Inbound Interdomain Traffic Engineering with LISP

Daniel Herrmann*, Martin Turba*, Arjan Kuijper*, and Immanuel Schweizer†
*Fraunhofer Institute for Computer Graphics Research IGD, Darmstadt, Germany
Email: firstname.lastname@igd.fraunhofer.de
†Technical University of Darmstadt, Darmstadt, Germany
Email: schweizer@tk.informatik.tu-darmstadt.de

***Abstract*—Stub autonomous systems usually utilize multiple links to single or multiple ISPs. Today, inbound traffic engineering is considered hard, as there is no direct way to influence routing decisions on remote systems with BGP. Current traffic engineering methods built on top of BGP are heuristic and time-consuming. The Locator/Identifier Separation Protocol (LISP) promises to change that. In this paper, we conduct the first comprehensive evaluation of LISP and its built-in traffic engineering methods on a real-world testbed. First, we compare LISP to plain BGP and BGP advertising more specific prefixes. This comparison shows that LISP allows effective load-balancing with an accuracy of approximately 5%, while being easier to configure than BGP and its variants. Further experiments show that these results are independent from the number of concurrent streams.**

***Index Terms*—LISP, Traffic Engineering, BGP.**

I. Introduction

Stub autonomous systems (AS) are increasingly using multiple links to connect to multiple ISPs. With BGP, AS have complete control over their outbound traffic, however, engineering the inbound traffic is much harder. In the past, various heuristics for BGP (e.g., AS path prepending, more specific advertisements, etc. [8], [12], [14]) have been proposed.

Today, the Routing Research Group (RRG) of the Internet Research Task Force (IRTF) is discussing different candidates to replace BGP. While the focus of these discussions is mostly scalability and performance, additional built-in capabilities, such as traffic engineering, might be a differentiation factor. LISP is one promising new routing architecture (specified in RFCs 6830-6836 [9]), which specifies built-in traffic engineering capabilities.

Using a real-world testbed we are able to analyze LISP and the built-in traffic engineering method. Focusing on two different scenarios for multi-homed stub AS, we can show that LISP offers reliable and easy to configure traffic engineering capabilities.

This paper is organized as follows. Section II shortly introduces and discusses the traffic engineering capabilities of LISP and different heuristics for BGP. Section III explains the setup of the experiments and presents the results. Section IV compares our contribution to related work, before Section V concludes the paper.

II. Inbound Interdomain Traffic Engineering

Interdomain traffic engineering had gathered considerable attention over the past years as traffic between AS has steadily increased [4]. A survey on general Internet traffic engineering is given by Wang et al. [14]. As BGP currently is used as EGP, only BGP was investigated. It is crucial to revive these efforts with the introduction of new routing architectures [3].

In this section we will first introduce guidelines for traffic engineering with an emphasis on stub AS and inbound traffic. Then we will shortly introduce common heuristics for BGP, before discussing the capabilities of LISP.

A. Guidelines for Traffic Engineering

Interdomain traffic engineering influences the routing decisions of other AS. [6], [7] propose a set of guidelines, which are summarized into four bullet points in [14]:

1) Predictable traffic flow changes
2) Limiting the influence of neighboring domains
3) Reducing the overhead of routing changes
4) Prefer customer routes

From these guidelines, the first three are of importance for inbound traffic engineering in stub AS. The last objective, however, is only applicable for either outbound or transit traffic.

B. BGP

The Border Gateway Protocol (BGP) is a distance-vector routing protocol, currently used as the exterior gateway protocol [11]. Routing reachability information is passed hop-by-hop using route advertisements, with several attributes attached to it. Each router evaluates all incoming routes based on the attributes and only forwards the best routes to the neighbor routers. These routes are also used to forward packets.

BGP traffic engineering leverages this process by manipulating BGP attributes to influence remote routers decisions. This section will shortly explain the different possible methods and describe the methods which were selected to be compared to LISP.

1) Specific Advertisements: To improve load balancing, the IP prefix of a site can be split in two or more parts which then can be advertised in addition to the full prefix. Due to the longest prefix match rule, traffic to each prefix will be sent to the router which advertises this specific prefix. Although it only advertises slightly more routes, it violates the third guideline introduced above.

978-1-4799-3780-6/14/$31.00 ©2014 IEEE

2) AS Path Prepending: Another method is AS Path Prepending [12] (ASPP), which prepends the own AS number multiple times to the AS Path attribute. This makes this path less attractive to other routers, so that they might choose another path instead. ASPP is a heuristic method as it is hard to predict how much traffic will be shifted to other links. It has been shown that any possible incoming traffic pattern can be implemented with ASPP [8]. However, it directly violates the first guideline of predictable traffic flow changes.

Although ASPP is often used for BGP traffic engineering today, it is highly dependent on the network topology nearby the examined site. It is not feasible to rebuild an Internet-like topology to effectively implement ASPP in the scope of this paper. Hence, we chose not to include ASPP in the evaluation.

C. LISP

In traditional IP networks, an IP address in the Internet serves two purposes: It defines the identity of a given node and also defines the location of this node. LISP proposes to split these two functions of the IP address into different parts: The Endpoint Identifier (EID), which defines the identity of the endpoint and the Routing Locator (RLOC), which defines, where this EID can currently be found. RLOCs are public routable IP addresses, reachable and advertised in the Default Free Zone (DFZ), while the EIDs are only routable locally in the sites. Mapping between EID and RLOC is achieved through a mapping system, similar to the Domain Name System (DNS).

The traditional, BGP speaking AS perimeter routers at each site are replaced by LISP tunnel routers (xTR). This router is responsible for handling the LISP packets and is the only device aware of LISP at the site. It consists of two modules: the ingress Tunnel Router (iTR) is responsible for handling incoming LISP packets. It strips the LISP header and forwards the packet natively. The second component is the egress Tunnel Router, which sends native packet from the inside over the Internet to the LISP destination, by performing a LISP mapping lookup and encapsulating the packet using the RLOC addresses. Proxy Tunnel Routers (PxTR)s connect LISP sites to the traditional BGP Internet. They use BGP to announce the EID prefixes of the connected sites to the DFZ to attract all traffic destined to these sites from the public Internet and then perform LISP encapsulation.

When a locator-set contains multiple entries, LISP can either provide an active/backup solution, load balancing, or both. Entries with lower priority value are strictly preferred over entries with higher values. With multiple equal-prioritized RLOCs in a locator-set, the router performs load balancing based on the relative weight value.

III. Experiments

The following two experiments are the main contribution of this paper. The first experiment compares the load balancing capabilities of LISP to plain BGP and BGP with more specific advertisement. We focus on the multi-homed stub AS scenario. After confirming the architectural advantages of LISP over BGP, the last experiment is designed to further evaluate the stability and performance of LISP's load balancing mechanism under more dynamic traffic conditions.

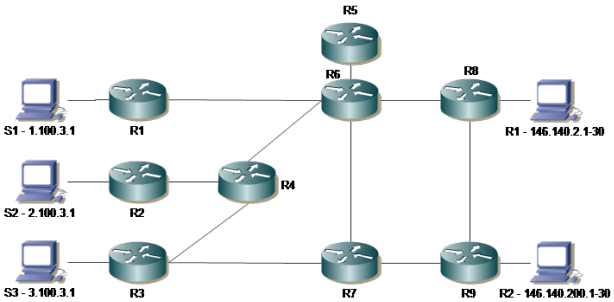

Fig. 1: Layer 3 topology of the testbed

A. Setup

To allow accurate comparison between BGP and LISP, the testbed for the following experiments contains multiple sending AS and one receiving AS. Each AS contains a virtual machine emulating multiple traffic senders and receivers.

Most of the traffic entering an AS is typically originated by only a small fraction of senders and the typical length of an AS-Path is between 4.2 and 4.5. [12]. Thus, our testbed is sufficient to emulate real-world conditions. We employ three sending sites in our topology, one LISP enabled, two BGP connected.

The next section will introduce the topology of our testbed, followed by the configuration parameters, used throughout the experiments.

1) Topology description: Figure 1 shows the Layer 3 topology. During all experiments, R1 is connected to the network via LISP, using R6 as PxTR. R5 provides the mapping system. R2 and R3 both use BGP, where R2 is single homed and R3 is dual homed. R8 and R9 are the AS perimeter routers of the target AS. When using BGP, both have a BGP session to their respective peer. When using LISP, both routers only have a default route pointing to the peer and use R5/R6 as LISP service provider devices.

At the receiving site, two servers (receiver 1 & receiver 2) listen on 30 different IP addresses each, thus representing a decent amount of receiving clients.

2) Configuration Parameters: To compare the TE methods, we measure the amount of inbound traffic at R8 and R9. Traffic is generated using the "D-ITG, the Distributed Internet Traffic generator" [1].

We assume that all paths have the same characteristics, e.g., enough bandwidth, delay etc. To capture the data for analysis, all perimeter routers send NetFlow [2] data to a management server, where the flow information is stored and analyzed. To allow for different experiments, we can tweak different configuration parameters. The most important parameters are as follows:

- *Layer 4 Protocol*: TCP is still the predominate L4 protocol [15]. Thus, we choose to use only TCP traffic for our

experiments. As the 5-tuple hash used by LISP takes L4 information into account, we generate the ports randomly.

- *Number of Streams*: An important parameter is the number of streams sent by each sender. This parameter will be evaluated in the second experiment, while the stream size is random for all experiments.
- *Size of Stream*: Every stream has a total traffic volume associated, which can be considered as the stream size, composed of the stream duration, the packet rate and the average size of the packets. In the following experiments we will choose this randomly.
- *Stream Distribution per subnet*: The distribution of traffic among the available IP space is especially important when advertising more specific prefixes with BGP. This is evaluated in Experiment 1.

During the next sections, we will discuss the results from both experiments.

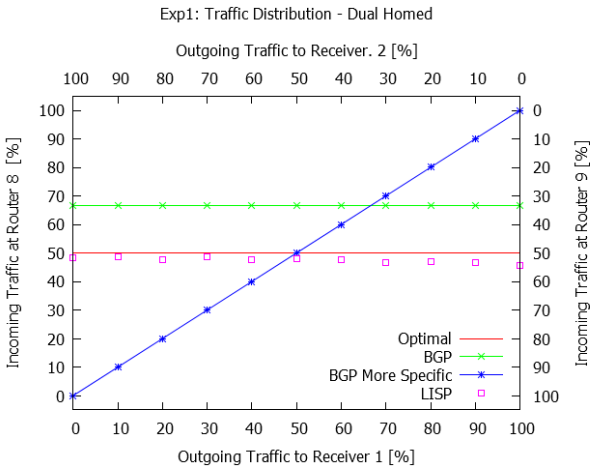

Fig. 2: Results for Load Balancing (Experiment 1)

B. Experiment 1: BGP vs. LISP using static traffic

This experiment measures the ability of LISP, plain BGP, and BGP with specific advertisements to load balance static traffic patterns. The goal, without loss of generality, is to provide 50/50 load sharing between both links without manual reconfiguration between runs. All three senders send a total of 100 streams in groups of 5 streams in parallel. In the first run, all traffic is sent to receiver 1 in the upper subnet. Each run 10% of the traffic is shifted to receiver 2. The IP addresses per receiver are chosen randomly from the respective prefixes. For BGP with specific advertisements the full prefix is split in two equal prefixes, one advertised by R8, the other one by R9.

As the scenario is static, the results for BGP are easy to predict. This is done on purpose to show the main architectural disadvantage of BGP. For plain BGP both routers announce their the full prefix to both ISPs. Both links are utilized arbitrary depending on external routing decisions. This arbitrary split will not change during the experiment. For BGP with more specific advertisements, we expect all traffic to the first part of the prefix to enter through R8, the other prefix through R9.

In LISP, both RLOCs are defined in the locator-set and configured to do 50/50 load sharing (priority 1, weight 1). In theory, LISP should load-balance the traffic regardless of the incoming traffic pattern. We expect to see values around optimal load balance.

1) Results and Discussion: The results are illustrated in Figure 2. The traffic volume to R8 and R9 is plotted over the percentage of traffic sent to receiver 1.

As expected, the traffic distribution with plain BGP on R8 and R9 is arbitrary, but constant for all traffic patterns. It depends solely on the structure of the network. This, in most cases, will not meet the requirements of the network operator, especially if the available paths have different properties.

BGP with advertisement of more specific prefixes also behaves exactly as expected. Because the more specific prefix is always used, the traffic is sent to the router originating the respective prefix. We are well aware that an easy reconfiguration between all runs would have led to a perfect load sharing. But what if traffic patterns change over time.Also, patterns might require a split into a very large number of small prefixes.

Now how does LISP compare? As expected, all measurements for LISP are close to the optimum line, we see a nearly ideal traffic distribution. LISP is close to 50/50 with 47.51/52.49 on average. The standard deviation is very low with 2.64%. The first results have shown the architectural disadvantage of BGP with respect to inbound traffic engineering. LISP has shown first promising results and the next experiment is designed to evaluate LISP under more realistic traffic conditions.

C. Experiment 2: Vary the number of streams

This experiment further evaluates LISP's traffic engineering capabilities using a varying number of inbound streams. We start by using only a single stream and then increase the number of streams from 1 up to 100 streams per sender. We also perform these measurements for all available combinations of number of senders and number of receivers, which results in a total of 600 test runs.

The destination address, destination port and the stream size are random variables to simulate different streams. While the port is completely random, the IP adress is chosen out of the 30 IP addresses of the respective prefix. The number of parallel streams is a random number between 1 and 6, calculated independently for each client.

Again, we expect LISP to load-balance the traffic equally to both receivers. However, LISP performs load-balancing on a per-flow basis, therefore we expect the load-sharing to be instable for a small number of flows. We also expect a higher deviation with a smaller amount of streams.

1) Results and Discussion: Due to space restrictions, we only show the results using two receivers in Figure 3. The traffic distribution between R8 and R9 is plotted against the number of streams used per sender.

Looking at the measurement results, we see a large deviation

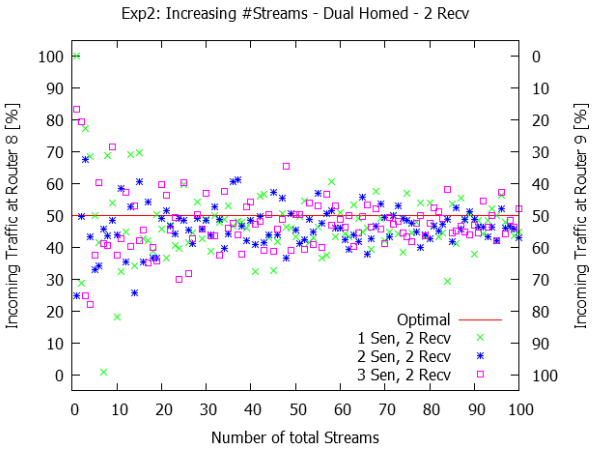

Fig. 3: Results for load balancing (Experiment 2)

when sending only a few streams. As the number of streams increases, the values converge to the ideal 50/50 line. To better understand the results, we calculate the mean square error for buckets of 10. We find that the mean square error stabilizes around $\pm 5\%$ with two receiving stations and more than 80 streams. As LISP is expected to be deployed in multi-homed stub networks, we expect the traffic volume to be quite high. It is fair to assume that the results would further improve when using more receiving stations and/or looking at more streams.

In summary, all experiments show the capability of LISP's traffic engineering method. If the accuracy can be around 5% (which is reasonable for most real-world deployments), LISP offers nearly zero-touch inbound traffic engineering.

IV. Related Work

A lot of research has dealt with incoming traffic engineering with BGP in the past, and we have seen some of the most common protocols during the last sections. The prevalence of BGP has lead to significant interest in these topics. However, up to this point only a few publication exists regarding LISP and its traffic engineering capabilities.

LISP is one of the architectural solutions being discussed by the RRG to build a more scalable inter-domain routing architecture. It was initially published in 2007 as "work in progress" [5]. Quoitin et al. evaluated the general benefits of the Locator/Identifier Separation in 2007 [13]. The authors evaluate LISP with a special emphasis on routing table size (shrinking the FIB) and route diversity in the Internet. The FIB table is greatly reduced when introducing Locator/Identifier splitting. LISP has also been evaluated in the context of intradomain traffic engineering [10]. However, neither of these papers evaluate the traffic engineering capabilities of LISP.

V. Conclusion

Current traffic engineering methods of BGP are commonly heuristic, thus, time-consuming to configure, while only providing inaccurate results. With the introduction of Locator/Identifier Separation Protocol (LISP), its built-in traffic engineering methods receive an increasing amount of attention. In this paper, we have evaluated LISP under various traffic patterns and compared it to BGP. We found LISP to provide excellent load-balancing in all our experiments, with no configuration changes (neither manual nor automatic) during all the experiments. This is the main difference to traffic engineering with BGP, which requires configuration adaptations (automatic, or even worse, manual) to provide even remotely similar results. In summary, LISP provides superior load-balancing independent of the current traffic pattern with minimal configuration effort.

Acknowledgment

We want to thank Fraunhofer IGD (Darmstadt) and Fraunhofer AISEC (Munich) for providing the hardware required for the experiments. Special thanks go to Gregg Schudell of Cisco Systems for his support whenever technical questions about LISP needed an answer. This work has been co-funded by the DFG as part of the CRC 1053 MAKI.

References

[1] A. Botta, A. Dainotti, and A. Pescapé, "A tool for the generation of realistic network workload for emerging networking scenarios," *Computer Networks*, vol. 56, no. 15, pp. 3531–3547, 2012.

[2] B. Claise, "Cisco Systems NetFlow Services Export Version 9," RFC 3954 (Informational), Internet Engineering Task Force, Oct. 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3954.txt

[3] F. Coras, D. Saucez, L. Jakab, A. Cabellos-Aparicio, and J. Domingo-Pascual, "Implementing a bgp-free isp core with lisp," in *IEEE Globecom*, 2012.

[4] "De-cix 5 year traffic statistics," https://www.de-cix.net/about/statistics/, last visited: March 20, 2014.

[5] D. Farinacci, V. Fuller, and D. Oran, "Locator/ID Separation Protocol (LISP)," Jan. 2007. [Online]. Available: http://tools.ietf.org/html/draft-farinacci-lisp-00

[6] N. Feamster, J. Borkenhagen, and J. Rexford, "Guidelines for interdomain traffic engineering," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 5, pp. 19–30, 2003.

[7] L. Gao and J. Rexford, "Stable internet routing without global coordination," *IEEE/ACM Transactions on Networking (TON)*, vol. 9, no. 6, pp. 681–692, 2001.

[8] J. Hui Wang, D. M. Chiu, J. C. Lui, and R. K. Chang, "Inter-as inbound traffic engineering via aspp," *IEEE Transactions on Network and Service Management*, vol. 4, no. 1, pp. 62–70, 2007.

[9] IETF, "IETF LISP workgroup documents." [Online]. Available: http://datatracker.ietf.org/wg/lisp/

[10] K. Li, S. Wang, S. Xu, and X. Wang, "Ermao: An enhanced intradomain traffic engineering approach in lisp-capable networks," in *IEEE Globecom*. IEEE, 2011.

[11] I. Pepnelnjak, "BGP essentials: The protocol that makes the Internet work," Nov. 2007. [Online]. Available: http://searchtelecom.techtarget.com/feature/BGP-essentials-The-protocol-that-makes-the-Internet-work

[12] B. Quoitin, C. Pelsser, L. Swinnen, O. Bonaventure, and S. Uhlig, "Interdomain Traffic Engineering with BGP," *IEEE Communications Magazine*, vol. 41, pp. 122–128, May 2003.

[13] B. Quoitin, L. Iannone, C. de Launois, and O. Bonaventure, "Evaluating the benefits of the locator/identifier separation," in *Proceedings of 2nd ACM/IEEE international workshop on Mobility in the evolving internet architecture*, 2007.

[14] N. Wang, K. Ho, G. Pavlou, and M. Howarth, "An overview of routing optimization for internet traffic engineering," *IEEE Communications Surveys Tutorials*, vol. 10, no. 1, pp. 36–56, First 2008.

[15] M. Zhang, M. Dusi, W. John, and C. Chen, "Analysis of udp traffic usage on internet backbone links," in *Ninth Annual International Symposium on Applications and the Internet*, 2009, pp. 280–281.