2017 29th International Teletraffic Congress

LISP-Views: Monitoring LISP at Large Scale

Yue Li*, Akram Abouseif†, Luigi Iannone*, Damien Saucez‡
∗ INFRES, Telecom ParisTech – France
† COMELEC, Telecom ParisTech – France
Email: first_name.last_name@telecom-paristech.fr
‡ Université Côte d'Azur, Inria – France
Email: damien.saucez@inria.fr

***Abstract*—The *Locator/Identifier Separation Protocol* (LISP) separates classical IP addresses into two categories: one for identifying terminals, the other for routing. To associate identifiers and locators LISP needs a specific mechanism, called mapping system. This technology is still at an early stage but two experimental platforms have already been deployed in the Internet: LISP Beta Network and LISP-Lab. However, only the LISP Beta Network is monitored with LISPmon that partially monitors the mapping system once a day. To accompany the growth of LISP, a dynamic and complete monitoring system is required. Therefore, we propose LISP-Views, a dynamic versatile large scale LISP monitoring architecture. LISP-Views allows to automatically conduct comprehensive and objective measurements. After running LISP-Views in the wild for several months and comparing the monitoring results with LISPmon, we confirm that LISP-Views provides more detailed and accurate information. We observe the different behaviours between every network entity within mapping system, and also explore the current LISP performance for further improvements.**

I. INTRODUCTION

Separating the locator and identifier roles of IP addresses was suggested to address the Internet scalability and mobility issues [1] and among the various proposed solutions, the *Locator/Identifier Separation Protocol* (LISP) [2] has been standardized at the Internet Engineering Task Force (IETF) for nearly ten years. Its main philosophy is to separate current IP addressing space into two sub-spaces: EID (Endpoint IDentifier) and RLOC (Routing LOCator). The former one is the identifier of end hosts, also used to locally route within the local domain; while the latter one refers to the interface of edge routers in the Internet topology, also used to route through the Internet core. In LISP, a new network entity, the so-called *Mapping Distribution System* (MDS), is introduced and uses a pull model to store and distribute the bindings between EIDs and RLOCs. In addition to directly addressing the tussle between location and identification in IP, LISP also brings the additional benefits of flexible inter-domain traffic engineering, Virtual Machines mobility in multi-site Data-Centers, facilitated transition to IPv6 and so on ([3][4][5]).

To promote the development of LISP and boost the related research, large scale flexible platforms are necessary. Two LISP-related platforms have been interconnected so far. The experimental LISP Beta Network testbed [6] is deployed since 2008, and the LISP-Lab platform [7] is open to external experimenters since 2015. Currently, a unique LISP monitoring system called LISPmon [8] supervises the global MDS and publishes the mapping information daily. However, it is known that the mapping information sometimes changes frequently within a day and that the elements constituting the MDS are not always consistent [9]. We hence propose a dynamic versatile LISP monitoring architecture, namely 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 accuracy, including comparing it with LISPmon. We show that LISP-Views is more accurate than LISPmon since it monitors all the MDS elements in parallel. In addition, thanks to its detailed reporting, LISP-Views allows to assess high level metrics of LISP deployments such as reliability, latency, or configuration issues.

In the remainder, Sec. II introduces the necessary LISP background, the current deployment status of LISP and the working mechanism of LISPmon. Sec. III describes our proposed LISP monitoring architecture in details. Sec. IV validates LISP-Views by comparing with LISPmon. Sec. V provides the snapshot of what kind of further analysis can be done with our proposal. Finally, Sec. VI concludes the paper.

978-0-9883-0453-6/17 $31.00 © 2017 ITC
DOI 10.1109/ITC.2017.38

II. LISP BACKGROUND

A. *LISP architecture overview*

The *Locator/Identifier Separation Protocol* (LISP) splits the conventional IP addressing spaces into two logical sub-spaces: (i) *Endpoint IDentifier (EID)*: is the traditional IP address, which identifies the terminal hosts. Leveraging on EID as the source/destination address of hosts, they are able to communicate with each other in a local scope, i.e., within LISP-Site (AS_s/d in Fig. 1). (ii) *Routing LOCator (RLOC)*: is also the conventional IP address, which indicates the attachment point in the Internet topology. RLOCs are used for packet transfer on the Internet core, i.e., between LISP-Sites (Internet in Fig. 1), by being used as the source/destination address to the edge routers. More specifically, when the host in the AS_s communicates with the host in the AS_d in Fig. 1, it needs to use EID_s as source address and EID_d as destination address. After the conventional IP packets routing to edge routers, called xTR (the combination of Ingress and Egress Tunnel Router), xTR_1 first checks in its mapping cache [10] to find out the association between EID_d and its RLOC. Otherwise, it sends a Map-Request to Map-Resolver (MR). If MR is authoritative, it directly returns back a Map-Reply containing the mapping information. If not, after forwarding the Map-Request within MDS and from Map-Server (MS) [11] to the destination side, xTR_1 receives a Map-Reply from one of xTRs of EID_d. Then, xTR_1 encapsulates the packets by adding RLOC_xTR3 as the destination address and RLOC_xTR1 as the source address in the outer header and send the encapsulated packets through Internet. Once xTR3 receives these packets, it dencapsulates and forwards them by using the original IP packets. If the destination site is non LISP-site, the MDS directly returns back the Map-Reply without mapping information to xTR_1, and the xTR_1 encapsulates the packets using the RLOC of PxTR (Proxy Ingress/Egress Tunnel Router) as the destination address and forwards the IP packets there. This mechanism will not be evaluated in this paper, more details about PxTR performance can be found in [12] and [13].

Based on LISP architecture, there are three types of Map-Reply that can be received:

(i) *LISP Map-Reply*, means that the queried IP address belongs to a LISP site (i.e., EID) and the Map-Reply contains the mapping information for this site, including: the association between the EID-prefix that queried EID belongs to, and the list of RLOCs of the destination site.

(ii) *Negative Map-Reply*, means that the prefix covering the queried IP address belongs to a non-LISP site (i.e., conventional site), and the Map-Reply contains no mapping information. We consider these two kinds of replies as a successful query.

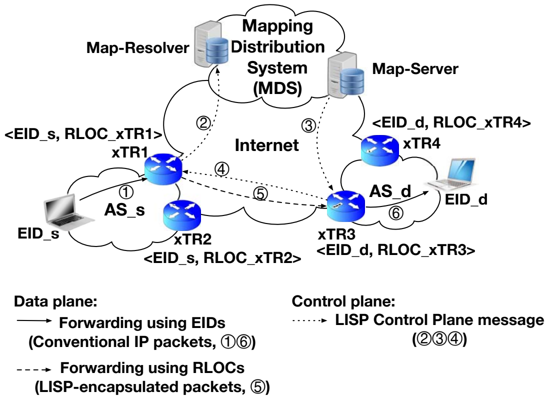

Fig. 1. LISP architecture overview

(iii) *No Map-Reply*, means that the xTR does not receive any reply during a certain time.[1] In this case, we consider the query as failed.

As described above, the locator/id separation paradigm is able to reduce the routing table size, since the local LISP-sites do not need to know the routing information of Internet core anymore, while the routers on Default Free Zone do not need maintain the routes to the stub-network, either. Further, opposed from the current BGP architecture, which uses a push-based model, LISP leverages on a pull model, which only updates the routing information when necessary, so to reduce the burden of overwhelming announcement traffic.

B. *Current LISP Deployment Status*

To boost the development of LISP and explore its real-world behavior, two experimental LISP testbeds have been deployed so far. One is the LISP Beta Network, which is initiated by Cisco as well as some companies nine years ago. The other one, opened to the external users from 2015, is the LISP-Lab platform, which is coordinated by a French consortium. It has already been interconnected to the LISP Beta Network with an OpenLISP DDT root [14]. According to the latest architecture, LISP Beta Network has 12 MRs (4 in Europe, 6 in US and 2 in Asia) and LISP-Lab platform has 2 additional MRs (both located in Europe), which allow to query the global Mapping Distributed System (MDS), consisting in a LISP-DDT deployment [14]. Besides, they also have all necessary LISP network entities, such as: xTRs, MSes and PxTRs.

---

[1] In LISP-Views the timeout is set to 3 seconds, according to the standard described in [2].

LISPmon is the only monitoring platform supervising the current LISP status, developed at the Advanced Broadband Communications Center of UPC. 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 one specific MR of the LISP Beta network (i.e., 12 MRs are candidates to be selected). If this MR does not function normally, it will choose another one as a replacement, and 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.

III. PROPOSED MONITORING ARCHITECTURE

A. Motivation

In order to move LISP forward, we need to deeply understand the behaviour of the different LISP network entities and since the MDS reflects the status of a LISP network as it stores all the mapping information, it is essential to be able to monitor the MDS. 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 may be unstable and inconsistent between MRs, i.e., the mapping information sometimes dramatically changes within a day, and the Map-Replies from the different MRs may not coincide at a given time [9]. It is similar to the world-wide distributed BGP looking glass servers, which do not always provide the same responses for an IP address as the whole routing system may not have converged or because of routing policies. 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. In fact, LISP-Views not only can be used to monitor LISP, but also can be used in the non-public networks, such as VxLAN [15].

LISP-Views is an open source implementation and has been designed to fulfill the following objectives:[2]

1) LISP-Views queries to all the working MRs existing in current MDS, in parallel, while LISPmon just queries one, aiming at building a complete view of the current LISP status.

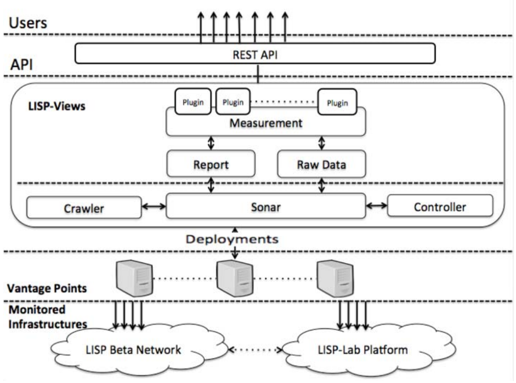

Fig. 2. Monitoring Architecture

2) LISP-Views periodically monitors all the MRs with arbitrary intervals,[3] while LISPmon just does it daily, aiming at providing information about the mapping evolution at smaller time granularity.
3) LISP-Views supervises the whole MDS without any manual process, automatically reacting to failing components (e.g., unresponsive MRs).
4) LISP-Views obtains the mapping information from all the MRs of the LISP Beta Network as well as LISP-Lab platform, whereas LISPmon prefers to leverage the MR of the former one.
5) LISP-Views is flexible and configurable, with users able to define different monitoring jobs and get the various measurements, whereas LISPmon just publishes the mapping list daily.
6) By design LISP-Views can be extended to be used as a monitoring facility for the internal deployed VxLAN.[4]
7) LISP-Views is able to be deployed on multiple VPs over the world, while LISPmon publishes the results depending on one VP.

B. Description of LISP-Views

The architecture of the proposed LISP-Views monitoring tools is depicted in Fig. 2. 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

---

[2] Source code available on Github: https://github.com/SeleneLI/LISP-Views

[3] This interval can be changed by the experimenters to accord their necessity. But in this paper, the interval is set the shortest according to the process capacity of our server.

[4] In this paper we focus on LISP.

VP in Paris, France, where the monitor is developed and the centralized module is deployed. Nevertheless, we are planing to implement the distributed version of LISP-Views. For this reason, we will not discuss further how to deploy LISP-Views on multiple VPs. All the modules are implemented in *Python* and described as follows:

Sonar module: the main module with two functions: 1) sending the LISP encapsulated Map-Requests and receiving the Map-Replies to/from all the existing MRs based on the standard [2]; 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.

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. II-A), *Crawler* sets the first IP beyond the returned prefix. For instance if the returned prefix is $192.0.2.0/25$, the next queried IP is 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 MRs that have previously responded. Thus, it decreases the possibility to receive no Map-Reply, so that to get LISP status within a shorter time.

Raw Data module: contains all the detailed information of Map-Replies for each MR, such as Map-Reply type, RLOCs, required EID-Prefix, Round Trip Time (RTT) and the returned source for each round specify with $TimeStamp$ (regardless the source of *Sonar*). 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*. The interval of generating *Report* and *Raw Data* can be changed according to the hardware processing capability. However, since the inconsistency between MRs exits, 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.

IV. LISP-VIEWS VALIDATION

A. 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 4^{th} to midnight of October 4^{th} 2016), by deploying LISP-Views on one VP, which is an xTR of LISP-Lab platform.

The interval to produce reports was 6 hours, and the interval to produce raw data was 2 hours. Unfortunately, both MRs of LISP-Lab platform just responded to the Map-Requests at the first time and then stopped. They were not able to handle large number of queries, because the Map-Requests fill the request queues, which are not deployed in manner to sufficiently follow up the queries, resulting a drop in the MRs. Per-se this is a success, because this bug was unknown, and the OpenLISP coders are working on a fix. All the results of evaluation used in this section only depend on the 6 working MRs of LISP Beta Network (3 in Europe, 1 in US and 2 in Asia) and the other 6 MRs were down at the moment of conducting the experiment. The aim of the work is to validate LISP-Views by assessing if it provides at least the same results as LISPmon.

B. LISP-Views vs. LISPmon

In this section, we compare LISPmon and LISP-Views, so to asses if subset of information comparable between the two monitoring plateforms are identical, hence validating LISP-Views. As we indicated in Sec. III-A, at any time LISPmon just queries one MR, on LISP Beta Network once per day and generally begins at 7:00 a.m.. The queried MR is changed sometimes when it has issues. LISP-Views, however, keeps sending Map-Requests to all the 6 MRs with an interval of 2 hours everyday, so to retrieve the actual real status of each MR.

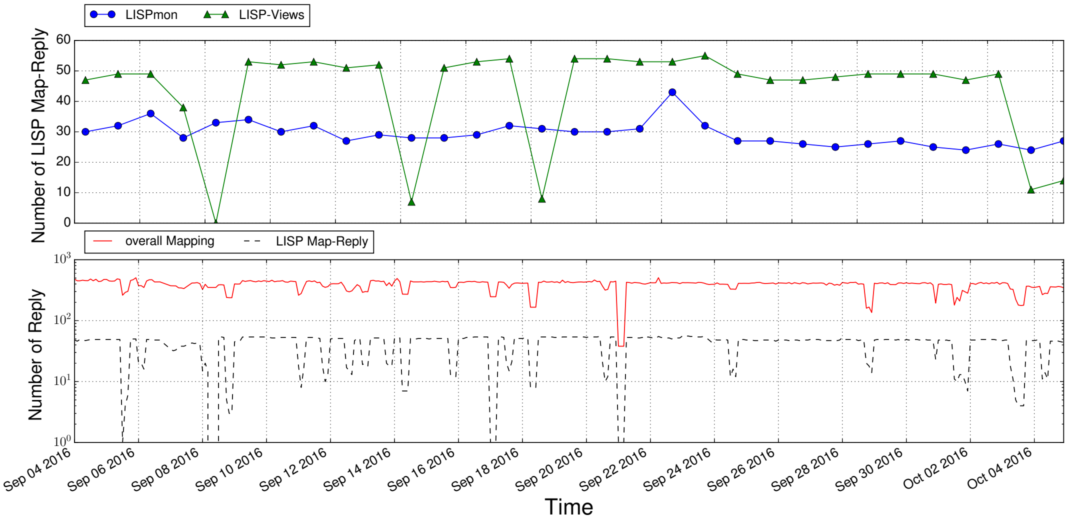

Fig. 3. Comparison between LISPmon and LISP-Views. The upper sub-figure shows the number of LISP Map-Reply from LISPmon generally at 7:00 and LISP-Views exactly at 8:00 over days. The bottom sub-figure indicates the LISP Map-Reply and overall Mapping from LISP-Views over time with an interval of 2 hours.

Fig. 3 shows the number of LISP Map-Replies received from LISPmon and LISP-Views over time. The upper sub-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 smoother trend but hides the reality.

The lack of stability can be better appreciated in the bottom sub-figure of Fig. 3. Such figure 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 even provide 0 or less than 10 LISP Map-Replies, but in the next experiment round, the number is recovered. Besides, we evaluate the total number of successful queries every 2 hours. It also presents the behavior of instability, where it generally oscillates between 400 and 500, but it occasionally drops lower than 300. Both solid and dashed lines have almost same changing trend, indicating that the change of overall Map-Replies is mainly affected by LISP Map-Replies. Monitoring rounds where the number of LISP Map-Replies approaches 0 but where Negative Map-Replies are still received indicate issues at the MRs, especially for answering the LISP Map-Reply.

We compared the number of received LISP Map-Replies within a whole day between LISPmon and LISP-Views over 30 days. Shown in Fig. 4, as LISPmon only publishes one record each day, the number (blue points) is exactly identical to those in Fig. 3. LISP-Views, however, provides a combination of LISP Map-Replies not only from all MRs but also from 12 monitoring rounds within one day. We observe that our monitor architecture receives more LISP Map-Replies than LISPmon in all days with a large difference. 83.3% of the time LISP-Views receives more than 20 LISP Map-Replies. The maximum number of LISP Map-Replies that our proposed monitor obtains is 55, while the maximum value for LISPmon is 43. Differently from the upper sub-

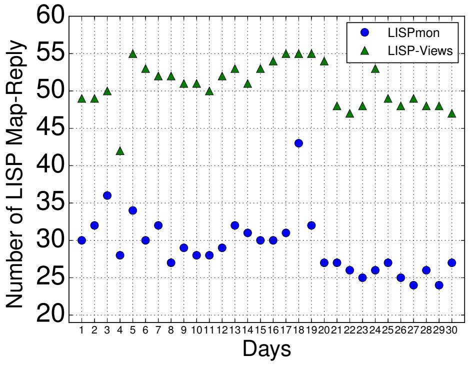

Fig. 4. Comparison between the LISP Map-Reply of LISPmon and LISP-Views over days

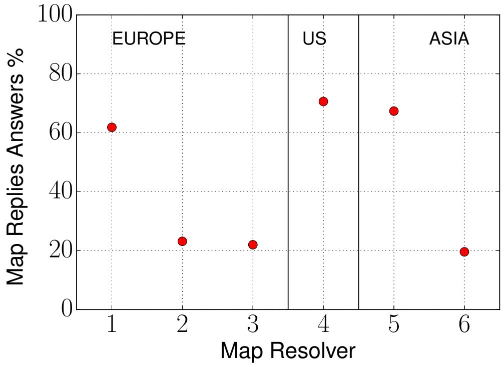

Fig. 5. Reliability of each MR

figure of Fig. 3, where LISP-Views occasionally receives nearly 0 LISP Map-Replies, the total number within a whole day is always more than 40, which illustrates that the MRs may not work normally in some rounds but they are able to recover fast.

Since LISP-Views repeats querying simultaneously to every MRs, it is able to report on the status of each MR at anytime. It provides more complete mapping information compared to LISPmon and is also able to highlight sporadic issues with MRs.

V. DISSECTING LISP WITH LISP-VIEWS

After validating LISP-Views in the previous section, this section presents several examples of how LISP-Views can be used to dissect LISP and obtain in-depth results. The experimental dataset used is the one in Sec. IV-A.

Fig. 5 shows the *Reliability* of each MR during our data collection, by calculating the percentage of successful queries over the total number of Map-Requests. MR1, MR4, and MR5 give the highest reliability values, which are more than 60%. The lowest reliability values, about 20% are observed for MR2, MR3, and MR6. In general, the reliability of each MR is different and low compared to the years of 2012 and 2013 presented in [16], which coincides with the fact that there was a change of the MRs architecture on LISP Beta Network that year, and an updated-software was tested on MRs as well. The low value of reliability is caused by MRs having an unstable behaviour, as shown in Fig. 3, the number of Map-Reply changes heavily over time, and especially the number of LISP Map-Reply sometimes drops to 0.

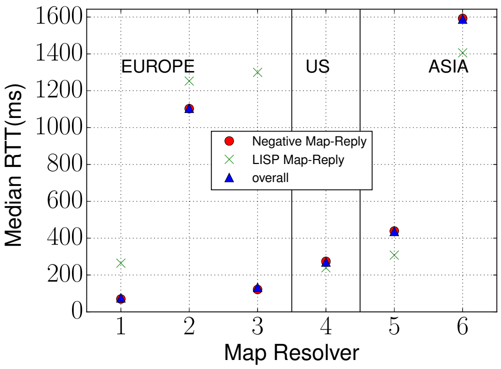

Fig. 6. Median RTT per MR (In the most time, the Negative Map-Reply and overall are overlapped.)

In order to understand the behavior of each MR in terms of latency, we analyze the median RTT obtained from our dataset. The RTT here refers to the Round Trip Time from sending out the Map-Request until receiving the Map-Reply. Fig. 6 shows that the best performance come from MR1, MR3, MR4, and MR5; since the overall RTTs are much lower than the others. For MR3, the number of LISP Map-Replies is much higher than the number of Negative Map-Replies, probably because its embedded Map-Server registers less EID-prefixes hence requiring Map-Requests to be forwarded to a remote MS. Then, the Map-Request is forwarded to the xTR, where the remote MS registers and the xTR gives back the Map-Reply. Compared to the Negative Map-Replies, which are normally returned by MR, LISP Map-Replies take longer time, especially in this case. However, MR2 and MR6 present very high latency, particularly for MR2 (located in Europe), but its RTT is even higher than MR5, which is located in Asia. It is mainly caused by a very high CPU usage of these two MRs preventing them sometimes to reply. If we focus on the RTT of LISP Map-Replies, only MR1, MR4, and MR5 provide the best behavior, which coincides with the result of Fig. 5 on reliability. The high RTTs of LISP Map-Replies are from the other 3 MRs, where the median RTT is around

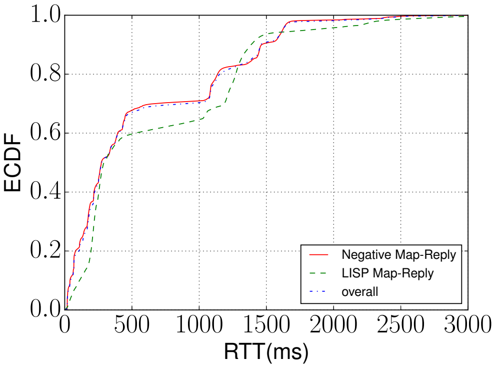

Fig. 7. ECDF RTT comparison between LISP and Negative Map-Reply

1300 ms, is caused by the failed queries. Furthermore, it also explains why we can always receive the Negative Map-Replies, but the number of LISP Map-Replies is sometimes quite low in Fig. 3. Half of LISP Map-Replies are more than 1300 ms and partly even more than 3 s. Since our measurement timeout is set to 3 s it implies that some Map-Replies are dropped by our measurement unit. As the RTT of Negative Map-Replies is generally lower, we can receive more of them. This phenomenon may be very biased by the VP location. Thus, deploying LISP-Views on multiple VPs to get more reliability information is a paramount future work. In addition, Fig. 6 presents that the number of Map-Replies returning from the MRs located in Europe and US is indeed higher than the number of Negative Map-Replies, which is expected. But both Asian MRs return LISP Map-Replies faster than Negative Map-Replies. The reason of this observation is unclear and requires further exploration.

TABLE I
PERCENTAGE OF MAPPING SOURCE

Map-Reply Type	Map Resolver	xTR	Other
Negative Map-Reply	98.79%	-	1.21%
LISP Map-Reply	0.14%	88.37%	11.49%

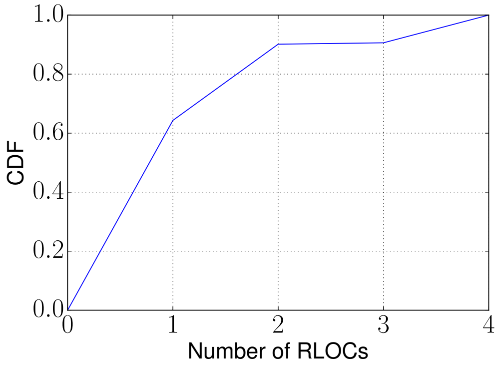

Fig. 8. CDF of the number of RLOCs

We also calculate the Empirical Cumulative Distribution Function (ECDF) of the RTT for the 6 MRs. Fig. 7 provides the ECDF of the RTT for Negative, LISP, and overall Map-Replies. As indicated in Sec. II-A, the LISP MDS needs more time to solve a complete mapping (LISP Map-Reply) than the Negative Map-Reply. We observe that the Negative Map-Reply is indeed faster than the LISP Map-Reply until the RTT reaches 1000 ms, then the behavior changes, i.e., the LISP Map-Reply sometimes becomes faster. What's more, we find that 67.03% of RTTs within 500 ms, 70.33% less than 1000 ms and 90.88% don't exceed 1500 ms for overall Map-Replies. In details, for the Negative Map-Reply, we found that 67.8% of RTT values within 500 ms, for the LISP Map-Reply that 59.7% of RTT values less than 500 ms. This figure almost presents a bi-modal distribution, where 500 ms is a peak and 1300 ms is another one. These two high occurrences of RTTs are exactly the most two frequent latency in Fig. 6.

The dataset obtained with our monitoring architecture also shows the information about *Mapping Source*. We explore the source answering the Map-Replies according to different types (LISP or Negative). In the case of LISP Map-Replies, we expect the source of replies to come either from one of the ETRs or the queried MR. On the contrary, for Negative Map-Replies, replies should just come from the queried MR. Tab. I presents the percentage of observations for the two types of Map-Replies. For the Negative Map-Reply, 98.79% come from the queried MR and 1.21% come from the other sources. Further, the other sources are actually the other MRs without query and it happens just for a fixed EID-Prefixes, i.e., if we send a Map-Request for one of these EID-Prefixes to a dedicated MR, the Map-Reply always comes from a fixed specific MR. As a conclusion, all the Negative Map-Replies are answered by MRs. For the LISP Map-Reply, we observe that 0.14% come from the queried MR, 88.37% come from one of their ETRs, but 11.49% come from the other sources in different locations. This unexpected behaviour needs further investigation.

The following type of measurement is the distribution of the size of RLOC set, i.e., how many RLOCs are associated to one EID prefix on average. As previously observed, the percentage of mappings using two or less RLOCs has increased between 2010 and 2012 [5]. Fig. 8 shows that this trend keeps going on, i.e., more mappings

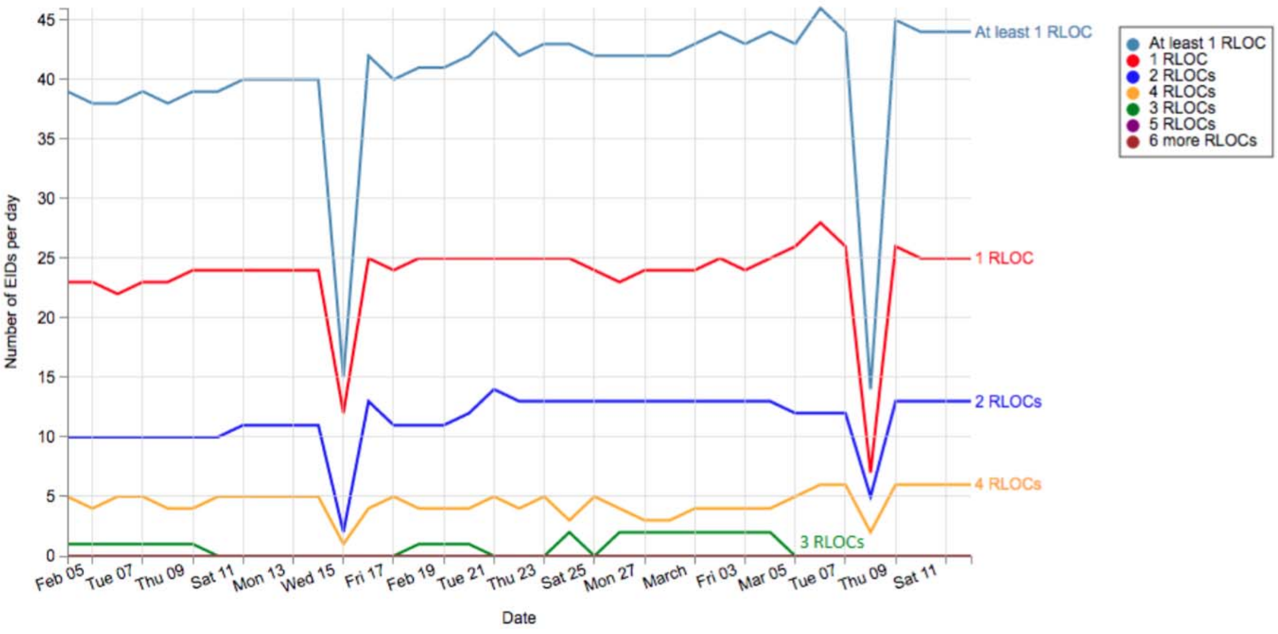

Fig. 9. EIDs by Quantity of Associated RLOCs [17]

use fewer RLOCs and the maximum number of RLOC is 4. An interesting point is that although LISP is a good candidate to support the increasing multi-homing in Internet, more than 60% LISP users are not multi-homed and among them the majority only has 2 RLOCs. Not only the number of RLOCs of each site does significantly change, but also the RLOCs themselves remains stable. We measured that the stability reaches 99.8% for all the dataset, i.e., once an EID-prefix – to – RLOCs mapping is decided, it rarely changes. Surprisingly, we have not found any mobile LISP sites.

All the aforementioned observations are based on the experiment made between September and October 2016 as described in Sec. IV-A. Later LISP-Views detects that MR2 and MR6 in Fig. 6 with very high overall RTTs are down, and three new other MRs are up. After confirming with the operators of the LISP Beta Network, these two MRs are indeed definitely down. The change of the architecture of MRs also proves the accuracy of LISP-Views, while the LISPmon presents a smooth change in its daily reports, hence hiding these facts. Since February 2017 LISP-Views publicly publishes preliminary daily reporting online [17]. Development efforts are still ongoing to provide a complete and production-level website. Fig. 9 is a capture from the website, it is one type of measurement about the number of EID-prefix quantified by the different size of RLOC set. The shown result is an union of all MRs during a whole day, to present the most complete mapping information of the actual LISP deployment. The line on the top indicates the number of LISP Map-Replies, i.e., the Map-Replies with at least 1 RLOC, which coincides to the results shown in Fig. 3 and is mainly affected by the EID-prefix with 1 RLOC. Moreover, the composition of the different sizes of RLOC set is also rather identical to the one in Fig. 8. The only difference is that in the previous dataset there is no EID-prefix with 3 RLOCs, while in the latest dataset it appears, but not very stable. In addition, the line with 3 RLOCs is almost complementary to the line with 4 RLOCs. It is probably caused by one LISP-site that has 4 xTRs but one being always down, or the LISP-site having 4 interfaces on a same xTR among which one is down. Besides, the number of observed EIDs per day heavily drops two times in the figure. Since the two valley don't drop to zero, we know that the problem doesn't come directly from the MDS. Instead, the problem comes from issues that occurred within the network of the LISP-Views server itself. This observation highlights the urgent need of distributing VPs.

VI. CONCLUSION

Very little is known about the behavior of LISP in operational environments and it still lacks of troubleshooting tools. 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. As LISP-Views aims at being deployed in large scale and dynamic networks, we make a comparison with LISPmon to validate the former one by comparing their behavior during one full month.

It demonstrates that LISP-Views provides more information by discovering more mapping information from all MRs and more complete mapping information of each MR. Furthermore, with our proposed monitoring platform, more mapping system performance metrics such as reliability, latency, and configuration issues can be assessed, which helps for further LISP improvements. 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.

ACKNOWLEDGMENT

Research work presented in this paper is supported by the French Research Agency under ANR-13-INFR-0009 LISP-Lab Project (www.lisp-lab.org) and 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.

REFERENCES

[1] T. Li, "Recommendation for a routing architecture," RFC 6115, Internet Research Task Force, February 2011.

[2] D. Farinacci and et al., "The locator/ID separation protocol (LISP)," RFC 6830, Internet Engineering Task Force, January 2013.

[3] D. Saucez, L. Iannone, O. Bonaventure, and D. Farinacci, "Designing a deployable internet: The locator/identifier separation protocol," *IEEE Internet Computing*, vol. 16, no. 6, pp. 14–21, 2012.

[4] D. Saucez, L. Iannone, A. Cabellos, and F. Coras, "Locator/id separation protocol (lisp) impact," tech. rep., 2016.

[5] D. Saucez and et al., "A first measurement look at the deployment and evolution of the locator/id separation protocol," *ACM SIGCOMM Computer Communication Review*, vol. 43, pp. 37–43, April 2013.

[6] "LISP Beta Network." See http://www.lisp4.net.

[7] "LISP-Lab." See http://www.lisp-lab.org/.

[8] "LISPmon." See http://www.lispmon.net.

[9] 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.

[10] L. Iannone and O. Bonaventure, "On the cost of caching Locator/ID mappings," in *Proc. ACM SIGCOMM CoNEXT*, December 2007.

[11] V. Fuller and et al., "Locator/ID separation protocol (LISP) map-server interface," RFC 6833, Internet Engineering Task Force, January 2013.

[12] Y. Li and L. Iannone, "Using ripe atlas to evaluate the locator/id separation protocol," in *Computer Communications Workshops (INFOCOM WKSHPS), 2016 IEEE Conference on INFOCOM 2016 Conference*, pp. 696–697, IEEE, 2016.

[13] Y. Li and et al., "Performance evaluation of locator/identifier separation protocol through ripe atlas," in *Proceedings of the 2016 conference on ACM SIGCOMM 2016 Conference*, pp. 561–562, ACM, 2016.

[14] V. Fuller, "LISP-DDT: A new mapping database for LISP," June 2012. Nanog 55 Talk.

[15] M. Mahalingam, D. Dutt, K. Duda, P. Agarwal, L. Kreeger, T. Sridhar, M. Bursell, and C. Wright, "Virtual extensible local area network (vxlan): A framework for overlaying virtualized layer 2 networks over layer 3 networks," tech. rep., 2014.

[16] F. Coras, D. Saucez, L. Iannone, and B. Donnet, "On the performance of the LISP beta network," in *Proc. IFIP Networking*, June 2014.

[17] "LISP-Views." See http://hunter.lisp-views.org/.