



LISP-Views: Monitoring LISP at Large Scale

Yue Li, Akram Abouseif, Damien Saucez, Luigi Iannone

► **To cite this version:**

Yue Li, Akram Abouseif, Damien Saucez, Luigi Iannone. LISP-Views: Monitoring LISP at Large Scale. Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication, May 2018, Roscoff, France. <hal-01784729>

HAL Id: hal-01784729

<https://hal.archives-ouvertes.fr/hal-01784729>

Submitted on 3 May 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

LISP-Views: Monitoring LISP at Large Scale

Yue Li¹ et Akram Abouseif² et Luigi Iannone¹ et Damien Saucez³

¹INFRES, Telecom ParisTech – France

²COMELEC, Telecom ParisTech – France

³Université Côte d’Azur, Inria – France

Le protocole LISP (Locator/Identifier Separation Protocol) découple les rôles d’identifiants et de locations des adresses IP. Afin de pouvoir associer des localisateurs à des identifiants, LISP repose sur un ”système de mapping”. Deux plateformes expérimentales – LISP Beta Network et LISP-Lab – déploient déjà de tels systèmes de mapping LISP dans l’Internet. Toutefois, à l’heure actuelle, seule la plateforme LISP Beta Network est monitorée en reposant sur LISPmon qui échantillonne le système de mapping de LISP Beta Network une fois par jour. Afin d’accompagner la croissance de LISP, il est nécessaire de déployer un système de monitoring complet. C’est ainsi que nous proposons la solution ”LISP-Views”, une architecture polyvalente de monitoring à grande échelle pour LISP. LISP-Views permet d’effectuer ces mesures de manière automatique. Après avoir fait fonctionner LISP-Views pendant plusieurs mois et comparé les résultats avec ceux de LISPmon, nous parvenons à la conclusion que les résultats de LISP-Views sont plus précis et détaillés. Entre autres, LISP-Views a permis d’observer différents comportements transitoires au sein du système de mapping qui n’apparaissent pas avec LISPmon. Cette connaissance plus fine de LISP promet de futures améliorations de performances pour LISP.

Mots-clefs : LISP, monitoring architecture, experiment, measurement

1 Introduction

The Locator/Identifier Separation Protocol (LISP) is currently standardized at the IETF to overcome the flexibility issue of current Internet [Fea13]. It separates the IP address space into identifier locator. Then it uses a map-and-encap mechanism in order to ensure end-to-end communication. Thus, a new network entity called *Mapping Distribution System* (MDS) is introduced to store and distribute the bindings between EIDs and RLOCs. In this way, LISP is able to improve the Internet routing scalability, enable enhanced inter-domain traffic engineering, lightweight multi-tenant carrier-grade VPN, and VMs mobility in Data-Centers [SIBF12].

To promote the development of LISP and boost the related research, two LISP-related platforms : LISP Beta Network [lisa] and LISP-Lab platform [lisb] have been deployed and interconnected so far. Currently, a unique LISP monitoring system called LISPmon [lisc] supervises the global MDS and publishes the mapping information daily. However, it is known that the MDS is not always stable and consistent [LSID16]. We hence propose a dynamic versatile LISP monitoring architecture, LISP-Views, to overcome these limitations. LISP-Views automatically explores the whole MDS every 2 hours and stores the detailed mapping information, so to facilitate the experimenters to evaluate the LISP comprehensive performance.

We used a one-month long set of traces produced by LISP-Views to evaluate its performance and compare it with LISPmon. We show that LISP-Views is more accurate than LISPmon since it monitors all the MDS elements in parallel and provides more information.

2 LISP Background

2.1 LISP architecture overview

The *Locator/Identifier Separation Protocol* (LISP) splits the conventional IP addressing spaces into two logical sub-spaces : (i) *Endpoint Identifier (EID)* : the identifier of end host and is also used to locally route

within the local domain, i.e., within LISP-Site. (ii) *Routing LOCator (RLOC)* : refers to the interface of edge routers in the Internet topology, and is also used to route through the Internet core, i.e., between LISP-Sites. The hosts produce conventional IP packets by using their EIDs respectively as source and destination address. After the IP packets routing to edge routers, called xTR (the combination of Ingress and Egress Tunnel Router), it first checks in its mapping cache to find out the association between the destination EID and its RLOC. Otherwise, it sends a Map-Request to Map-Resolver (MR) of MDS to obtain the Map-Reply containing mapping information. It encapsulates the packets by adding the queried RLOC as the outer destination address and its own RLOC as the outer source address. Then, it sends the encapsulated packets through Internet. Once the destination xTR receives these packets, it decapsulates and forwards them to the destination host.

Based on LISP architecture, there are three types of Map-Reply can be received : (i) *LISP Map-Reply* : the queried IP address belongs to a LISP site and the Map-Reply contains the mapping information for this site. (ii) *Negative Map-Reply* : the prefix covering the queried IP address belongs to a non-LISP site, and the Map-Reply contains no mapping information. We consider these two kinds of replies as a successful query. (iii) *No Map-Reply* : the xTR does not receive any reply during 3 seconds. In this case, we consider the query failed.

2.2 Current LISP Deployment Status

Two experimental LISP testbeds have been deployed and interconnected so far. One is the LISP Beta Network, which is deployed in 2008 and the other one is LISP-Lab platform, which is opened since 2015. According to the latest architecture, LISP Beta Network has 12 MRs and LISP-Lab platform has 2 MRs. Besides, they also have all necessary LISP network entities, such as : xTRs, MSes and PxTRs.

LISPmon is the only monitoring platform supervising the current LISP status. It scans the whole IPv4 addressing space everyday, normally begins at 7 :00 a.m. (UTC), and queries them by sending the Map-Request to 1 specific MR only of the LISP Beta network. If this MR does not function normally, it will choose another one as a replacement, and will start the queries again, to guarantee that the receiving of LISP Map-Replies changes smoothly between two days. The results of the available mapping information are published once per day, ever since beginning of 2010.

3 Proposed Monitoring Architecture

3.1 Motivation

LISPmon was the first step towards a systematic LISP monitoring. However, it monitors the MDS just from one vantage point (VP) once per day and only queries one MR. Upon MR issues, LISPmon must be manually re-configured to monitor another MR. Yet the mapping information sometimes dramatically changes within a day, and the Map-Replies from the different MRs may not coincide at a given time [LSID16]. From such point of view, LISPmon has strong limitations since it is not able to detect the changes of mapping information within one day, and is not able to show the differences among MRs. Thus, we propose a new versatile LISP monitoring architecture, called LISP-Views, to monitor public LISP deployments, as well as to enable further performance evaluation of LISP defined by the users themselves.

3.2 Description of LISP-Views

The architecture of the proposed LISP-Views monitoring tools is depicted in Fig. 1. LISP-Views consists of several modules with different functions. The *Measurement*, *Report*, and *Raw Data* modules are deployed on a centralized server while, the *Crawler*, *Sonar*, and *Controller* modules can be deployed on several different VPs. As for now, LISP-Views is just deployed on one VP, where the monitor and centralized module are deployed together. All the modules are implemented in *Python* and described as follows :

Sonar module : the main module with two functions : 1. sending the Map-Requests and receiving the Map-Replies to/from all the existing MRs based on RFC 6830 [Fea13]; 2. storing the received information in *Report* and *Raw Data* with different purposes. The IP address that *Sonar* uses to query to MRs is selected either from the output of *Crawler*, or from the recorded information in the previously produced *Report*. The reasons are explained in the corresponding modules.

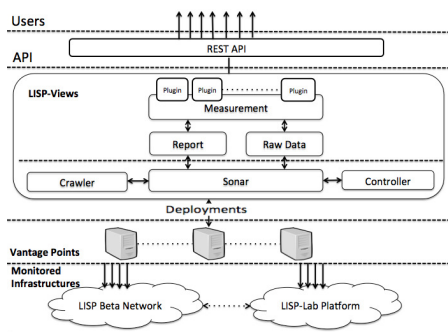


FIGURE 1: Monitoring Architecture.

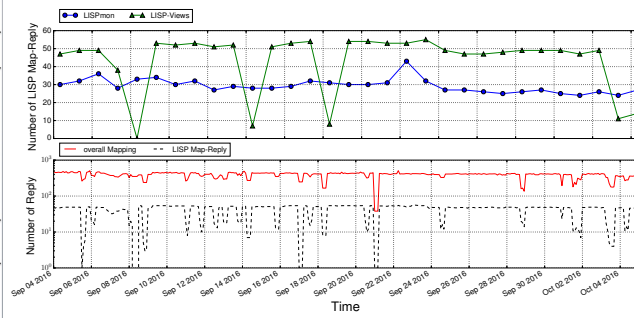


FIGURE 2: Comparison between LISPmon and LISP-Views.

Crawler module : scans all the existing IPv4 addressing space using *Sonar*. If the queried IP (e.g., 192.0.2.1) has no Map-Reply, *Crawler* increments the IP by one as the next queried IP (i.e., 192.0.2.2) and lets *Sonar* send the Map-Request. If *Sonar* obtains Negative or LISP Map-Reply, since the Reply contains a prefix (mentioned in Sec. 2.1, e.g., 192.0.2.0/25), *Crawler* sets the first IP beyond the returned prefix (e.g., 192.0.2.128). Scanning the whole IPv4 address space takes very long time, and is mainly caused by time wasted waiting for no Map-Reply. Indeed, no Map-Reply means to wait 3 seconds, and the next queried IP is just increased by one, instead of skipping a block of IP addresses like in the case of Negative/LISP Map-Reply.

Report module : contains the collected EID-prefixes in a list, as well as a list of MRs that answered. As crawling the address space may take long time but we aim to obtain the status of MRs as frequently as possible, *Sonar* sends Map-Requests for the EIDs recorded in *Report* only to previously responded MRs. Thus, it decreases the possibility to receive no Map-Reply, so to get LISP status within a shorter time.

Raw Data module : contains all the detailed information of Map-Replies for each MR. So, it can be used to perform thorough performance analysis of MDS. Moreover, since the *Raw Data* is stored according to the MR, it is possible to track the performance of the MRs individually.

Measurement module : provides the composition of requested measurements (i.e., select different measurement plug-ins) based on the analysis of *Raw Data* and *Report*.

REST API : is connected to *Measurement module* so that the users can launch a custom experiment by setting the experiment time, the monitored MRs, and obtain the different aspects of LISP status. We are currently implementing the REST API, so for this paper the execution of measurements was performed using command lines.

Controller module : synchronizes all the modules in LISP-Views and also specifies the start and stop time of producing both *Report* and *Raw Data*. Since the inconsistency between MRs exists, the interval of producing Report and Raw Data for every MR differs from each other. Thus, the interval set in Control module should cover the slowest MR.

4 LISP-Views Validation

4.1 Methodology

In order to validate and evaluate the LISP-Views monitoring tool, we used raw data and report collected during one month (from 0 :00 September 4th to midnight of October 4th 2016), by deploying LISP-Views on one VP, which is an xTR of LISP-Lab platform. The interval to produce reports and raw data was respectively 6 and 2 hours. Unfortunately, both MRs of LISP-Lab platform just responded to the Map-Requests at the first time and then stopped. They are not able to handle large number of queries, because the Map-Requests fills request queue and are dropped. Per-se this is a success, because this bug was unknown, and the OpenLISP coders fixed the issue after our measurements. All the results of evaluation used in this paper only depend on the 6 working MRs of LISP Beta Network. The aim of the work is to validate LISP-Views by assessing if it provides at least the same results as LISPmon.

4.2 LISP-Views vs. LISPmon

Fig. 2 shows the number of LISP Map-Replies received from LISPmon and LISP-Views over time. The upper figure makes a comparison between the 2 monitors. The line with point is obtained from the daily LISPmon publications, indicating the number of LISP Map-Replies returned by one MR in an uncertain time monitoring round every day. To facilitate the comparison, we pick up the results from LISP-Views at 8 :00 a.m., which is the nearest experimental round to LISPmon. As presented in the curve with triangle symbols, LISP-Views reflects the fact of a combination of the number of LISP Map-Reply containing the different EID-prefixes from all the MRs at a fixed experiment time. In most cases, LISP-Views receives around 20 LISP Map-Replies more than LISPmon, reflects the fact that the MRs are not coherent at any time due to the existence of convergence time for synchronizing the mapping information among them. The other 5 days, where LISP-Views receives very few LISP Map-Replies, show that there were issues to normally receive LISP status at that time, whereas LISPmon always presents a more smooth trend but hides the reality.

The bottom figure of Fig. 2 depicts the number of LISP Map-Reply (in dashed line) and the overall Mapping (LISP and Negative Map-Reply, in solid line) from LISP-Views for every 2 hours monitoring round during the whole experiment. Although the number of Map-Replies presented in the figure is still the combination of results from 6 MRs when the returned EID-prefixes are different, it shows that the MRs are not quite stable. Sometimes the MRs provide less than 10 (or even 0 LISP Map-Reply), but in the next experiment round, the number is recovered. Besides, the total number of successful queries every 2 hours also presents the behavior of instability, where it generally oscillates between 400 and 500, but it occasionally drops lower than 300. Both two lines have almost same changing trend, indicating that the change of overall Map-Replies is mainly affected by LISP Map-Reply. In some monitoring rounds, where the number of LISP Map-Reply approaches 0, but we can still receive the Negative Map-Replies, means that our proposed monitoring architecture works well so that this behaviour is caused by issues in the MRs, especially for answering the LISP Map-Reply.

5 Conclusion

Motivated by the only LISP monitor deployed so far, named LISPmon, which records the current LISP status only from a specific MR just once per day, we propose a more dynamic LISP monitoring architecture, namely LISP-Views, so to deepen the understandings on LISP and to ease day-to-day operations and troubleshooting. After one full month of comparison with LISPmon, we find that LISP-Views provides more information by discovering more mapping information from all MRs and more complete mapping information of each MR. LISP-Views is still in the first phase, so the implementation of a REST API, the deployment on multiple VPs and the test of IPv6 behavior are still ongoing and future work.

Acknowledgments : Research work presented in this paper benefited support from NewNet@Paris, Cisco's Chair "NETWORKS FOR THE FUTURE" at Telecom ParisTech (<http://newnet.telecom-paristech.fr>). Any opinions, findings or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of partners of the Chair.

Références

- [Fea13] D. Farinacci and et al. The locator/ID separation protocol (LISP). RFC 6830, Internet Engineering Task Force, January 2013.
- [lisa] LISP Beta Network. See <http://www.lisp4.net>.
- [lisb] LISP-Lab. See <http://www.lisp-lab.org/>.
- [lisc] LISPmon. See <http://www.lispmon.net>.
- [LSID16] Y. Li, D. Saucez, L. Iannone, and B. Donnet. Stability and consistency of the lisp pull routing architecture. In *Proc. Workshop on Network Measurement (WNM)*, November 2016.
- [SIBF12] Damien Saucez, Luigi Iannone, Olivier Bonaventure, and Dino Farinacci. Designing a deployable internet : The locator/identifier separation protocol. *IEEE Internet Computing*, 16(6) :14–21, 2012.