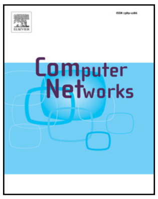
Computer Networks 91 (2015) 174–183

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



On the scalability of LISP mappings caches

Florin Coras*, Jordi Domingo-Pascual, Albert Cabellos-Aparicio

Departament d'Arquitectura de Computadors, Universitat Politècnica de Catalunya, BarcelonaTech (UPC), Barcelona 08034, Spain

ARTICLE INFO

Article history:
Received 4 October 2014
Revised 27 July 2015
Accepted 20 August 2015
Available online 29 August 2015

Keywords:
Locator/ID Separation Protocol
LISP
Cache scalability
Temporal locality

ABSTRACT

The Locator/ID Separation Protocol (LISP) limits the growth of the Default-Free Zone routing tables by creating a highly aggregatable and quasi-static Internet core. However, LISP pushes the forwarding state to edge routers whose timely operation relies on caching of location to identity bindings. In this paper we develop an analytical model to study the asymptotic scalability of the LISP cache. Under the assumptions that (i) long-term popularity can be modeled as a Generalized Zipf distribution, independent of Internet and LISP site growth and (ii) temporal locality is predominantly determined by long-term popularity, we find that LISP cache miss rate scales $O(1)$ with respect to the amount of prefixes (Internet growth) and users (LISP site growth). We validate the model and discuss the accuracy of our assumptions using several one-day-long packet traces.

1. Introduction

The growth of the Default-Free Zone (DFZ) routing tables [1] and associated churn observed in recent years has led to much debate as to whether the current Internet infrastructure is architecturally unable to scale. Sources of the problem were found to be partly organic, generated by the ongoing growth of the topology, but also related to operational practices which seemed to be the main drivers behind prefix deaggregation within the Internet's core. Diverging opinions as to how the latter could be solved triggered a significant amount of research that finally materialized in several competing solutions (see [2] and the references therein).

In this paper we focus on location/identity separation type of approaches in general, and consider the Locator/ID Separation Protocol (LISP) [3] as their particular instantiation. LISP semantically decouples identity from location, currently overloaded by IP addresses, by creating two separate namespaces that unambiguously address end-hosts (identifiers) and their Internet attachment points (locators). This new indirection level has the advantage that it supports the implementation of complex operational practices (e.g., traffic engineering mechanisms) but at the same time enables the locator space to remain quasi-static and highly aggregatable [4].

Although generally accepted that such solutions alleviate the scalability limitations of the DFZ, they also introduce new network elements, chiefly a mapping-system, and querying mechanisms needed for obtaining dynamic bindings (mappings) between the two new namespaces. To avoid growing router memory requirements with the size of the identity namespace, as is the case today, routers retrieve mappings according to user traffic. Therefore, to speed-up packet forwarding and to avoid generating floods of resolution requests, routers must store in use mappings in map-caches. This then begs the question: *does the newly introduced LISP edge cache scale?*

A considerable number of studies have empirically evaluated map-cache performance, however these results cannot be extrapolated to provide insight into what traffic parameters affect cache performance nor quantify their impact

---

* Corresponding author. Tel.: +0034 644350770.
E-mail addresses: fcoras@ac.upc.edu (F. Coras), jordi.domingo@ac.upc.edu (J. Domingo-Pascual), acabello@ac.upc.edu (A. Cabellos-Aparicio).

http://dx.doi.org/10.1016/j.comnet.2015.08.009
1389-1286/© 2015 Elsevier B.V. All rights reserved.

[5,6,7,8,9]. Additionally, in [10] and [11] we showed how the working-set [12] can be leveraged to estimate temporal locality of real network traces and finally to build a model that links miss rate and cache size. Nevertheless, despite its predictive power, the model is also unable to uncover locality sources and, ultimately, unable to predict by itself how map-cache performance is to scale.

This paper complements our work in [11] and provides an answer to the previous question by analyzing map-cache miss rate scalability with respect to Internet and LISP site growth. To this end we leverage well known results that characterize temporal locality of reference strings [13,14] to show that the relation between cache size and miss rate depends only on the popularity distribution of destinations. Notably, the result holds only if (i) long-term popularity can be modeled as a constant Generalized Zipf [15] and (ii) temporal locality is predominantly determined by long-term popularity. This further enables us to conclude that, for a given miss rate, cache size should scale constantly, $O(1)$, with respect to the growth of the Internet and LISP site size, if popularity is independent of the two. We believe this result is crucial for LISP's deployment since it ensures that if a map-cache is provisioned for a certain performance point, its performance should not degrade over time. If, however, the property does not hold, then the miss rate scales linearly, $O(N)$, with respect to the number of destinations in the Internet. To support our results, we analyzed the popularity distribution of destination prefixes in several one-day-long, real-world packet traces, from two different networks and spanning a period of 3.5 years. All assumptions were empirically validated using the traces.

The rest of the paper is structured as follows. We provide a brief overview of LISP in Section 2. In Section 3 we derive the cache model under a set of assumptions and thereafter discuss its predictions and implications for LISP. In Section 4 we present empirical evidence that supports our assumptions and evaluate the model, while in Section 5 we discuss the related work. Finally, we conclude the paper in Section 6.

2. LISP background

LISP [3] belongs to the family of proposals that implement a location/identity split in order to address the scalability concerns of the current Internet architecture. The protocol specification has recently undergone IETF standardization [16], however development and deployment efforts are still ongoing. They are supported by a sizable community spanning both academia and industry and rely for testing on a large experimental network, the LISP-beta network [17].

The goal of splitting location and identity is to insulate core network routing that should ideally only be aware of location information (locators), from the dynamics of edge networks, which should be concerned with the delivery of information based on identity (identifiers). To facilitate the transition from the current infrastructure, LISP numbers both namespaces using the existing IP addressing scheme, thus ensuring that routing within both core and stub networks stays unaltered. However, as locators and identifiers bear relevance only within their respective namespaces, a form of conversion from one to the other must be performed. LISP makes use of encapsulation [18] and a directory service to perform such translation.

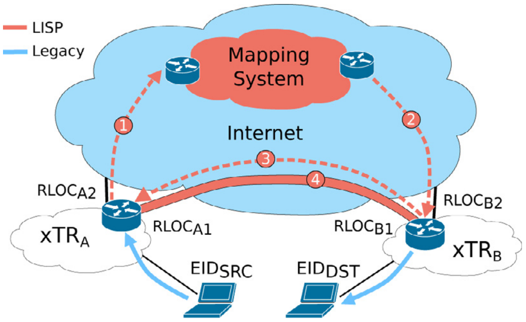


Fig. 1. Example packet exchange between EID_{SRC} and EID_{DST} with LISP. Following intra-domain routing, packets reach xTR_A which obtains a mapping binding EID_{DST} to RLOC_{B1} and RLOC_{B2} from the mapping-system (steps 1–3). From the mapping, xTR_A chooses RLOC_{B1} as destination and then forwards toward it the encapsulated packets over the Internet's core (step 4). xTR_B decapsulates the packets and forwards them to their intended destination.

Prior to forwarding a host generated packet, a LISP router maps the destination address, or Endpoint IDentifier (EID), to a corresponding destination Routing LOCator (RLOC) by means of a LISP specific mapping system [8,19]. Once a mapping is obtained, the border router tunnels the packet from source edge to corresponding destination edge network by means of an encapsulation with a LISP-UDP-IP header. The outer IP header addresses are the RLOCs pertaining to the corresponding border routers (see Fig. 1). At the receiving router, the packet is decapsulated and forwarded to its intended destination. In LISP parlance, the source router, that performs the encapsulation, is called an Ingress Tunnel Router (ITR) whereas the one performing the decapsulation is named the Egress Tunnel Router (ETR). One that performs both functions is referred to as an xTR.

Since the packet throughput of an ITR is highly dependent on the time needed to obtain a mapping, but also to avoid overloading the mapping-system, ITRs are provisioned with map-caches that store recently used EID-prefix-to-RLOC mappings. Stale entries are avoided with the help of timeouts, called *time to live* (TTL), that mappings carry as attributes. Whereas, consistency is ensured by proactive LISP mechanisms through which the xTR owner of an updated mapping informs its peers of the change. Intuitively, the map-cache is most efficient in situations when destination EIDs present high temporal and/or spatial locality and its size depends on the diversity of the visited destinations. As a result, performance depends entirely on map-cache provisioned size, traffic characteristics and the eviction policy set in place.

3. Cache model

We start this section by discussing some of the fundamental properties of network traffic that may be exploited to gain a better understanding of cache performance. Then, assuming these properties are characteristic to real network traces we devise a cache model. Finally we analyze and discuss the predictions of the model. Table 1 gives an overview of the main parameters and variables used in defining the cache model.