LISP-EC: Enhancing LISP with Egress Control

Ho Dac Duy Nguyen, Stefano Secci
Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6. Email: {ho-dac-duy.nguyen, stefano.secci}@upmc.fr

***Abstract*—The Locator/Identifier Separation Protocol (LISP) was specified a few years ago by the Internet Engineering Task Force (IETF) to enhance the Internet architecture with novel inbound control capabilities. Such capabilities are particularly needed for multihomed networks that dispose of multiple public IP routing locators for their IP networks, and that are willing to exploit them in a better way than what possible with the legacy Border Gateway Protocol (BGP) [8]. In this work, we specify how to enhance the LISP routing system to perform egress control too. Our goal is to give the highest possible routing optimization degree to LISP networks, so that ingress and egress traffic engineering strategies can be jointly performed, without requiring coordination between LISP and BGP. We design the enhancement to the LISP router system, specify the required protocol extensions, open sourcing the code and proving the overhead and the achievable gains by experimentation.**

I. INTRODUCTION

In this work, we focus on giving a more efficient inter-domain traffic engineering scope of operation to multihomed stub Internet networks, which account for the majority of Internet Autonomous Systems (ASes). In fact, roughly 84% of the them are stub ASes, and most of them are multihomed [2].

The growing number of multihomed networks challenges the scalability of the whole Internet routing system. When a multihomed AS announces its network prefixes to several providers, a common practice consists in de-aggregating the parent prefixes to perform fine-grained inbound traffic engineering, so that, with n transit providers, a multihomed AS typically announces about n different sub-prefixes for each single prefix, hence contributing to Internet routing table bloat [4]. As shown in [3], over a period of 4 years multihomed ASes created approximately 20-30% more prefixes to the BGP routing table than single-homed ones. In order to preserve Internet scalability, a wide range of alternative solutions have been proposed, most of them relying on IPv6 addressing and/or following the concept of separating the locator and the identifier roles of an IP address.

Besides the primary goal of a highly available Internet interconnection, multihomed ASes also target to improve their network performance by employing intelligent route control. With BGP, egress traffic engineering (e.g., to which transit provider to send which traffic) can be performed by means of local preferences in the routing decision process [3], while ingress traffic engineering (e.g., through which transit provider which incoming traffic comes from) is strongly limited by the absence of adequate control-plane functions: despite some tricks are possible, there is not direct control on incoming traffic routing. Also to enhance this aspect of Internet routing, the Local/Identifier Separation Protocol (LISP) [5] was proposed; indeed, LISP allows to associate to each prefix (announced via BGP) a preferred routing locator, among many possible ones, by means of a mapping system (independent of BGP). In this way, a multihomed network can perform egress traffic engineering using BGP, while using LISP for ingress traffic engineering. However, to dispose of both ingress and egress traffic engineering for a given multihomed network, the two protocols are supposed to inter-work, which is not specified in the standard. More precisely, it is not explicitly specified how LISP and BGP should run in a same node, or how physically separated LISP and BGP routers should be interconnected, etc. This is also complicated by the fact that a multihomed LISP network may not be running BGP, as it happens with edge networks with provider dependent addressing, or targeting a specific LISP deployment use-case that poses no strict external addressing requirements as BGP deployment does. Indeed, as it is evidenced in the new LISP Working Group charter, LISP has many applications (e.g., data-center networking, mobile user mobility management) that do not encompass BGP routing, hence having egress control in LISP without requiring integration with BGP is appealing.

In order to cope with these operational limitations, we specify in this paper how the LISP routing system can be enhanced in order to integrate egress control functions, besides the standard ingress control ones, in a way that does not impact the LISP architecture, nor any other protocols, and that can stay purely local to a LISP site. The remainder of this paper is organized as follows. In Sect. II, we position our work with respect to various inter-domain traffic engineering solutions for multihomed networks. Sect. III describes LISP-Egress Control (LISP-EC), our proposed extension of LISP routers to enhance control over the outbound traffic. The implementation of LISP-EC is documented in Sect. III-B. In Sect. IV, the performance evaluation results are presented and discussed. Further extensions on standard LISP protocol to support collaborative traffic engineering policies are specified in Sect. V. Finally, we conclude the paper in Sect. VI.

II. BACKGROUND

Internet protocols offering inter-domain multihoming traffic engineering (TE) capabilities can be classified into two major categories: host-centric and network-centric solutions. In the former one, the capability to decide source gateway for outgoing packets relies on local TE or scheduling policies at individual hosts, as it can be done with Site Multihoming by IPv6 Intermediation (SHIM6) [6], Host Identity Protocol (HIP) [11], Multipath TCP (MPTCP) [11]. With these host-centric approaches, the selection of outgoing interface can be

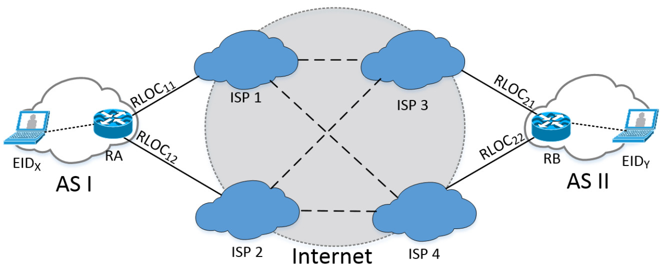

Fig. 1: LISP communication scenario example

a purely local decision, or the result of a negotiation between end-points, possibly passing via a server as proposed in [7]. When it is not a purely local decision, intensive signaling involves the end-hosts and/or the server. Another drawback of host-based solutions is that, in order to influence the egress network exit point when multiple ones are present, forms of source-specific routing, e.g. [13], are needed to follow ingress filtering policies implemented at upstream providers [10].

In network-centric solutions, traffic engineering mechanisms are defined and operated at the border router level and are made transparently from the end systems; the aforementioned host-centric constraints therefore no longer exist. Among the few network-centric locator-identifier separator protocols, LISP is the one that has been standardized since a decade, undergoing industrial adoption for network multihoming. In the following, we synthetically present the LISP architecture and its ability to support network multihoming traffic engineering.

A. LISP in a nutshell

Differently from the legacy flat routing structure, LISP involves two independent addressing spaces: for the Routing Locator (RLOCs) and the Endpoint Identifiers (EIDs), the latter being mapped to RLOCs by a mapping system. RLOC addresses are attached to LISP router interfaces, i.e. border routers that connect a LISP site to the Internet. While the RLOC addresses are globally routable, the EID addresses can stay routable only within the local LISP site.

Fig. 1 depicts a basic LISP communication scenario, where traffic is sent from host EID_X in AS I, to host EID_Y in AS II. RA and RB are the border routers of AS I and AS II, respectively. $RLOC_{11}$, $RLOC_{12}$ are the network interfaces connecting RA with two upstream providers, ISP 1 and ISP 2, respectively. In the LISP jargon, RA is a tunneling router (xTR), $RLOC_{11}$ and $RLOC_{12}$ are its two routing locators. Similarly, $RLOC_{21}$ and $RLOC_{22}$ for RB and AS II. Traffic from EID_X to EID_Y first reaches RA, which looks up its mapping cache to find the destination RLOCs. Assuming that the mapping for EID_Y is already installed in the RA mapping cache, and $RLOC_{21}$ is the preferred locator, an IP-UDP tunnel is established, encapsulating all the packets originated from EID_X with an outer IP header with $RLOC_{21}$ as destination address. The source address is selected from the routing table as the best outgoing interface toward $RLOC_{21}$ from RA, i.e., $RLOC_{11}$ or $RLOC_{12}$: this decision is taken by the underlay IP routing protocol, e.g. BGP or an internal gateway protocol or the default IP configuration. Hence traffic from EID_X is forwarded to $RLOC_{21}$, then RB decapsulates the received packets and forwards them internally according to destination address specified in the inner IP header.

B. Inter-domain traffic engineering with LISP

Traffic engineering support in LISP relies on two metrics that are assigned to RLOCs and distributed by the mapping system: the priority and weight metrics. When multiple RLOCs exist for a LISP EID prefix, the best priority one is preferred (i.e., least priority cost metric value), and in case of equal priority, traffic is distributed among them in proportion to their weight metric.

The usage of the RLOC metrics is often referred to as LISP-TE in the literature. By regulating the EID-to-RLOC mappings that a LISP site (and its xTRs) registers with the LISP mapping system, then distributing to other LISP sites transmitting traffic to it, the LISP site can control how traffic enters in its network from the LISP-capable Internet. Considering the scenario in Fig. 1, AS I may have local preference on its default inbound ISP, e.g. ISP1 because less expensive or with better performance. To express that policy, RA can register its mapping entry so that both $RLOC_{11}$ and $RLOC_{12}$ are announced as the routing locators for EID_X, but with a priority cost metric value for $RLOC_{11}$ set to a lower value than the one assigned for $RLOC_{12}$, the backup locator. This mapping is distributed by the mapping system (in a pull mode), and then employed by all other LISP sites that send traffic to EID_X, therefore via $RLOC_{11}$ and ISP1.

Despite LISP provides inbound TE capabilities, it does not offer outbound control features, i.e., which source RLOC to use when sending traffic to a destination RLOC. From a TE perspective, this can be seen as a limitation, and could also lead to override destination network RLOC preferences if opposed to local ones as argued in [2]. The main reason for not having included egress control capabilities in LISP is that the reference use-case that drove its standardization effort saw a deployment of LISP in conjunction with BGP, with as main goal the reduction of the BGP routing table size and the addition of inbound traffic capabilities to Internet routing. In such a case, with LISP and BGP deployed in the same xTR node, egress control can be left to the BGP decision process that includes local preference and multi-exit discriminator metrics for outbound traffic engineering. However, most of LISP deployments today do not involve BGP; they rather concern intra-AS traffic engineering use-cases, overlay management in data-center networking or access network mobility management. These use-cases have instead been considered recently in the LISP working group and are explicitly mentioned in the new LISP charter.

III. LISP EGRESS CONTROL

To cope with the lack of egress control in LISP, we propose LISP Egress Control (LISP-EC), an enhancement of the LISP routing behavior that gives an xTR the control on source RLOC selection, independently of the underlying IP routing decision or static configuration. LISP-EC is based on an alternative EID-RLOC mapping structure that allows associating destination RLOCs with multiple source RLOCs. The novel design permits xTRs to determine both head and end points of a tunnel without consulting the underlying routing protocol. In Fig. 2 we depict the proposed LISP-EC mapping entry structure. Fundamentally, it is an extension of the legacy mapping structure, in which each destination RLOC is associated with an extra list of source RLOCs. These attached RLOCs maintain the same properties as the destination RLOC, but hold a different meaning. While the destination RLOC is the routing locator for the remote EID-prefix(es), the source RLOC is the routing locator for the prefixes originated from local network. Within a LISP site, these local or source locators can be seen as the gateways for end hosts. Thanks to the LISP-EC mapping design for remote EIDs, an xTR is now capable of relating source RLOC choice with the selection of destination RLOC, and vice-versa.

It is worth mentioning that LISP-EC mapping design is not a traffic engineering mechanism per se, it is rather an extended behavior of LISP routers giving them the novel traffic engineering capability to distribute traffic among the gateways, which could be used by an external control-plane. LISP-EC mapping design introduces a new dimension for jointly controlling inbound and outbound traffic.

A. From RLOC selection to LISP-EC traffic engineering

Traffic engineering in standard LISP is limited to the capability, for the destination network, to announce its preferences over its RLOCs through the LISP mapping system; the source network is supposed to follow the destination network preferences. However, there may be a strategic clash in case the destination network preferences are for some reasons opposed to source network preferences. In such a case, the source network can bypass destination network preferences, knowing that if it sends traffic to an RLOC that is currently not the preferred one by the destination, such traffic will not be dropped (this is the case of all the public LISP implementations as of today).

With LISP-EC, we allow the source network taking into consideration its upstream preferences in a way that (i) it still permits to take into account destination network preferences, and (ii) increases the path diversity available between two edge networks. Indeed, while with BGP the number of available paths is equal to the number of external BGP peers, and with standard LISP it is equal to the number of destination network RLOCs, with LISP-EC it is equal to the product between the number of source RLOCs (possibly equal to the number of external BGP peers) and the number of destination RLOCs.

The processes configuring the egress priorities and weights at the source LISP network and the ingress priorities and weights at the destination LISP network can be two independent processes – as considered in [2], supposing the two edge networks are independent autonomously managed networks – or can be the result of a bilateral routing decision of Internet routing optimizers (commercial solutions exist, e.g.[14]) – which makes sense when the border routers of the two edge networks are operated by a same administrative entity.

Therefore, with LISP-EC, a new dimension of outbound traffic engineering mechanism is defined: it is no longer restricted to the determination of gateway or destination locator solely, it is now the control of load distribution over all possible RLOC-to-RLOC paths. By evaluating all combinations of gateway and destination locator, the best RLOC-to-RLOC path can be decided. Thanks to the LISP-EC extended mapping design, such a decision can be expressed and operated by means of RLOC priorities and weights.

Different traffic engineering policies can emerge in a LISP-EC communication context. In the following we list some we could identify - from one requiring no coordination whatsoever between LISP sites, to one requiring full TE control of both sites, passing through light coordination ones.

- *best source locator*: this policy consists in determining the best source RLOC based on local policies, whereas selecting the destination RLOC preferred by the destination. The decision on the best source RLOC can be taken following local egress TE preferences, for instance based on interconnection costs or performance (e.g., delay).
- *best forward path*: this policy consists in selecting the best RLOC-to-RLOC forward path, among all paths from the source xTR to the destination network, based on local policies, hence overriding the standard LISP behavior for which destination RLOC is chosen following destination preferences (RLOC priorities and weights) distributed by the mapping system. The decision on the best forward path is therefore entirely based on local policies, with a local preference on the destination RLOC that can be opposed to the inbound traffic engineering preference of the destination, because of forward path performance (e.g., delay, reliability) or whatever policy reasons.
- *equilibrium path*: under the hypothesis that two LISP networks communicate with equivalent traffic volumes over the two directions, this policy consists in selecting the RLOC-to-RLOC path corresponding to a routing equilibrium solution that strategically takes into consideration the preferences of both parties on both inbound and outbound routes. As a possible approach to compute the egress control metrics we refer to the routing interaction between LISP networks that was modeled previously in [2] as a non-cooperative game; a polynomial-complexity equilibrium computation framework was proposed and evaluated by a simulator assuming LISP egress control capabilities were available at xTRs.
- *global optimum path*: this policy considers, as the previous one, that two LISP networks exchange traffic with each other, but it differs from the equilibrium one in that the source RLOC and the destination RLOC are chosen accordingly to the global optimum path (i.e., what in the non-cooperative game modeling would correspond to the social welfare profile), which could differ from the equilibrium one, and which could override the unilateral

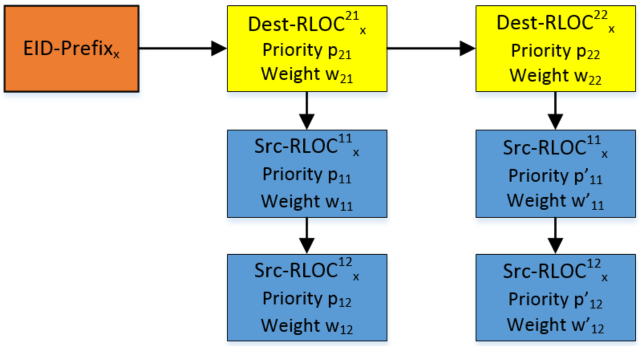

Fig. 2: Extended mapping entry structure

preferences of both networks.

Besides the selection of one or multiple destination RLOC(s), the outcome of a LISP-EC TE policy is the configuration of source RLOC priority and weight in a novel LISP mapping entry processing system as proposed hereafter.

B. Implementation Requirements

We address the LISP-EC implementation requirements based on the LIP6-LISP OpenLISP node system architecture [1]. Such a system has four components: the mapping database, the control-plane, the data-plane and the mapping socket. The control-plane runs in the user space and holds the responsibility for constructing and distributing mapping entries. Packet encapsulation as well as decapsulation runs in the kernel space relying on the mapping data. The mapping socket handles all the communications between user and kernel spaces, and it helps to populate the mapping databases. LISP-EC requires extending all four components.

For the sake of incremental deployability, LISP-EC should inherit the mapping structure from the legacy implementation, relying on the same mapping server and resolver interfaces, and should have no binding impact on control-plane messages. LISP-EC deployment should require upgrades at the xTRs only. The additional xTR operations needed are:

- to encapsulate outgoing packets with source address set to the source RLOC address determined by a local traffic engineering policy.
- to maintain a novel mapping structure that allows coupling the selection of destination and source locators.
- to manage the independent setting of priority and weight for both source and destination RLOCs associated with a given EID in mapping entries.
- to differentiate traffic control policies for different outbound flows.

C. System architecture

For the xTR system to integrate LISP-EC features, we design the mapping structure in Fig. 2; it requires modifications to user and kernel spaces. Accordingly, the mapping socket that handles the interactions between control and data plane also needs to be updated. In Fig. 3, we draw the system architecture of LISP-EC capable xTR in which new and modified processes (e.g., egress control, mapping socket, packet encapsulation) are denoted with a different color (green).

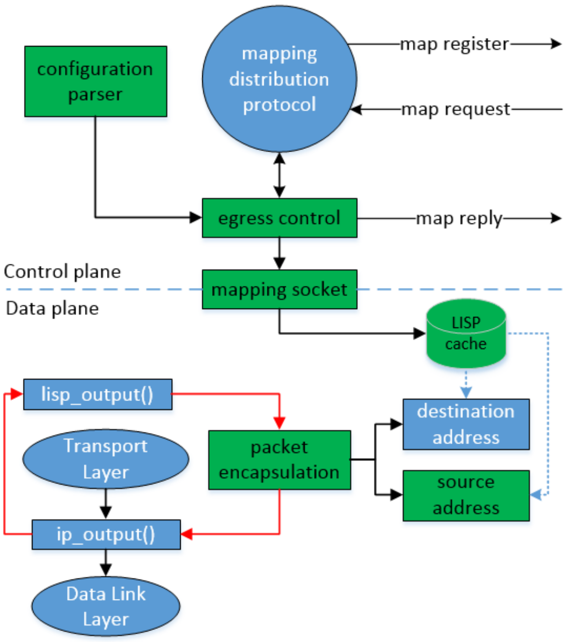

Fig. 3: LISP-EC system architecture

With the purpose of validating and manipulating the mapping entries received from destination networks before adding them into the mapping cache, the egress control module is developed by updating the MAP-REPLY processing logic. More precisely, the *read_rec*() function defined in *plugin_openlisp.c* is extended: once received a MAP-REPLY, *read_rec*() populates the mapping entry for the announced EID-prefix with destination RLOCs and associated attributes parsed from the message. Instead of employing the attached priority and weight, the extended logic allows to use an alternative set of RLOC metrics. More importantly, it is possible to associate a destination RLOC with one or several source locators. After parsing one destination RLOC, a list of *source_locator* is constructed by querying the local mapping database. The corresponding priority and weight for each source locator in the list can be statically configured or dynamically computed regarding to the employed traffic engineering policies. It is worth noting that, in order to keep track of all the received mappings in the mapping cache (own by the control-plane before transferring to the kernel mapping system), we extend the EID-RLOC mapping structure at the user space as well.

The EC traffic engineering logic, i.e., how to couple source and destination locators as well as how to combine priority and weight for source and destination locators for path selection, has to be integrated in the egress control module.

Extending from the standard procedure, the LISP-EC configuration parser allows peering relations to be established between distant edge networks, more precisely between EID-prefixes. Besides specifying its RLOCs, each EID-prefix can now be associated to a peer remote prefix from another LISP site. Once receiving a MAP-REPLY, the xTR checks for a flow control agreement between the local EID-prefixes and the announced prefixes. If local prefix x peers with remote prefix y, depending on agreed TE policy the EC module associates a subset or all source locators of x queried from the mapping database with each RLOC of y. Thanks to such a 'virtual peering' agreement between LISP sites, different control policies can be applied for the same pair of LISP sites depending on the source and destination prefixes.

Besides expanding the control-plane processing module, we also upgrade the kernel space with an extended version of the mapping socket and a novel source address selection procedure. The major modifications on those two processes could be summarized as follows.

- *Mapping socket*: to adapt with the extended mapping sent by control plane, it is needed to define an alternative message structure. The legacy mapping message format consists of a message header, followed by an EID socket address and then a list of the routing locators. The total number of locators in the list is specified in the header. Each locator is represented by its socket address, followed by a $rlocs\_mtx$ structure in which RLOC attributes are included. Our design consists in inserting a list of source RLOCs after each destination RLOC. Both source and destination RLOCs share the same format. In order to differentiate them, a new locator control flag is introduced, and for each destination RLOC, the number of associated source RLOCs is also included as a new attribute in $rlocs\_mtx$. Besides that, a different logic for message building and handling is developed at *opl_add_rloc*() in *plugin_openlisp.c* and $map\_insertrloc\_withsrc()$ in $maptables.c$ respectively.
- *Packet encapsulation*: the modifications made in packet encapsulation module could be reflected via the changes in its source locator selection process. For packets sending to peering EID-prefixes, instead of looking up the routing table, the extended $map\_select\_srcrloc()$ function queries the mapping cache to find the corresponding local gateway for selected destination locator. To enable load balancing among selected gateways, we employ a technique similar to the one implemented in OpenLISP for destination RLOC handling.

IV. PERFORMANCE EVALUATION

Extending the standard mapping structure, LISP-EC offers higher control over the inter-domain routing paths, and consequently opens opportunities for improving network performance. However it also introduces some extra operational costs at the system level. In the following, we present different experiments showing the trade-off between performance and execution time overhead introduced by LISP-EC.

A. Edge to edge delay

We simulate the edge-to-edge interconnection of two arbitrary ASes, each one has a random number of upstream providers between 2 and 6. At each AS, there is one RLOC per upstream provider. The RLOC-to-RLOC tunnels are simulated to have a random one-way delay between 20 and 250 ms. Inbound RLOC priorities are generated randomly for each simulation instance. In the simulations, we run 500 random network instances and we show the results by boxplots (showing the maximum, third quartile, median, first quartile, minimum and outliers).

We capture the RLOC-to-RLOC path choices at the two LISP sites under different TE policies. The LISP-EC TE policies previously presented are employed with the delay between source and destination RLOCs as unique performance metric.

Besides LISP-EC TE policies, we also include the 'legacy LISP' behavior (i.e., no source RLOC selection and the destination RLOC is chosen as the one with the highest destination-set priority), and a LISP-based TE approach (indicated 'Legacy LISP with TE') that overrides the destination RLOC preferences and selects the source-view best destination RLOC based on the RLOC-to-RLOC delay. For instance, let D_1 and D_2 be the two destination RLOCs, and let S_1 and S_2 be the best source locators toward D_1 and D_2, respectively, from the source viewpoint. If the delay on the S_1-to-D_1 path is less than the one on the S_2-to-D_2 path, then the D_1 RLOC priority is locally overridden by the source xTR, updating it with the smallest priority value in its mapping entry.

Delay performance simulation results are shown in Fig. 4, which report the forward delay as seen by one of the two LISP sites. As one could expect, the legacy LISP routing decision not being based on source-to-destination forward path performance criteria, it always experiences a sensibly higher delay than the TE policies. When outbound TE policies are applied, the forward delay performance is instead under control. Applying various LISP-EC TE policies, described in Section III-A, the highest gain with respect to 'legacy LISP with TE' can be observed when the best forwarding path is selected ('LISP-EC best fw path'). Controlling source RLOC selection only ('LISP-EC best src-RLOC') yields a performance gain comparable to when controlling destination RLOC selection only ('legacy LISP with TE'). Combining both source and destination RLOC selection capabilities leads to a significant improvement, as we can see in the LISP-EC best fw path case. The median edge-to-edge delay is significantly decreased: compared with legacy LISP, it offers a reduction of roughly 77%. LISP-EC policies adopting forms of collaborative TE between source and destination LISP sites, either by two-side minimization of the delay sum ('LISP-EC global optimum') or by selecting the routing equilibrium ('LISP-EC equilibrium'), are obviously a bit lower in performance with respect to the best forwarding path case (with the equilibrium policy statistically slightly outperforming the global optimum case due to the fact that this plot shows the delay performance as seen by only one of the two networks, and not their sum).

Overall, we show that the statistical gain one could get in terms of performance by applying LISP-EC TE can range from

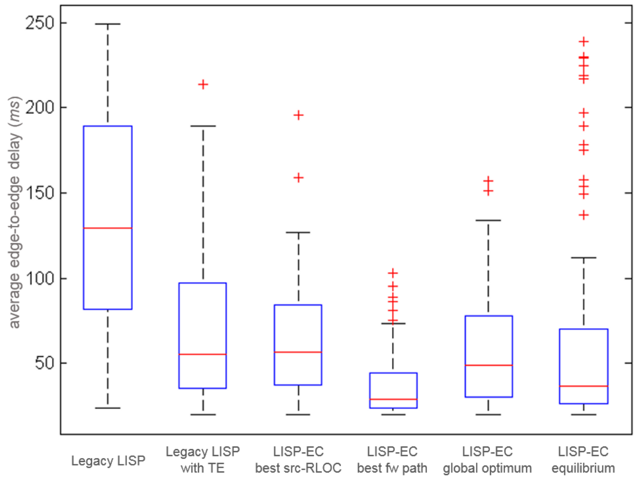

Fig. 4: Boxplot statistic of edge-to-edge delay

roughly 55% to 77% with respect to legacy LISP, and from 12% to 47% with respect to a performing TE optimization in a legacy LISP setting (i.e., without egress control).

B. *System level performance*

The benefits from enabling egress control in LISP come at a price, as it obviously introduces extra packet forwarding and control-plane delays. From a practical deployment perspective, we need a better understanding of the extended mapping structure impact on the LISP routing system. In the following, we report the system level performance of LISP-EC router in two different scenarios: (i) when adding a new mapping entry and (ii) when retrieving data from the mapping cache. In both experiments, the performance is measured in term of processing time. The experimented routers are built in FreeBSD virtual machines with one 2.397GHz CPU and 2GB of live memory. We implemented LISP-EC in the LIP6-LISP OpenLISP node, open sourcing the code [9].

In the first experiment, we measure the average delay when a new mapping entry is added into the mapping cache. It takes into account the total amount of time for parsing the MAP-REPLY, executing traffic engineering policies (associating source to destination locator, retrieving priority and weight for each RLOCs), constructing and finally adding the new mapping to the kernel space. In Fig. 5 we report the average processing time with legacy LISP and LISP-EC as a function of an increasing number of routing locators. For LISP-EC we include both the case when the egress RLOC metrics are preset, and the case when the egress metrics are computed on the fly. We refer for the latter case to the equilibrium routing computation, which has a linear time complexity [2].

Fig. 5 shows that the performance gap increases linearly with the number of locators between two LISP sites - more precisely, the number of RLOC-to-RLOC paths. We can observe that, with 4 paths, LISP-EC leads to a processing time 3 times higher than legacy LISP. Then, the router performance is strongly influenced by the number of additional locator fields appended in the mapping message sent from control-plane to data-plane spaces in the xTR. LISP-EC mapping associates each destination RLOC with a list of source RLOCs, thus multiplying the total number of locator field carried on a message. That explains for the high sensibility of LISP-EC to the number of RLOCs. However the amount of locators is restricted by the number of upstream providers, and for the large majority of edge stub ASes the number of upstream provider is less than 6, and about 2/3 less than 3 [2]. In a quite worst case scenario where each site maintains up to 5 RLOCs, it introduces a difference of 100 ms with respect to the standard LISP. As adding a mapping entry is not a frequent operation in most of LISP use-cases, such a system performance gap could be considered as unimportant.

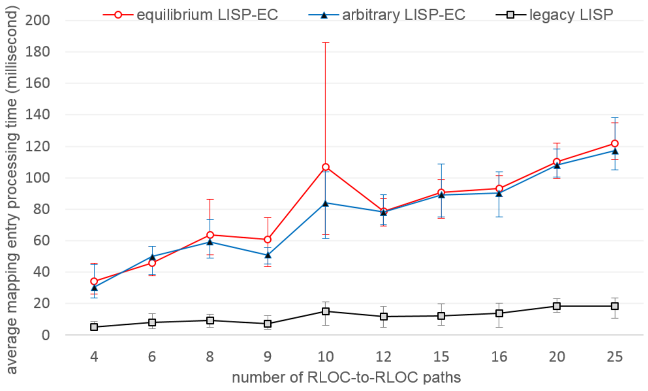

Fig. 5: Average processing time for adding a mapping entry

In the second experiment, our focus moves to the processing time overhead experienced at the kernel space where incoming packets are forwarded. We performed two cases with different mapping cache sizes to capture the amount of time taken for querying source and destination addresses while encapsulating incoming traffic: a first case when the router maintains a small mapping cache with less than 10 entries and a second case with more than 10000 entries. For both cases, we simulate the same traffic condition with more than 1000 incoming packets per second. The experimental results are reported in Fig. 6. The median processing time captured at a standard LISP router is around 4 microseconds in case of a small mapping cache, a bit higher than with LISP-EC. In the latter case with a very large mapping cache, we observe the major shift in performance: the median delay experienced with standard LISP is now lower than LISP-EC. The median processing time of LISP-EC capable router is increased from roughly 3000 ns to more than 4000 ns. It indicates the dependence of the novel source address selection with mapping cache size. However, such an overhead can be seen as negligible.

V. RELATED LISP CONTROL-PLANE FEATURES

Integrating LISP-EC traffic engineering policies in LISP could imply control-plane signaling extensions. Besides the system enhancement we described in the previous section, a LISP operator may see the need to include specific control-plane signaling in support of LISP-EC.

Among the described LISP-EC traffic engineering policies, those purely unilateral one, such as the best source RLOC

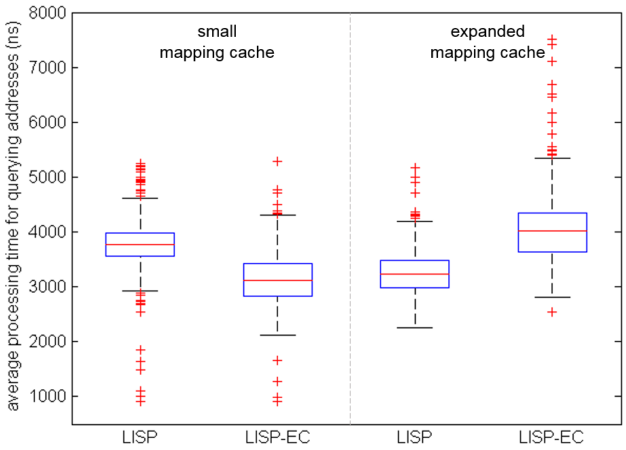

Fig. 6: Boxplot statistic of look up delay

or best forwarding path policies, rely on local information to optimize the outgoing flows of traffic, and LISP-EC specific information exchange between LISP sites is not needed.

Nevertheless, collaborative LISP traffic engineering policies such as the equilibrium and global optimum ones may benefit from a specific control-plane support. As their routing decision does rely on LISP metrics from both sites, it combines the ingress RLOC preferences of the destination network with the egress RLOC preferences of the source network.

Standard LISP distributes RLOC preferences for inbound traffic via three main mapping system messages: MAP-REGISTER, MAP-REQUEST and MAP-REPLY. We identify two possible modes to disseminate also outbound preferences:

- global outbound preferences dissemination: in this mode, the destination has the same outbound preferences independently of the source. In such a case, MAP-REGISTER messages can be extended to register both inbound and outbound preferences over the local RLOCs, provided the mapping server support such an operation mode. If such an extension is not supported by the mapping system, this could be included only at the ETR-level by extending the MAP-REPLY to also include outbound preferences, provided proxy reply (i.e., the mapping system can reply to MAP-REQUEST messages on behalf of the ETR) is not enabled by the target LISP site.
- source-specific outbound preferences dissemination: in this mode, the destination LISP site wants to reply in a different way as a function of the source LISP site, which is possible when proxy reply is not enabled, hence implementing local TE policies. In such a case, the same extension to the MAP-REPLY message addressed above can be used for this purpose.

The extensions required to MAP-REGISTER and, MAP-REPLY messages are straightforward as the outbound RLOC preferences can be included as additional RLOC objects in the control-plane message structure, and the connotation of the RLOC object (inbound or outbound) can be indicated using a flag in the available ‘Reserved’ space. In either mode, MAP-REQUEST messages can transport an explicit flag to request outbound RLOC preferences, which can also be taken from the available ‘Reserved’ space. One could easily add these features to the LIP6-LISP OpenLISP implementation, however excluding the process requiring MAP-REGISTER messages and mapping server interface update as it is a bit more cumbersome. These latter features may indeed become desirable only at a later stage of deployment.

VI. CONCLUSIONS

In this paper we proposed LISP-EC (Egress Control), an extended behavior to include outbound traffic engineering in LISP communications. The benefits of LISP-EC over the legacy system is expressed via the capability to balance traffic among upstream providers and the ability to coupling the choice of source and destination routing locators. We presented possible LISP-EC traffic engineering policies, comparing them with each other and legacy LISP behavior, showing an improvement ranging from 55% to 77%.

We implemented and released LISP-EC capable OpenLISP-based router at [9]. The implemented system allowed us to confirm the feasibility of the proposed design in working network experiments, and to validate its interoperability with the existing systems. By comparing the system level performance of LISP-EC enabled router with a standard LISP system in realistic emulated setting, we showed that traffic engineering mechanisms emerged from LISP-EC can be deployed at a low computation overhead.

ACKNOWLEDGMENT

This work was supported by the ANR LISP-Lab Project (http://www.lisp-lab.org; Grant No: ANR-13-INFR-0009).

REFERENCES

[1] D.C. Phung, S. Secci, D. Saucez, L. Iannone, “The OpenLISP Control-Plane Architecture”, *IEEE Network Magazine*, Vol. 28, No. 2, pp: 24-40, March-April 2014. https://github.com/lip6-lisp/
[2] S. Secci, K. Liu, B. Jabbari, “Efficient Inter-Domain Traffic Engineering with Transit-Edge Hierarchical Routing", *Computer Networks*, 57(4), 2013.
[3] T. Bu, L. Gao, D. Towsley, “On characterizing BGP routing table growth", *Proc. of GLOBECOM 2002*.
[4] A. Lutu, M. Bagnulo, “The Tragedy of the Internet Routing Commons", *Proc. of IEEE ICC 2011*.
[5] D. Farinacci, V. Fuller, D. Meyer, D. Lewis, “The Locator/ID Separation Protocol (LISP)", RFC 6830, 2013.
[6] E. Nordmark, M. Bagnulo, “Shim6: Level 3 Multihoming Shim Protocol for IPv6", RFC 5533, 2009.
[7] C. Launois, O. Bonaventure, M. Lobelle, “The NAROS approach for IPv6 multihoming with traffic engineering", *Proc. of QoFIS, LNCS 2811, 2003*.
[8] Y. Rekhter, T. Li, “A Border Gateway Protocol 4 (BGP-4)", RFC 1771, 1995.
[9] Routing games LIP6 open source project (website): https://routing-games.lip6.fr.
[10] F. Baker, P. Savola, “Ingress Filtering for Multihomed Networks", RFC 3704, 2004.
[11] R. Moskowitz et al., “Host Identity Protocol", RFC 5201, 2008.
[12] A. Ford et al., “TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, 2013.
[13] M. Boutier, J. Chroboczek, “Source-specific routing", *Proc. of IFIP Networking 2015*
[14] Optimized BGP multi-homing commercial solution (website): http://border6.com/?q=solutions.