-alt-01.txt*, May 2009.

[7] D. Saucez, B. Donnet, L. Iannone, and O. Bonaventure, "Interdomain Traffic Engineering in a Locator/Identifier Separation Context," in *Proc. INM'08*, Oct. 2008.