# Advantages of a PCE-based Control Plane for LISP *

Alberto Castro†⋆, Martin German†⋆, Xavi Masip-Bruin†, Marcelo Yannuzzi†
†Technical University of Catalonia, Spain
{acastro, mgerman, xmasip, yannuzzi}@ac.upc.edu
Roque Gagliano⋆, Eduardo Grampin⋆
⋆University of the Republic, Uruguay
{rgaglian, grampin}@fing.edu.uy

## ABSTRACT

The Locator/Identifier Separation Protocol (LISP) is one of the candidate solutions to address the scalability issues in inter-domain routing. The current proposals for its control plane (e.g., ALT, CONS, NERD) have various shortcomings, including the potential dropping of packets at LISP routers during the resolution of the EID-to-RLOC mapping. In this paper, we introduce a new Control Plane (CP) for LISP supported by an architecture that borrows concepts from both the Path Computation Element (PCE) and Intelligent Route Control (IRC). Our CP is able to tackle three different problems simultaneously: (i) packets sourced from end-hosts are neither dropped nor queued during the mapping resolution; (ii) the EID-to-RLOC mapping can be obtained and configured approximately within the DNS resolution time needed to fetch the destination EID address; and (iii) our approach can blend IRC with the PCE capabilities, to perform upstream/downstream Traffic Engineering (TE) through the dynamic management of the mappings. In particular, our CP supports the utilization of different LISP ingress and egress local routers for the same flow sourced from a domain.

## 1. INTRODUCTION

The current discussions in the Routing Research Group (RRG) of the IRTF suggest that, scaling benefits could be realized by separating the current IP address space into two different types of address: identifiers and locators. The basic idea is to use an Endpoint Identifier (EID) to represent an end-host's address, while it's associated Routing Locators (RLOCs) describe how an end-host is attached to the internetwork. The scaling benefits arise when EID addresses are not routable through the Internet—only the RLOCs are globally routable [2].

One of the solutions under discussion at the RRG is the Locator/Identifier Separation Protocol (LISP), which has the advantage that it can be adopted readily today given its non-disruptive nature [1]. LISP uses IP-over-IP tunnels deployed between border routers located at different domains. In brief, LISP operates as follows. When a local end-host ($E_S$) wants to communicate with an end-host in a different domain, the first step is the usual look up of the destination address ($E_D$) in the DNS. Once $E_D$ is obtained, the packets sourced from $E_S$ traverse the domain and reach one of the local border routers. In a LISP-aware domain, these later are referred to as Ingress Tunnel Routers (ITRs). Since only RLOCs are globally routable, a mapping system is necessary between EIDs and RLOCs. When an ITR receives packets toward $E_D$, it consults the mapping system. After the EID-to-RLOC mapping resolution, the ITR encapsulates and tunnels packets between the local RLOC (the ITR address) and the RLOC retrieved from the mapping, namely, the Egress Tunnel Router (ETR) address in LISP terminology. At the destination domain, the ETR decapsulates the packets received through the tunnel and forwards them to $E_D$—which is assumed to be locally routable within the domain.

This approach, however, has three major weaknesses. First, the initial packets sent from $E_S$ to $E_D$ can be dropped at the ITR during the EID-to-RLOC mapping resolution. Although caching techniques are being proposed to store the mappings at ITRs, a hit might not necessarily be found, either because the mapping has aged out, or simply because it was never requested before. To cope with these issues, palliative solutions are being discussed. However, these palliatives either require some major changes to the DNS system, the addition of some debatable features to border routers, or the undesirable effect of using the Control Plane (CP) to transport data while the mapping is being resolved. Second, without using the abovementioned palliatives, LISP might considerably increase the latency to start up the communication between $E_S$ and $E_D$. At present, a TCP connection between $E_S$ and $E_D$ is established roughly around $(T_{DNS} + 2OWD_{E_S,E_D} + OWD_{E_D,E_S})$, whereas with LISP it would roughly demand $(T_{DNS} + T_{resol}^{map} + 2OWD_{E_S,E_D} + OWD_{E_D,E_S})$, assuming that ITRs and ETRs can encapsulate/decapsulate at line rate. Third, for each flow, the ITR is also used as the local ETR for the packets sent from $E_D$ to $E_S$. This is to avoid a two-way mapping resolution, which would increase even more the latency mentioned above. Clearly, this introduces a limitation in terms of inbound Traffic Engineering (TE). In light of this, we propose here a new CP that offers a promising

---

*This work was partially funded by the Spanish Ministry of Education through AECI-PCI A/014117/07.

alternative to tackle these three issues.

## 2. ARCHITECTURE PROPOSAL

Our goal is threefold. First, we aim at preventing the potential dropping of packets at the ITRs during the mapping resolution, and we want to achieve this without changing the DNS system or mixing the control and data planes. Second, we aim at obtaining and configuring the corresponding mapping during the DNS resolution process for destination $E_D$, i.e., we seek that: $(T_{DNS} + T_{resol}^{map}) \approx T_{DNS}$. Third, we aim at having the TE flexibility to choose different local ITR and ETR routers for any given flow sourced at the domain. To this end, we propose the scheme depicted in Fig. 1.

**Step 1:** $E_S$ queries DNS$_S$ to obtain $E_D$. The Path Computation Element (PCE$_S$) obtains E$_S$ by Inter-Process Communication (IPC) with the DNS (see the dashed line in Fig. 1), and computes the local RLOC to be used for the reverse mapping (i.e., for the incoming traffic from E$_D$ to E$_S$) based on TE constraints. The algorithms used to determine the ingress RLOC are inherently the same used today by Intelligent Route Control (IRC) techniques.

**Steps 2–5:** The PCEs are in the data path of the DNS servers, so the iterative queries performed by DNS$_S$ and the replies received from the corresponding DNS servers (root server, second-level server, and so on) can be transparently analyzed by the PCEs.

**Step 6:** When PCE$_D$ detects that the reply issued from DNS$_D$ carries the address $E_D$, it encapsulates the reply into a new UDP message, with source address PCE$_D$, destination address DNS$_S$, and a special transport port $P$ that will be listened by PCE$_S$ at the source domain $S$. The payload of the outer-packet contains the mapping for $E_D$. It is worth highlighting that the mapping selection performed at PCE$_D$ is made by an online IRC engine running in background, so the mapping is always known aforehand. This means that PCE$_D$ can encapsulate the answer from DNS$_D$ roughly at line rate.

**Step 7:** PCE$_S$ detects a packet toward DNS$_S$ using the port number $P$, it decapsulates the packet and forwards the DNS answer to DNS$_S$ (7a). From the outer-packet PCE$_S$ learns the address of PCE$_D$, it retrieves the mapping for $E_D$, and configures all the ITRs according to that mapping (7b). The advantage of pushing the mapping to all ITRs is that PCE$_S$ can carry out local TE actions, and move part of its internal traffic, without caring whether a mapping will be in place in the relevant ITRs after the TE optimization. The mapping information pushed to the ITRs in (7b) consists of the tuple ($E_S$, $E_D$, RLOC$_S$, RLOC$_D$), supporting the utilization of two independent one-way tunnels depending on the reverse mapping computed by PCE$_S$ during Step 1. In other words, an ITR is capable of forwarding traffic to $E_D$, using as source address in the encapsulation an RLOC that might be different from its own RLOC address.

**Step 8:** DNS$_S$ responds the DNS query to $E_S$.

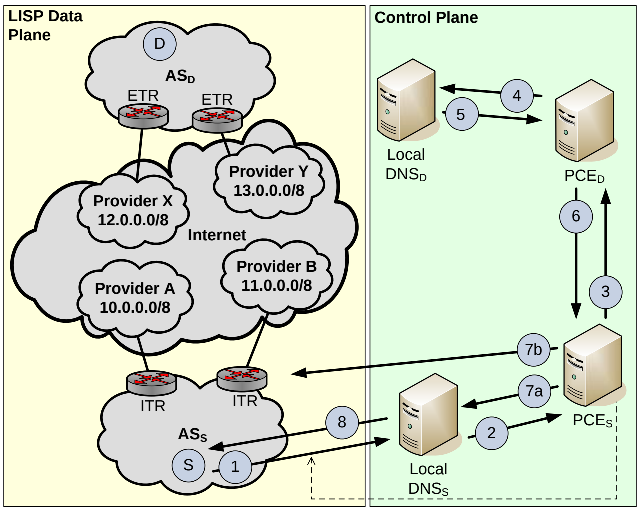

**Figure 1: Control plane architecture.**

After the usual DNS resolution process, $E_S$ starts sending packets toward $E_D$, with the advantage that the mapping has already been configured at the ITRs, hence avoiding the potential dropping of packets. The overall process (Steps 1 – 8) can be completed in approximately $T_{DNS}$, which we claim should be used as the upper bound for solving the mapping. When the first data packet reaches the corresponding ETR, this latter: (i) decapsulates the packet and forwards the inner packet to $E_D$; (ii) obtains the reverse mapping, i.e., the $E_S$-to-RLOC$_S$ mapping; (iii) pushes this mapping to the rest of the ETRs (and updates the PCE$_D$ database) via multicast. This action completes the two-way mapping resolution process. An interesting point is that our CP allows each domain to achieve its TE policies congruently, since each domain has the freedom to independently decide its ingress and egress mappings.

## 3. CONCLUSIONS AND FUTURE WORK

The approach proposed here effortlessly decouples the control and data planes in LISP. It has the advantages of neither requiring changes to the DNS system nor adding complex features to LISP routers. Our next-steps are to explore the TE opportunities of this CP in the context of Latin America, which has a number of important constraints, like the lack of inversion in networking infrastructures, as well as some noticeable peculiarities, such as the world's largest IPv4 deaggregation factor.

## 4. REFERENCES

[1] D. Farinacci, V. Fuller, and D. Meyer. Loc/ID Sep. Protocol (LISP). draft-farinacci-lisp-08, July 2008.

[2] B. Quoitin, L. Iannone, C. de Launois, and O. Bonaventure. Evaluating the Benefits of the Locator/Identifier Separation. In *Proceedings of MobiArch (ACM SIGCOMM)*, Japan, August 2007.