LISP Working Group
Internet-Draft
Intended status: Informational
Expires: July 13, 2019

A. Rodriguez-Natal
Cisco Systems
A. Cabellos-Aparicio
Technical University of Catalonia
M. Portoles-Comeras
M. Kowal
D. Lewis
F. Maino
Cisco Systems

January 9, 2019

LISP-OAM (Operations, Administration and Management): Use cases and requirements
draft-rodrigueznatal-lisp-oam-09

Abstract

This document describes Operations Administration and Management (OAM) use-cases and the requirements that they have towards the LISP architecture.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 13, 2019.

Table of Contents

1. Introduction

LISP with its location/ID split in place creates two separated namespaces, the RLOC space where the transit network elements are deployed and the EID space that applies to the end-hosts. This inherently splits the network in an underlay, represented by the RLOC space, and an overlay, represented by the EID space.

However, LISP introduces some drawbacks since relevant details of the underlay network are hidden to the overlay nodes (e.g, xTR). With LISP, an overlay node can learn about the reachability of a path towards a locator and its liveness. In terms of control, it can -by means of priorities and weights- load-balance traffic across different locators and, taking advantage of LISP-TE [I-D.ietf-lisp-te] and LISP-SR [I-D.brockners-lisp-sr], control how the traffic flows through the underlay topology. However, overlay nodes lack of appropriate knowledge about the characteristics of the paths, such as loss, latency, delay, length in IP/AS hops, etc. Furthermore, LISP nodes have little knowledge about the topological location of the RTRs as well as the characteristics of the underlay paths interconnecting them.

The mechanisms specified by LISP to monitor and control the underlay may not be enough for the complex overlay services that are arising today. Indeed, nowadays there are a plethora of services that require fine-grain control and real-time information of the network

state. Such services could take advantage of the programmable overlay scheme that LISP introduces as long as the appropriate mechanisms to control and monitor the underlay are in place.

LISP can leverage the mapping system to operate, administer, and manage the underlay-overlay relationship. Network devices can push to the Mapping System information about the capabilities and state of the network in order to allow it to take the best network operation and management decisions.

In this document we analyze the most common use-cases of overlay services and the requirements -from an abstract point of view- that they impose on the LISP architecture.

2. Definition of terms

- OAM: The term OAM is used in this document as the acronym for Operation, Administration and Management. It refers to the set of procedures and mechanism that ensure that a network deployment behaves as expected and adapts properly to new situations.

- Underlay: In this document, underlay is used to refer to the set of physical devices (i.e. hosts, routers, servers, etc) that support the networking operation in general and the LISP operation in particular. It also refers to the address space on where those devices communicate. In most cases the underlay is equivalent to the RLOC space, however it can also comprise information from external sources such traffic engineering databases, monitoring tools, etc.

- Overlay: The term Overlay is used here to denote the virtual network that sits on top of the underlay thanks to the LISP namespace split. It also refers to the address space that the virtual network uses as well as to the devices that are deployed on that address space. The overlay corresponds to the EID space.

The rest of the terms are defined in their respective documents. See the LISP specification [RFC6830] for most of the definitions, [RFC6832] for PxTR, [RFC8060] for LCAF and [I-D.ietf-lisp-te] for RTR.

3. Use Cases

3.1. General LISP operation

The overlay introduced by LISP provides an abstract view of the network that simplifies the deployment and operation of the network and its services. However this abstraction also hides the details of

the underneath physical topology. While the overlay deployment can be fully defined at a logical level, the underlay is permanently subject to physical state changes that can affect the overall performance. Any LISP deployment has to deal with both the overlay and underlay management and with underlay issues that can impact the overlay operation. In this context, the overlay needs to be aware of the underlay state in order to adapt itself to the current network conditions.

A LISP deployment where the overlay has detailed information of the underlay presents several advantages. First it can help troubleshooting the deployment. For instance, when a problem is detected, it is easy to know if it is due to misconfiguration on the LISP overlay, or rather from a physical problem on the underlay. Second, the underlay information can be used to influence policy decisions such as dynamically adapting the locators' priority and weight values based on the network state observed on the underlay. Finally, it can serve to automate the configuration of certain parts of the overlay deployment.

This is the case when underlay topological information is used to automatically select on a xTR which PxTR to use. Nowadays, PxTRs are generally manually configured, PITRs are provisioned with the EID prefixes they announce and the PETR to use is fixed on xTR boxes. With the proper overlay-underlay information exchange, these settings can be adapted over time. For instance, the PITR that is announcing an EID prefix can change to a secondary PITR in order to reduce round-trip time (RTT) if the EID prefix moves to a different RLOC, or the PETR used by a certain xTR can be replaced with a new one when the PETR goes down or the underlay network conditions change (e.g. the delay increases or the throughput decreases).

In order to provide the ability to operate with knowledge of the underlay, the LISP protocol could be extended to allow collection of underlay metrics that could then be pushed to the overlay. In terms of collected metrics, there are a few that would improve LISP operations. Some of these metrics could be extracted from the network state, by passive measurement or active probing, such as locator reachability, delay and throughput for a path, packet loss and MTU for a link, etc. Those metrics can be directly applied to the LISP policies (e.g. announcing a locator as down if it is not reachable anymore), can incrementally modify the policies (e.g. changing dynamically LISP weight values based on the observed delay or throughput), or can be applied after a threshold has been reached (e.g. setting a locator as down if the packet loss goes above a certain value). In addition to network state, it would be useful to keep track of LISP operation statistics, such as the size of the Map Cache or the last time a locator status changed. This would give

more context of the underlay state and help the overlay to make better decisions.

3.2. MPTCP

Multipath TCP (MPTCP) [RFC6824] introduces several sub-flows in a single end-to-end TCP session while keeping a legacy TCP interface to the applications. This provides both resilience and bandwidth aggregation to hosts with multiple interfaces. MPTCP capabilities are negotiated between end-systems, which includes the capability of falling back to legacy TCP if negotiation is not possible. If the other end supports MPTCP, the original TCP flow is split into several sub-flows which are then forwarded over the different available links. [RFC6824] states that MPTCP "should achieve no worse throughput" and "must be no less resilient" than a single TCP connection, beyond that baseline the room for optimization of MPTCP is limited by the network conditions over the different paths used for the sub-flows.

As a consequence of this, MPTCP is really sensitive to non-optimal conditions on different links. Moreover, in an ideal deployment, the multiple sub-flows should follow disjoint paths to ensure best link backup scenario, and/or avoid bottle-neck paths to achieve increased throughput. Another possible desirable scenario would be to forward a sub-flow, or a set of sub-flows, over a secured path to prevent a potential attacker from rebuilding the stream of data. However, there is no way to ensure that the sub-flows will follow optimal paths beyond sending them through different interfaces from the end-point. On the other hand, legacy hosts do not support MPTCP and, in that case, proxies should be provisioned for them. All of these constraints make the overlay architecture proposed by LISP a suitable scenario for MPTCP deployments. Assuming the appropriate LISP-OAM mechanisms in place, MPTCP traffic over LISP should work as follows. Consider that a MPTCP capable source sends traffic towards a non-MPTCP capable destination. The LISP overlay has relevant information about the underlay and thus knows the best topology to deliver the traffic. It enforces this topology on the underlay by defining the points the flows will go through and where the flows will just be forwarded or balanced over different links. Since the destination is not MPTCP capable, all the flows will be eventually be gathered at a proxy that will collapse them into a single flow that is forwarded to the destination. To handle the reply traffic, the single flow will first go through the proxy MPTCP and then the MPTCP subflows will be balanced again on the underlay via overlay management.

With LISP in place, and the MPTCP sub-flows being routed on the overlay, it is possible to adapt the overlay topology to match one that offers better performance for the MPTCP session. Optimal paths

may be enforced by means of using RTRs on the underlay. MPTCP proxies can be deployed at xTRs or RTRs and the traffic then routed to/from them using LISP. In order to compute this suitable topology, the Mapping System needs to be provided with several pieces of information regarding the network components themselves: which prefixes should use MPTCP for their communications, which among them are not MPTCP enabled and thus have to go through a proxy, where are these proxies located and which RTRs can be used to create the topology. The Mapping System would need to know the state of the underlay network to create the best paths among the devices. Some metrics that would be of interest to retrieve, in terms of MPTCP, are the bandwidth among the xTRs, the RTRs and the proxies, the latency observed on their connections, etc. Finally, the Mapping System needs a way to tell the participants of the overlay what to do with the traffic, i.e. it needs to tell a MPTCP proxy which EID prefixes flows should be split or merged, it needs to indicate an RTR how to balance the different sub-flows it receives among the different paths that are available, etc.

3.3. Multicast

LISP defines several options to handle multicast operation between LISP sites. [RFC6831] describes how LISP interacts with traditional multicast protocols, i.e. how multicast traffic generated and managed by multicast specific protocols are handled by LISP devices. The multicast distribution tree creation and the multicast interaction with the network is leveraged on those legacy multicast protocols. "LISP Control-Plane Multicast Signaling" [I-D.farinacci-lisp-mr-signaling] proposes an alternative method to support multicast operation among LISP sites fully supported by the LISP control-plane. It covers the signaling to build the multicast distribution tree, however how it computes the tree topology is not within the scope of the document. "Signal-Free LISP Multicast" [I-D.ietf-lisp-signal-free-multicast] proposes to connect multicast capable LISP sites through a non-multicast capable transit network. The replication is done at the LISP edge devices and the packets are forwarded via unicast on the core network. In that proposal, there is no multicast tree built on the transit network. Finally, "LISP Replication Engineering" [I-D.coras-lisp-re] describes a mechanism to build multicast distribution trees over a unicast-only transit network by means of using RTRs as multicast replication points.

In general, multicast traffic management relies on building a multicast distribution tree where the multicast source is the root and the multicast receivers are the leaves. The multicast traffic is forwarded according to that distribution tree and replicated when needed. The topology of the tree impacts both the performance of the multicast deployment and the quality of service of multicast traffic

delivery. In order to provide the best service, the multicast algorithm can use the overlay capabilities of LISP to build an optimized tree for the multicast participants based on their underlay topological location and the dynamic network conditions.

LISP-OAM mechanisms can be applied to build and maintain an optimized multicast tree. In a similar fashion to what is done in LISP-RE, underlay information can be pushed to the overlay management. In LISP-RE, the RTRs involved in the multicast process register themselves in the Mapping System, letting it know that they may be used to build the distribution tree. Beyond multicast-capable device discovery, a LISP-OAM architecture could potentially feed the Mapping System with underlay information relevant to the multicast tree computation, such as the replication capacity in the underlay devices or the latency among them. Also, the multicast policies can be enforced in detail from the Mapping System, for instance setting up some nodes for only forwarding while keeping others for both forwarding and replication.

3.4. NFV/SFC

Network Function Virtualization (NFV) is a methodology that brings the advantages of traditional server virtualization to network functions. Virtual Network Functions (VNFs) are no longer tied to the hardware and can be dynamically instantiated, moved, and modified on demand. On the other hand, Service Function Chaining (SFC) is a proposal to provide a framework to manage and orchestrate chains of service functions that are applied to traffic across the network. In both proposals, LISP can play a role, since the overlay it provides can be used to deploy or improve deployments of NFV and/or SFC. An architecture of LISP for NFV is already described in [I-D.barkai-lisp-nfv]. The applicability of LISP to support SFC is discussed in [I-D.ietf-lisp-te] and in [RFC7498]

The network functions (virtualized or not), of a LISP-based NFV or SFC deployment, will be deployed on LISP devices on the underlay (either xTRs or RTRs) and the data traffic will be managed over the overlay. The Mapping System will store the functions chains that should be applied to specific traffic and traffic engineering policies, such as the ones described in [I-D.ietf-lisp-te], will be used to ensure that traffic goes through the network functions.

Deploying NFV or SFC solutions on top of LISP, in order to leverage its overlay, requires a bi-directional communication among the underlay devices and the overlay. The overlay must discover the underlay devices that provide network functions and understand how they are connected. It also needs to know the state of both the underlay network and the underlay devices in terms of latency or

bandwidth among the devices as well as current load per device. In the NFV/SFC use-case, it is particularly important that the devices are able to announce the functions (virtual or not) that they provide, or that they are capable of providing. On the other hand, a LISP-OAM architecture for NFV/SFC must be able to program the appropriate service chains in the Mapping System and to instantiate and manage on demand VNFs in the capable devices.

4. Requirements

The use-cases presented in Section 3 show the importance of including OAM mechanisms into the LISP protocol to make a better use of the overlay-underlay architecture. Based on those use-cases, this section proposes a set of requirements that should be fulfilled by a LISP-OAM solution. These requirements may be modified and/or extended in the future based on further use-cases discussion or experimental experience. Note that each requirement is meant to cover a specific need, all of them are independent and can be individually added to LISP. However, the more requirements addressed, the better the overlay can leverage the underlay.

- Device discovery: The overlay needs to know the LISP devices (ITR, ETR, PxTR and RTR) that are available and that can be used to handle traffic. This is solved for ETRs by sending Map Register messages, that implicitly serve to announce the availability of the ETRs to the Mapping System. A similar approach can be followed to automatically discover other LISP devices.

- Capability discovery: The overlay must be aware of the capabilities of the nodes participating in the overlay, although LISP functionality is assumed in all LISP devices, the OAM mechanisms need further information. Based on the use-cases discussed in this document the capabilities to be announced by the devices are:

  * Support for MPTCP flow balancing

  * Network functions implemented on the device

  * VNFs that the device can instantiate

  * Capacity to replicate packets

  The capabilities should be encoded on a specific format (e.g a YANG [RFC6020] model in XML, a new LCAF, JSON [RFC7159] data, etc) and submitted to the overlay using LISP signaling (e.g. including capabilities information on the Map Registers) or leveraging on other existing protocols.

- Underlay state access: The overlay needs as much underlay information as possible to make the best topology and policy decisions. Underlay devices have to implement ways to collect, store and offer this information to the overlay. According to the use-cases described in this document the metrics to be collected are:

  * Latency
  * Packet loss
  * Path length (IP/AS hops)
  * MTU
  * LISP state (map-cache, locator status, etc)
  * System load
  * Replication capacity
  * VNFs instantiated

  The metrics have to be encoded (e.g. YANG, LCAF, JSON, etc) and communicated to the overlay. The way to communicate them can be either a push mechanism (e.g. Map Register) that would simplify operation but requires a central administration entry, or a pull approach (e.g Map Request) that would allow the overlay to retrieve only on-demand information. The pull mechanism also serves as a way to specify which information is relevant for the overlay and to trigger metric collection if it was not already ongoing. In any case, the underlay device may decide to limit the information that it shares with the overlay.

- Forwarding actions: Some use-cases require that the overlay defines actions on how to process packets. According to the use-cases analyzed in this document the actions are:

  * Forwarding: the basic forwarding action as defined in LISP.
  * Replicate: Replicate an EID packet and forward it to a set of RLOCs.
  * Balance flows: Distribute EID flows across different RLOCs. The flows are identified by a source/destination tuple, a 5-tuple, etc.

* Apply NF: Apply a (virtual or not) network function to the EID traffic.

These actions can be implemented as extensions to the current specifications of LISP-TE or LISP-SR, leverage on reusing existing LCAF types or be defined by means of a new LCAF. Some use-cases will narrow down actions via options, i.e. to define the algorithm to balance flows, the specific network function to be applied, etc.

Some of the required LISP extensions to support OAM may be offloaded to existing solutions, for instance using configuration protocols such NETCONF [RFC6241] to get the PETR address on an xTR, build a YANG model to express devices capabilities or instantiate VNFs via NFV specific protocols.

5. Acknowledgements

The authors would like to thank Matthieu Coudron and Stefano Secci for their feedback and helpful suggestions.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

In certain environments, multiple components of the LISP architecture may be managed in a distributed fashion (i.e., a Map Server, an ITR, and an ETR may be managed each individually by three separate organizations). When including capabilities to allow for the discovery of devices and its capabilities, as well as the collection of metrics regarding the underlay and the local device itself, it should be taken into consideration that proper controls are put in place to enforce strict policies as to which devices can access what type(s) of information.

8. Informative References

[I-D.barkai-lisp-nfv]
Barkai, S., Farinacci, D., Meyer, D., Maino, F., Ermagan, V., Rodriguez-Natal, A., and A. Cabellos-Aparicio, "LISP Based FlowMapping for Scaling NFV", draft-barkai-lisp-nfv-12 (work in progress), June 2018.

[I-D.brockners-lisp-sr]
Brockners, F., Bhandari, S., Maino, F., and D. Lewis, "LISP Extensions for Segment Routing", draft-brockners-lisp-sr-01 (work in progress), February 2014.

[I-D.coras-lisp-re]
Coras, F., Cabellos-Aparicio, A., Domingo-Pascual, J., Maino, F., and D. Farinacci, "LISP Replication Engineering", draft-coras-lisp-re-08 (work in progress), November 2015.

[I-D.farinacci-lisp-mr-signaling]
Farinacci, D. and M. Napierala, "LISP Control-Plane Multicast Signaling", draft-farinacci-lisp-mr-signaling-06 (work in progress), February 2015.

[I-D.ietf-lisp-signal-free-multicast]
Moreno, V. and D. Farinacci, "Signal-Free LISP Multicast", draft-ietf-lisp-signal-free-multicast-09 (work in progress), March 2018.

[I-D.ietf-lisp-te]
Farinacci, D., Kowal, M., and P. Lahiri, "LISP Traffic Engineering Use-Cases", draft-ietf-lisp-te-03 (work in progress), October 2018.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6824] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, DOI 10.17487/RFC6824, January 2013, <https://www.rfc-editor.org/info/rfc6824>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <https://www.rfc-editor.org/info/rfc6830>.

[RFC6831] Farinacci, D., Meyer, D., Zwiebel, J., and S. Venaas, "The Locator/ID Separation Protocol (LISP) for Multicast Environments", RFC 6831, DOI 10.17487/RFC6831, January 2013, <https://www.rfc-editor.org/info/rfc6831>.

[RFC6832] Lewis, D., Meyer, D., Farinacci, D., and V. Fuller, "Interworking between Locator/ID Separation Protocol (LISP) and Non-LISP Sites", RFC 6832, DOI 10.17487/RFC6832, January 2013, <https://www.rfc-editor.org/info/rfc6832>.

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <https://www.rfc-editor.org/info/rfc7159>.

[RFC7498] Quinn, P., Ed. and T. Nadeau, Ed., "Problem Statement for Service Function Chaining", RFC 7498, DOI 10.17487/RFC7498, April 2015, <https://www.rfc-editor.org/info/rfc7498>.

[RFC8060] Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format (LCAF)", RFC 8060, DOI 10.17487/RFC8060, February 2017, <https://www.rfc-editor.org/info/rfc8060>.

Authors' Addresses

Alberto Rodriguez-Natal
Cisco Systems
California
USA

Email: natal@cisco.com

Albert Cabellos-Aparicio
Technical University of Catalonia
Barcelona
Spain

Email: acabello@ac.upc.edu

Marc Portoles-Comeras
Cisco Systems
170 Tasman Drive
San Jose, CA
USA

Email: mportole@cisco.com

Michael Kowal
Cisco Systems
111 Wood Avenue South
ISELIN, NJ
USA

Email: mikowal@cisco.com

Darrel Lewis
Cisco Systems
170 Tasman Drive
San Jose, CA
USA

Email: darlewis@cisco.com

Fabio Maino
Cisco Systems
170 Tasman Drive
San Jose, CA
USA

Email: fmaino@cisco.com