

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 24, 2014

F. Coras
A. Cabellos-Aparicio
J. Domingo-Pascual
Technical University of
Catalonia
F. Maino
cisco Systems
D. Farinacci
lispers.net
October 21, 2013

LISP Replication Engineering
draft-coras-lisp-re-04

Abstract

This document describes a method to build and optimize inter-domain LISP router distribution trees for locator-based unicast and multicast replication of EID-sourced multicast packets.

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 <http://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 April 24, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Definition of Terms	4
3. Overview	5
4. Overlay Signaling	6
4.1. RTR Registration	6
4.2. ETR/RTR Subscription	6
4.3. ETR/RTR Unsubscription	8
5. Overlay Management	8
5.1. RLOC Failure and Unreachability	8
5.2. Other Overlay Management Considerations	8
5.3. Automated Computation of RTR Level	9
5.3.1. Algorithm for Computing Optimized Distribution Trees	9
6. Security Considerations	11
7. IANA Considerations	11
8. Acknowledgements	11
9. References	11
9.1. Normative References	11
9.2. Informative References	12
Appendix A. MADDBST heuristic	13
Authors' Addresses	13

1. Introduction

The Locator/Identifier Separation Protocol (LISP) [RFC6830] provides the mechanisms for the separation of Location and Identity semantics presently overloaded by IP addresses. The split results in the creation of two namespaces that unambiguously identify edge-site network objects, Endpoint IDs (EIDs), and core routing objects, Routing LOCators (RLOCs). Apart from aiding the scalability of the core routing infrastructure, the decoupling also enables the (re)implementation of new or existing inter-domain routing mechanisms.

One such mechanism is inter-domain IP source-specific multicast (SSM) [RFC4607]. In this sense, [RFC6831] defines the procedures carried out for delivering multicast packets from a source host in a LISP site to receivers residing in the same domain or in other LISP or non-LISP sites when an underlying multicast infrastructure exists. The signaling protocol it specifies for conveying (S-EID,G) state and building the distribution tree that connects the source ITR and the receiving ETRs is PIM [RFC4601]. An alternative method that uses Map-Requests for propagating (S-EID,G) state from ETRs to the ITR is established in [I-D.farinacci-lisp-mr-signaling].

Although desirable to use multicast routing in the core network when available, a mismatch between the multicast capabilities of receiver ETRs and source ITR might impede their interconnection. In such a case, unicast RLOC encapsulation is necessary to deliver multicast packets directly to the ETRs. This however leads to high ITR head-end replication for large sets of ETRs. Therefore, to reduce the replication load of the ITR and scale the service with the number of multicast receivers, the ITR may choose to offload replication to a set of RTRs.

The current document describes how multicast RTRs can be used to build an inter-domain distribution tree rooted at the ITR that can perform unicast and/or multicast encapsulated replication of multicast packets. This concept, of distributing the replication load from ITR to other RTRs downstream on the core overlay distribution tree, is known as Replication Engineering or LISP-RE. Since unicast replication in such overlays can be suboptimal when compared to the underlay network, methods to optimize packet delivery over the distribution tree are also presented.

This specification does not define the mechanisms used to build (S-EID,G) state in source and receiver domains, nor does it describe the messages used to propagate such state from receiver ETRs to source ITR. What it defines is how (S-EID,G) state is built in the ITR, RTRs and ETRs participating in the overlay distribution tree.

2. Definition of Terms

The terminology in this document is consistent with the definitions in [RFC6830] and [RFC6831] however, it is extended to account for LISP-RE concepts:

Delivery Group (DG): This is the outer destination address of a packet when LISP encapsulating a multicast packet with an EID source within a multicast packet.

Re-encapsulating Tunnel Router (RTR): An RTR is a router that implements the re-encapsulating tunnel function detailed in Section 8 of the main LISP specification [RFC6830]. Such router performs packet re-routing by chaining ETR and ITR functions, whereby they first remove the LISP header of ingressing packets and then prepend a new one prior to forwarding them.

Unicast Replication: Is the notion of replicating a multicast packet with an EID source address at an ITR or RTR by encapsulating it into a unicast packet. That is, the oif-list of a multicast map-cache entry can not only have interfaces present for link-layer replication and multicast encapsulation but also for site-facing interfaces and unicast encapsulation.

Overlay Distribution Tree: A degree-constrained spanning tree that represents the path followed by unicast and/or multicast encapsulated multicast packets from the root (ITR) to the leaves (ETRs) through intermediary nodes (RTRs). The ITR and RTRs unicast and/or multicast replicate packets to their tree children.

LISP Replication Node: A router (either the ITR or an RTR) participating and replicating packets downstream in the distribution tree.

Multicast Ingress Tunnel Router (ITR): An ITR as specified in [RFC6830] that supports multicast and participates as the root in the distribution tree. In this document we use the term "ITR" to mean a multicast capable ITR.

Multicast Egress Tunnel Router (ETR): An ETR as specified in [RFC6830] that participates as a leaf in the distribution tree. ETR are the only members of the tree that do not unicast replicate. In this document we use the term "ETR" to mean a multicast capable ETR.

Multicast Re-encapsulating Tunnel Router (RTR): An RTR as specified in [I-D.farinacci-lisp-te] that participates as an intermediary node in the distribution tree. In this document we use the term "RTR" to mean a multicast capable RTR.

Replication Target: A multicast channel-id (S-EID,G) or a set of multicast channel-ids (S-EID-prefix,G).

Joining-OIF-list: Represents a collection of state per multicast routing table entry at an RTR or ETR that is created by received Map-Request/Join-Request.

Forwarding-OIF-list: Represents the outgoing RLOC list a multicast router stores per multicast routing table entry such that it knows to which RLOCs to replicate multicast packets. Although the Joining-OIF-list contains sufficient information to allow the forwarding of encapsulated multicast packets, using it is inefficient. Thereby, an RTR implementation may wish to build an efficient Forwarding-OIF-list. Ways of implementing a Forwarding-OIF-list are out of the scope of this document.

Upstream: Towards the root of the tree.

Downstream: Away from the root of the tree.

3. Overview

This document describes a method to diminish the ITR's replication load by using RTRs to build an inter-domain distribution tree. The tree is managed by the source domain's Map-Server. RTRs join the overlay due to either manual or automatic configuration and advertise to the Map-Server their availability to replicate traffic for a multicast channel (S-EID,G). Out of all the RTRs registering for the same multicast channel, the Map-Server builds one mapping and organizes the RLOCs in a multi-level hierarchy. The hierarchy is rooted at the ITR and computed based on the configured information RTRs register or by means of local policy and algorithms. ETRs always join the overlay as leaves and their attachment prompts the creation of a path, which traverses the RTR hierarchy, from the ITR. The path is built at receiver request by incrementally linking all distribution tree levels, starting at the joining ETR up to the source ITR.

The way the distribution tree is built has several benefits. First, it ensures that packets in the source domain do not reach the ITR if no ETR is joined. Second, it ensures that packets are forwarded from ITR to all ETRs without mapping database lookups since the state that

defines the distribution tree, i.e., the replication hierarchy, is created prior to forwarding/replicating the packets. Finally, the multicast source is allowed to roam since a first level RTR, when informed of the roam event, can do a new database lookup to find the new ITR to join to.

It is worth pointing out that because of the receiver-initiated approach multicast employs to build distribution trees, whereby receivers join upstream sources, LISP-RE operates backwards from LISP point of view. That is, ETRs are the ones to send Map-Requests to discover potential upstream parents and the ITR answers with Map-Replies to joining downstream clients.

4. Overlay Signaling

This section describes the signaling the ITR, RTRs and ETRs use in order to participate in the overlay and build a distribution tree. The signaling messages used are described in [I-D.farinacci-lisp-mr-signaling] and [RFC6831].

4.1. RTR Registration

RTR participation in the overlay is condition by the configuration of a replication target, a multicast channel (S-EID,G) or set of channels (S-EID-prefix,G), the RTR is to perform replication for. Once configured, manually or by automated mechanisms, an RTR Map-Registers its replication target with merge-semantics to the appropriate Map-Server. In the registration it also provides its list of RLOCs to be used by overlay peers and a set of corresponding weights and priorities. If present, information about the level of the hierarchy where the RTR should attach is also conveyed by means of an Replication List Entry canonical address [I-D.ietf-lisp-lcaf].

Due to the merge-semantics, the Map-Server aggregates all RTR originated Map-Register messages in a single, per replication target mapping. If no level information is provided or if so configured, the Map-Server should use local policy to compute a hierarchy and associate a level within it to each entry in the list (more details in Section 5.3). It should be noted that the entries that are pointed to in the resulting mapping are not RLOCs but Replication List entries.

4.2. ETR/RTR Subscription

When an ETR creates (S-EID,G) state from a site based multicast join, i.e., its oif-list goes non-empty, it must send an upstream Join request. If the ETR does not have multicast connectivity to its

upstream and unicast replication must be performed, the ETR requests that a path from ITR to itself, over the RTR hierarchy be constructed. The following procedure is followed to build the path:

1. ETR sends a Map-Request/Join-Request for (S-EID,G) multicast channel to the mapping database system which further ensures its delivery to the authoritative Map-Server.
2. The Map-Server looks up the mapping associated to (S-EID,G) and, out of the distribution tree hierarchy encoded within, it selects a set of leaf RTRs, i.e., members of the level furthest away from the ITR, with spare replication capacity. The set of potential parents is encoded in a new (S-EID, G) mapping the Map-Server conveys to the ETR in a Map-Reply.
3. From the list it receives, the ETR selects the best upstream RTR RLOC according to local policy, taking into account the associated priorities and weights and sends to the owning RTR a Map-Request/Join-Request for (S-EID,G). If the ETR itself has multiple RLOCs it wishes to use in the overlay, it may convey them all to the upstream RTR encoded in the Map-Reply field of the Map-Request/Join-Request together with associated priorities and weights.
4. The RTR stores the ETR's subscription information in the join-oif-list associated to (S-EID,G) and inserts the RLOC obtained after evaluating the priorities and weights in the oif-list for (S-EID,G). It then confirms the ETR's subscription with a Map-Reply.
5. If not already a member of (S-EID,G), the RTR initiates it's own attachment to the distribution tree by repeating the steps 1-4. An important difference at step 2, the Map-Server replies to a joining RTR with a list of RTRs in the adjacent upstream layer, as opposed to a list of leaf RTRs, like in the case of an ETR join. This procedure may recurse upstream up to when the ITR or an RTR already attached to the distribution tree is joined. On completion, there should exist a path from ITR to joining ETR.
6. If the ITR is already member of (S-EID,G) the process stops. Otherwise, the ITR sends a PIM join to the intra-domain multicast source ensuring the creation of a path from the multicast source to the receiver end-hosts.

If at any point, when creating a link between two adjacent layers, native multicast replication can be performed, instead of unicast replication, the router joining its upstream could set as source of the Map-Request/Join-Request a delivery group. However, group naming

must be coordinated between the participating parties in this case, if core network replication is to be exploited.

4.3. ETR/RTR Unsubscription

When an ETR's oif-list goes empty a Map-Request/Leave-Request is sent to the upstream RTR which will result in the removal of the ETR's associated entry from the RTR's oif-list. The procedure is repeated by the RTR, and it may recurse upstream, if its own oif-list also goes empty.

When an RTR with active downstreams departs, it should first change the priority of the RLOCs it registers with the Map-Server to 255 and set its locators as unreachable in the RLOC-Probing replies it sends downstream. Finally, once all adjacent lower level members have sent Map-Request/Leave-Request messages the RTR can stop registering (S-EID,G) with the mapping database system and thus leave the overlay.

5. Overlay Management

5.1. RLOC Failure and Unreachability

RLOC failure is detected at control-plane level through RLOC-probing [RFC6830] by both upstream and downstream routers. When an RTR detects the failure of an downstream RLOC, it ceases replicating towards it. The affected RLOC is removed from the forwarding-oif-list and marked as unreachable in the join-oif-list. If a backup RLOC was provided by the downstream router in the Map-Request/Join-Request, it is instead inserted in the forwarding-oif-list and the failure results in no packet loss.

The routers downstream of a failed RTR RLOC, or who lost connectivity to said RLOCs, remove their Map-Request/Join-Request associated state and reperform the join procedure. Packet loss in this case must be solved by out-of-band mechanisms that are out of the scope of the current document.

5.2. Other Overlay Management Considerations

An overloaded RTR, i.e., one whose fan-out can not be increased, should change the priority of the RLOCs it registers with the mapping database system to 255. In such a situation, the Map-Server updates the associated mapping and informs all routers having requested it about the change through solicit Map Request (SMR) messages. Both new ETRs attaching to the distribution tree and those already connected but reperforming the join procedure must not use the RLOCs

with a priority of 255 as specified in [RFC6830]. However, routers having performed Join-Requests prior to the change should not break their existing connections to the affected RTR.

All routers part of an (S-EID,G) multicast channel should re-evaluate their attachment point to the distribution tree whenever the Map-Server updates the associated mapping. This ensures the overlay member routers attach to the best suited parent when new RTRs join or previously attached ones stop being overloaded. Change of a parent should be done following a "make before break" procedure. Specifically, the router changing attachment point first connects to the new parent and only afterwards sends the Leave-Request.

When a downstream RTR subscribes to a set of channel-ids (S-EID-prefix,G) using multiple RLOCs in a load-balancing configuration, the upstream RTR may choose to load-split channel-ids (S-EID,G) over the given set of RLOCs.

5.3. Automated Computation of RTR Level

Operators wishing to automate the RTR joining procedure may wish to use an algorithm for computing an optimized distribution tree. The algorithm could be implemented in the Map-Server and its output should be used to associate to all RTRs a level in the distribution tree. Due to the centralized management, on-line switching between algorithms may be possible in accordance to the required distribution tree performance. However, their use of such algorithms is dependent on the presence of overlay topological information. Ways of obtaining topological information will be discussed in future versions of this document.

5.3.1. Algorithm for Computing Optimized Distribution Trees

The current document does not recommend an algorithm for computing optimized distribution trees. However, it provides as an example a low computation cost heuristic, which, in the scenarios simulated in [LCAST-TR], can produce latencies between the ITR and the multicast receivers close to unicast ones. Its choice is to be influenced by operational requirements and the hardware constraints of the equipment in charge of running it. Future experiments might result in a recommendation.

In what follows, we use the term "distance" when referring to a relative length or amplitude of a metric, observed on a path connecting two points, but when the exact nature of the metric is of no interest.

Considering as goal the delivery of content for delay sensitive

applications, the function the algorithm minimizes is the maximum distance (e.g. latency or number of AS hops) from a multicast receiver to the ITR source. Notice that the reference is the multicast receiver host and not an ETR. Thus, what matters in deciding a member's position in the distribution tree is not solely its distance to the ITR but also the number of multicast receivers it serves. Then, a router close to the source but serving few receivers might find itself lower in the distribution tree than another with a slightly higher distance to the source but with a larger receiver set. The algorithm optimizes the quality of experience for multicast receivers and not for tunnel routers.

The problem described above, that searches for a minimum average distance, degree-bounded spanning tree (MADDBST), can be formally stated as:

Definition: Given an undirected complete graph $G=(V,E)$, a designated vertex r belonging to V , for all vertices v in V , a degree bound $d(v) \leq d_{\max}$, d_{\max} a positive integer, a vertex weight function $c(v)$ with positive integer values, and an edge weight function $w(e)$ with positive values, for all edges e in E . Let $W(r,v,T)$ represent the cost of the path linking r and v in the spanning tree T . Find the spanning tree T of G , rooted at r , satisfying that $d(v) \leq d_{\max}$ and the distance to the source per multicast receiver is minimized.

The heuristic used to solve this problem works by incrementally growing a tree, starting at the root node r , until it becomes a spanning tree. For each node v , not yet a tree member, it selects a potential parent node u in the tree T , such that the distance per receiver to r , is minimized. At each step, the node with the smallest metric value is added to the tree and the parent selection is redone. The pseudocode of the heuristic is provided in Appendix A.

[SHI] and [BAN] have previously defined and solved similar optimization problems. Shi et al. [SHI] also prove that a particular instance of the problem, where all vertices have weight 1, is NP-complete for degree constraints $2 \leq d_{\max} \leq |V|-1$.

The algorithm can optimize an unicast overlay however, it should not be used to optimize multicast underlay delivery. As a result, if multicast is used as underlay between part of the overlay members, once one of the members of such Delivery Group is added to the distribution tree, the others should be marked as attached also. These nodes should receive multicast encapsulated multicast packets from the chosen node over the underlying multicast distribution tree.

Finally, since the RTRs do not replicate packets for multicast receiver hosts, prior to applying the MADDBST heuristic, a Minimum Spanning Tree (MST) algorithm should be used to compute the RTR distribution tree. In this case, the MADDBST heuristic should start attaching ETRs having as input the tree resulting from MST.

6. Security Considerations

Security concerns for LISP-RE the same as for [RFC6831] and [I-D.farinacci-lisp-mr-signaling].

7. IANA Considerations

This memo includes no request to IANA.

8. Acknowledgements

The authors would like to thank Noel Chiappa for his technical and editorial commentary.

9. References

9.1. Normative References

- [I-D.farinacci-lisp-mr-signaling]
Farinacci, D. and M. Napierala, "LISP Control-Plane Multicast Signaling", draft-farinacci-lisp-mr-signaling-03 (work in progress), September 2013.
- [I-D.farinacci-lisp-te]
Farinacci, D., Lahiri, P., and M. Kowal, "LISP Traffic Engineering Use-Cases", draft-farinacci-lisp-te-03 (work in progress), July 2013.
- [I-D.ietf-lisp-lcaf]
Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format (LCAF)", draft-ietf-lisp-lcaf-03 (work in progress), September 2013.
- [RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, August 2006.
- [RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for

IP", RFC 4607, August 2006.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, January 2013.

[RFC6831] Farinacci, D., Meyer, D., Zwiebel, J., and S. Venaas, "The Locator/ID Separation Protocol (LISP) for Multicast Environments", RFC 6831, January 2013.

9.2. Informative References

[BAN] Banerjee, S., Kommareddy, C., Kar, K., Bhattacharjee, B., and S. Khuller, "Construction of an efficient overlay multicast infrastructure for real-time applications", INFOCOM , 2002.

[LCAST-TR] Coras, F., Cabellos, A., Domingo, J., Maino, F., and D. Farinacci, "Lcast: Software-defined Inter-Domain Multicast", Technical Report <http://personals.ac.upc.edu/fcoras/lcast-tr.pdf>, 2012.

[SHI] Shi, S., Turner, J., and M. Waldvogel, "Dimensioning server access bandwidth and multicast routing in overlay networks", NOSSDAV , 2001.

Appendix A. MADDBST heuristic

```
INPUT: G = (V,E); r; dmax; w(u,v); c(v); u, v in V
OUTPUT: T

  FOREACH v in V DO
    delta(v) = w(r,v)/c(v);
    p(v) = r;
  END FOREACH

  T takes (U = {r}, D={});
  WHILE U != V DO
    LET u in U-V be the vertex with the smallest delta(u);
    U = U U {u}; L = L U {(p(u),u)};
    FOREACH v in V-U DO
      delta(v) = infinity;
      FOREACH u in U DO
        IF d(u) < dmax and
           W{r,u,T} + w(u,v)/c(v) < delta(v) THEN
          delta(v) = W{r,u,T} + w(u,v)/c(v);
          p(v) = u;
        END IF
      END FOR
    END FOR
  END WHILE
```

Figure 1

Authors' Addresses

Florin Coras
Technical University of Catalonia
C/Jordi Girona, s/n
BARCELONA 08034
Spain

Email: fcoras@ac.upc.edu

Albert Cabellos-Aparicio
Technical University of Catalonia
C/Jordi Girona, s/n
BARCELONA 08034
Spain

Email: acabello@ac.upc.edu

Jordi Domingo-Pascual
Technical University of Catalonia
C/Jordi Girona, s/n
BARCELONA 08034
Spain

Email: jordi.domingo@ac.upc.edu

Fabio Maino
cisco Systems
Tasman Drive
San Jose, CA 95134
USA

Email: fmaino@cisco.com

Dino Farinacci
lispers.net

Email: farinacci@gmail.com

LISP
Internet-Draft
Intended status: Informational
Expires: August 18, 2014

Y. Hertoghs
M. Binderberg
Cisco Systems
February 14, 2014

End Host Mobility Use Cases for LISP
draft-hertoghs-lisp-mobility-use-cases-00

Abstract

This memo proposes use cases for LISP in the area of end Host mobility. The applicability of end host mobility can be found in data centers, where Virtual Machines (VM's) can be moved freely from one physical server onto another physical server, independent of location, without having to change the IP/MAC-addresses inside those VMs, nor impacting traffic flows to and from those VMs. Wireless end hosts are another area of applicability. Although this draft will not address wireless end host mobility, most of the same principles apply.

Traditionally L2 extension technologies have been used to handle mobility events, but they could lead to suboptimal routing of traffic to and from the end host after the mobility event, as well as created big broadcast domains. This memo describes how LISP solves the traffic optimization issues caused by a mobility event of an end host like a Virtual Machine, as it decouples the identity of the end host from its location, such that traffic will always be forwarded to the correct location. More-over the LISP control plane can be leveraged to discover and distribute the reachability information of end hosts such that end to end broadcast domains, and their associated problems, are no longer needed.

Various sub-use cases will be looked at in this draft, depending on whether mobility is achieved at L2 (using MAC-addresses as EID) or at L3 (using IP addresses as EIDs), and whether subnets are L2 extended across LISP sites or not. This memo also describes how to handle mobility in the case where the default gateway of the end host is not capable of performing the LISP map-and-encap function, while the LISP xTR function is located one or more L3 hops away from the default gateway.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119]

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 <http://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 August 18, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Terminology	3
2. Introduction	3
3. LISP Use Cases for Mobility	5
3.1. End Host Mobility, extended subnet, IP as EID	6
3.2. End Host IP Mobility, non-extended subnet, IP as EID	7
3.3. End Host L2 Mobility, extended subnet/VLAN, MAC-address as EID	9
3.4. End Host Mobility, using a Combined L2/L3 LISP solution	9
3.5. End Host Mobility, using a Unified L2/L3 LISP solution	10
4. LISP Multi-hop Mobility	11
4.1. Overview	11
4.2. LISP Use Cases for Multihop Mobility	13
4.2.1. End Host IP Mobility, non-extended subnet	13
4.2.2. End Host IP Mobility, extended subnet	14

5. Protocol Extension Requirements	15
6. Acknowledgements	15
7. IANA Considerations	15
8. Security Considerations	15
9. References	16
9.1. Normative References	16
9.2. Informative References	16
Authors' Addresses	17

1. Terminology

LISP specific terminology such as Ingress-Tunnel-Router (iTR), Egress-Tunnel-Router (eTR), xTR, Proxy-iTR (PiTR), Proxy-eTR (PeTR), PxTR, Endstation IDentifier (EID), Routing Locator (RLOC), Mapping-Server (MS), Mapping-Resolver (MR), MS/MR can be found in [RFC6830] and [RFC6832].

2. Introduction

Moving a network node around while keeping it's IP address can be addressed in multiple ways. One way is to put the LISP xTR functionality on the mobile node, as described in [I-D.meyer-lisp-mn]. Another approach is to offload the mobility support to the site network. We divide the overall network into sites, with every site having one or more xTRs. Within the site mechanisms must be in place to detect a mobile host.

LISP [RFC6830] is a routing architecture that separates the location from the identity of hosts. A Host is part of a prefix known as an Endpoint Identity (EID), and an EID is attached to a LISP site. Traffic between EIDs at different locations is encapsulated by LISP xTR's, and the outer header's source and destination IP addresses are set to the source and destination Routing Locators (RLOCs) associated with the source and destination LISP Site. LISP eTR's register local EIDs and their associated RLOCs with the LISP Mapping System through using LISP Map-Register Messages, while LISP iTRs request the destination RLOC for a given destination EID from the LISP Mapping System through LISP Map-Request Messages and associated LISP Map-Reply Messages.

As such it can be used to offer mobility support to end hosts e.g. to Virtual Machines (VMs) in data center networks, through the LISP xTR's registering the mobile end hosts IPv4 (/32), IPv6 (/128) and/or MAC-address (/48) as host-routes to the Mapping System, next to the existing least-specific LISP site EIDs where these hosts belong into. The LISP RLOC namespace functions as an underlay, while end host to end host communication is across the overlay in the EID namespace.

Moves between sites are handled through a combination of LISP Map-Notify and LISP Solicit-Map-Request Messages.

Figure 1 shows the reference architecture diagram we will use throughout this document. It shows two locations i.e. LISP sites A and B, with their specific xTR's XTR_A and XTR_B and Routing Locators (RLOCs) IP_A and IP_B. Note that within a typical Data Center architecture, these LISP sites can be as small as a set of interfaces where the end hosts are attached to on the (virtual) switch performing as a LISP xTR, or can be as large as a set of VLANs/EID subnets under one common Router performing as a LISP xTR. As such in these use cases, the name LISP site is a bit misleading as multiple 'LISP sites' can exist in one physical site, or even a rack in a data center, and is really dependent on the physical placement of the LISP xTR function. Typically a LISP xTR is placed as close as possible to the end hosts, and as such the scope of the LISP site is small. See [draft-moreno-lisp-datacenter-deployment] for some deployment considerations.

Figure 1 also depicts a couple of end hosts X, Y and Z, each associated to a LISP Instance ID (IID) and a MAC-address based EID (/48) and/or an IP address based EID (/32 or /128). The IID used is dependent on the use cases, and can be used to identify a L2 instance or a L3 instance.

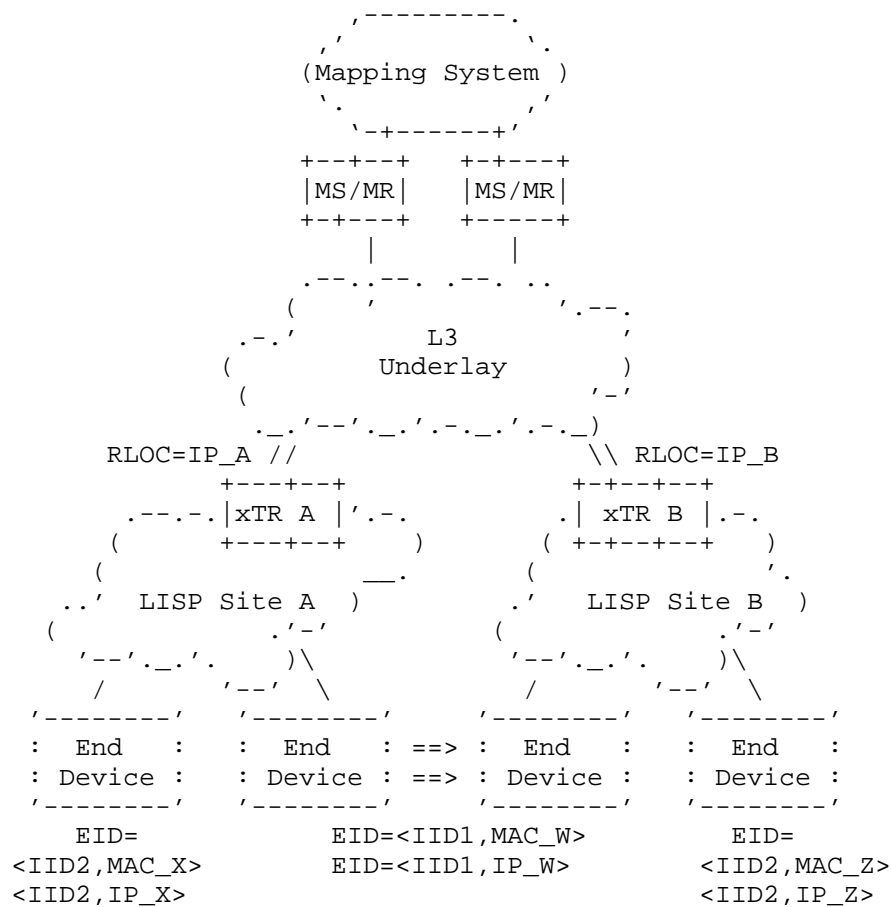


Figure 1: LISP End Host Mobility Reference Architecture

3. LISP Use Cases for Mobility

The following sections address the various ways in how LISP can be utilized to achieve end host mobility. The use cases differ in whether the end hosts subnet is extended towards the destination location or not, and whether IP-addresses (supporting IP flows) or MAC-addresses (supporting IP and non-IP traffic flows) are used to achieve mobility, while insuring no loss of sessions to and from the end host. The following use cases are addressed:

1. End host mobility, where the end host subnet is extended across the locations using some non-LISP L2 extension technique, where the EID is an IP address.

2. End host IP mobility, where the end host subnet is not extended across the locations, where the end host identifier is an IP address.
3. End host L2 mobility, by using the MAC-Address as an EID. This effectively makes LISP a L2 extension technology, but without the disadvantage of traditional L2 extension techniques.
4. A Combined L2/L3 LISP based mobility solution, where Intra-subnet traffic is handled using the MAC-address as EID, while inter-subnet traffic is handled using the IP address as EID. This effectively combines use case 1 and 3.
5. A Unified L2/L3 LISP based mobility solution, where non-IP traffic is handled using the MAC-address as EID, while IP Intra- and Inter-subnet traffic are handled using IP addresses as EIDs. This is a combination of a generalized version of use case 2 and use case 3.

3.1. End Host Mobility, extended subnet, IP as EID

This use case caters for both IP and non-IP traffic.

xTR's at the site are configured with a set of prefixes containing EIDs of hosts which are mobile-eligible. This effectively causes discovered hosts to be registered as host-routes with the LISP Mapping System by the eTR function at the site. This has the advantage that traffic towards the mobile hosts will always be routed towards the correct site. Two (or more) sites have the same subnet/ prefixes configured, and the subnet is extended across them using a L2 extension. Inter-subnet IP traffic is therefore handled by LISP, while non-IP and Intra-subnet IP traffic is taken care of by L2 forwarding across the L2 extension. How the L2 extension works and how loop avoidance is achieved is out of scope of this memo.

When a Host (e.g. Host X in instance 2 Figure 1) wants to send an IP packet to a moved Host (e.g. Host W in instance 1), which is in a different, but locally configured subnet, the local xTR (xTR_A) will route those packets across the LISP overlay to the correct destination rather than locally route them. When the same Host sends a non-IP packet, or an IP packet destined within the same subnet (e.g. host Z in instance 2), it will be sent across the L2 extension. Care needs to be taken that every site has a default gateway configured for the same prefix, and it uses the same (virtual) MAC-address in order to allow traffic from hosts to exit out of the local xTR rather than getting L2 switched back to another site. First Hop Redundancy Protocol (FHRP) originated packets (such as VRRP) have to be filtered between the two sites such that they

cannot cross the L2 extension, and the FHRP protocol has to be configured identical in both sites, such that the virtual MAC-address of the default gateway is identical. In this way, the moved Host will always find the 'same' default gateway, irrespective of its location. Hosts that are moving between sites will be discovered by the local xTR, and they will inform other xTR's which are connected to the extended subnet via LISP Map-Notify Messages. xTRs receiving traffic for a moved host will use LISP Solicit-Map-Request Messages to aid in clearing up the state of any eventual stale information at other xTRs.

For a discussion of the use case where the hosts are one or more L3 hops away from the site xTR, see Section 4.

3.2. End Host IP Mobility, non-extended subnet, IP as EID

This use case caters for IP traffic only. It is assumed that the LISP xTR in the site is the default gateway for those hosts that want mobility.

In this use case unique per-site IP EID prefixes are pre-configured in various LISP sites, and their location has been registered by the LISP eTR function at the site. A block of prefixes, part of the IP EID namespace associated with a site, can be configured across all sites as 'mobile-eligible'. If an xTR notices that one of these mobile-eligible prefixes match locally configured EID prefixes, the xTR will mark that site as 'home' of the prefix, when registering the prefix with the LISP Mapping System (standard LISP operation). The remaining prefixes are therefore 'away', when seen from that site perspective. All hosts discovered at an 'away' site which are part of one of those 'mobile-eligible' prefixes are registered with their /32 or /128 host route with the LISP Mapping System. In summary, prefixes are registered at home sites, and host-route prefixes are registered at away sites.

When a host (e.g. Host W in Figure 1) moves from its LISP 'home site' A to LISP Site B, xTR_B will notice the new Host, and will determine that it is part of a 'mobile-eligible' prefix. It will then register the new location of Host W with the LISP Mapping System, and inform xTR_A of the fact that it has 'gone away' through LISP Map-Notify Messages. Sources sending traffic to the moved Host W might still send traffic to the old location site A. When receiving LISP encapsulated traffic, xTR_A will notify the origin LISP xTR that it needs to update its mapping cache (this can be a LISP xTR, or a LISP PxTR in case the source is not part of a LISP site) via LISP Solicit-Map-Request Messages. That specific xTR will then consult the Mapping System to achieve the correct location of the end host, and update its local cache.

This use case assumes that LISP xTRs are able to 'discover' hosts within the site which are part of the configured 'mobile-eligible' prefixes. This discovery can be triggered as a result of an ARP or IPv6 ND packet sourced by the Host, or by gleaning the source IP traffic of packets sourced from the Host.

The LISP xTR can be quite centrally located within the site i.e. a couple of L2 hops (or even L3 hops, see Section 4) away. In the case where the LISP xTR is a couple of L2 hops away, care needs to be taken making sure that ARP and IPv6 NDs tables of other hosts part of the same subnet as the mobile Host are synchronized after the move. There are three broad ways of achieving that:

1. When a LISP xTR gets notified of a Host that has 'moved away', it will issue an IPv4 GARP or an IPv6 Unsolicited Neighbour Announcement (NA) towards all hosts which are local. The GARP or NA will contain the MAC-address of the LISP xTR rather than the host's MAC-address.
2. All traffic between hosts is always forced to be hairpinned through the local LISP xTR (i.e. all traffic from hosts underneath the xTR will flow 'through' the xTR, even intra-subnet traffic) and the LISP xTR can therefore catch and respond to all ARP and ND requests from hosts with a well-known per-subnet MAC-address that is shared between all xTRs that take care of 'mobile-eligible' prefixes. The hairpinning can be achieved by installing forwarding rules in the L2 switches underneath the LISP xTRs, achieving the hairpinning result, or by placing the LISP xTR function L2 adjacent to the mobile hosts (e.g on the Top Of Rack/End of Rack/Virtual Switch). How this is accomplished is out of scope of this draft. Every host's ARP/ND table will therefore have entries pointing to the same MAC-address.
3. LISP xTRs register all active hosts with both their IP EID as well as their MAC EID. All traffic between hosts is always forced to be hairpinned through the local LISP xTR (see above), and the LISP xTR will do a LISP database lookup, either to respond on behalf of the Host with the real MAC-address of the Host, or by relaying the ARP/ND end to end. The LISP xTRs will also route all IP packets independent of their destination MAC-address, regardless of whether the destination is local or remote.

For a discussion of the use case where the hosts are one or more L3 hops away from the site xTR, see Section 4.

3.3. End Host L2 Mobility, extended subnet/VLAN, MAC-address as EID

This use case caters for both IP and non-IP traffic, by treating the IP traffic as L2 traffic. End hosts MAC-address /48 host routes are registered with the Mapping System rather than IP prefixes and IP host routes, effectively creating a LISP L2 extension solution.

[I-D.smith-lisp-layer2] documents how LISP can be used to register and forward based on MAC-addresses, while the underlay is IP based. LISP xTRs performing the L2 forwarding can be placed at any place in the hierarchical network topology of the site, and there is no need to hairpin all traffic upstream through the site's LISP xTR. However, the L2 nodes on a specific site have to clear their respective bridge table entry for all hosts that moved away. How this is done as a result of a host moving is out of scope for this draft, although the most easy way is to colocate the LISP xTR function with the switch L2 adjacent to the 'mobile-eligible' hosts.

Loop avoidance in case of two LISP sites L2 interconnecting by some means unknown to LISP is needed, but is out of scope of this draft.

Although traditional bridging ('flood-and-learn') is used within the site, inter-site flooding is prevented, assuming that all active mobile hosts are registered with the Mapping System, and as such no flooding to unknown destinations needs to be performed as all hosts are known. Optionally the IP addresses can also be registered with the Mapping System, and this can aid in not having to broadcast ARP and ND packets across LISP sites, as the local LISP xTR can respond to the ARP/ND on behalf of the end-station. In case the ARP/ND packets do need to be relayed to the correct destination, registering the IP next to the MAC-address makes this easy to achieve and effectively turns the ARP/ND MAC level broadcast/multicast into an IP unicast in the underlay. MAC-level multicast and broadcasts can be encapsulated into multicasts in the RLOC namespace, or can be ingress replicated to the correct destination sites, using the techniques in [RFC6831] and [I-D.farinacci-lisp-mr-signaling]. This significantly reduces the need for multicast in the underlay.

For a Network Virtualization Overlay (NVO3) specific based implementation and for a description of LISP Messages used when hosts move between sites, see [I-D.maino-nvo3-lisp-cp].

3.4. End Host Mobility, using a Combined L2/L3 LISP solution

This use case caters for both IP and non-IP traffic. This use case is effectively a combination of use case 1 (Section 3.1) and use case 3 (Section 3.3), where LISP provides the L2 extension functionality required.

Hosts IP addresses and MAC-addresses are independently registered with the LISP Mapping System upon discovery. The placement of the xTR functions for both namespaces needs to be co-located on the same device. This functionality is often referred to a 'Integrated Routing and Bridging'. The combined L2/L3 LISP xTR can be placed at any place in the hierarchical network topology of the site, and there is no need to hairpin all traffic upstream through the site's LISP xTR, although making the LISP xTR function colocate within the switch L2 adjacent to the mobile hosts can have its advantages, see Section 3.3. All LISP xTR's which offer mobility support for the same prefix, need to be configured with the same virtual MAC-address, such that hosts will think they talk to the same default gateway independent of which site they are located at.

Intra-subnet traffic (that includes both IP and non-IP traffic) is handled within the MAC-address EID namespace as per Section 3.3 , where as Inter-subnet traffic is handled within the IP-address EID namespace as per Section 3.1.

3.5. End Host Mobility, using a Unified L2/L3 LISP solution

This use case caters for both IP and non-IP traffic. It differs with the use case in Section 3.4 in that it handles all IP traffic i.e. Intra-subnet and Inter-subnet within the IP EID namespace, while it handles all non-IP traffic within the MAC-address EID namespace. In other words it is a combination of the use case in Section 3.2, and the use case in Section 3.3 for non-IP traffic. Again the L2 and L3 xTR functions need to be colocated on the same physical node.

Since the Unified L2/L3 is also responsible for intra-subnet IP forwarding, all traffic from within the subnet/VLAN needs to be hairpinned across the Unified L2/L3 LISP xTR. The simplest way to achieve this is to make it L2 adjacent to the mobile hosts. Other methods of achieving this hairpinning are out of scope of this draft. As a result the notion of a subnet equaling a broadcast domain goes away. The subnet is purely used as a common address pool across participating LISP sites. The Unified L2/L3 LISP xTR will route all IP packets, independent of their destination MAC-address and whether these packets are part of Intrasubnet or Inter-subnet flows.

An important distinction of this use case is the fact that, for IP traffic, also the MAC-address will be registered, next to the IP address with the Mapping System. This allows the LISP xTR to optimise ARP and IPv6 ND handling for intra-subnet traffic. For non-IP traffic, the MAC-address of the host is also registered as a separate entry with the LISP Mapping System.

The Unified L2/L3 LISP xTRs are configured with a uniform default gateway MAC-address and IP address across all LISP sites. Mobile hosts will thus think they talk to the same default gateway independent of which site they are located at.

For a Network Virtualization Overlay (NVO3) specific based implementation of the Unified L2/L3 LISP solution and for a description of LISP Messages used when hosts move between sites, see [I-D.hertoghs-nvo3-lisp-controlplane-unified].

4. LISP Multi-hop Mobility

Depending on how much the mobile host can cooperate one may need to detect the mobile host on the next layer 3 hop that is connected to the mobile host, which is not necessarily identical with the xTR of the site. As a result we need to separate the mobile host detection from the xTR. The detection node - which we want to name First Hop (FH) from here on - needs to signal the address information of the detected mobile host to the site xTR(s).

4.1. Overview

Figure 2 shows how a host H1 has moved from the LAN attached at node FH-1 to the LAN attached at node FH-2. This goes undetected by xTR-2 as it is not L2 adjacent to the moved host H1.

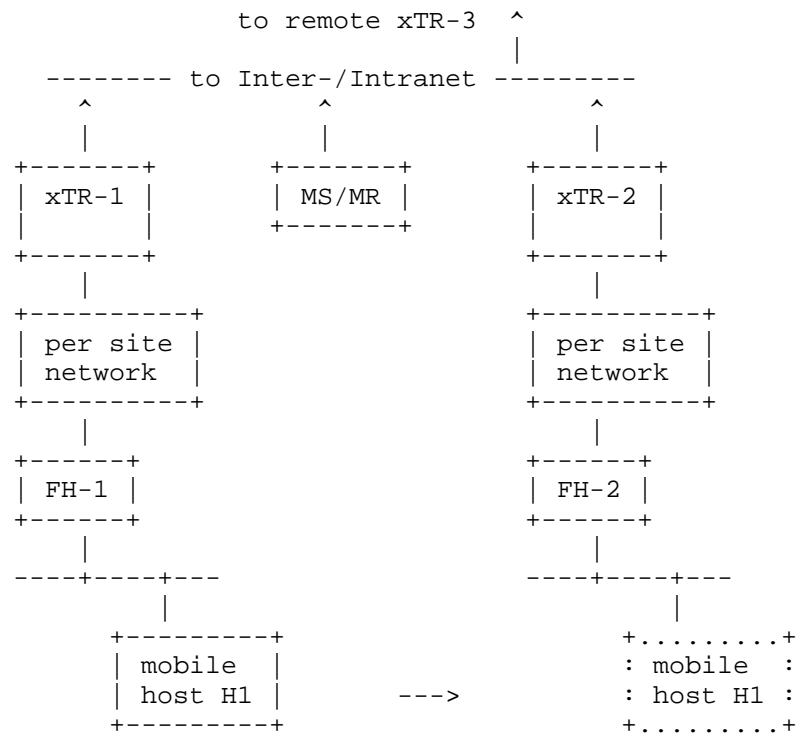


Figure 2: LISP Multihop Mobility

As a result the map server (MS) needs to be informed that the IP address of host H1 is reachable now via xTR-2. For this to happen it requires FH-2 to detect that host H1 is now connected to it's LAN. Then FH-2 needs to inform xTR-2, which in turn registers the IP address as an EID with it's own RLOC(s). The map server then will send a notification to xTR-1, which was holding the registration so far. xTR-1 in turn sends a notification to FH-1 to ensure the necessary state setup or cleanup happens fast. In theory one could combine detection and xTR functionality on the first hop nodes FH-1 and FH-2; these are the scenarios discussed in the earlier sections. Sometimes though a requirement exists for firewalls, IDS, load-balancers, WAN optimizers and other functionality in the "per site network" boxes in the figure above. Some of these services require access to the EID-space packet and may not have the ability to look into the LISP data packet. This is why detection and actual LISP encapsulation are separated for the following discussion.

4.2. LISP Use Cases for Multihop Mobility

The use cases are similar to the scenarios discussed in Section 3. Our focus will be on L3 though as the services implemented in the "per site network" are typically IP based:

- o End host IP mobility , where the end host subnet is not extended across the locations, where the EID is an IP address.
- o End host IP mobility, where the end host subnet is extended across the locations using some non-LISP L2 extension technique, where the EID is an IP address.

The difference between multihop and singlehop host mobility (as in Section 3) is mainly in the additional routing setup required in each site to match the host detection and LISP signaling. This will be discussed in the following sections.

4.2.1. End Host IP Mobility, non-extended subnet

The discussion of Section 3.2 applies for this case as well. The aspects covering ARP are handled by the First Hop routers while LISP registration and associated signaling is handled by the xTR. As a minimum the site xTR must register moved-in hosts to the mapping system, based on the detection on the FH router, as well as notify the xTR of the 'old' site of the host. This particular xTR will also generate SMR Messages to clear up stale state of remote LISP sites. This will attract traffic for the hosts that moved-in from the LISP network.

What needs to be added is the routing inside the sites. Assume host H1 has moved away from site-1 into site-2. For site-1 in this example two fundamental options exist: xTR-1 will be informed by the LISP Mapping System that host H1 has been detected elsewhere. This information can be used to inject a host route for H1 from xTR-1 into site-1 IGP. The FH-1 would inject the network prefix P1 into site-1's IGP. A detection of active hosts with an address within P1 would not be necessary as long as a detection of the return of absent hosts is guaranteed. xTR-1 would register the network prefix P1. For foreign hosts from site P2 the FH must detect them and inject a host route into site-1 IGP. The xTR of site-1 would register these foreign hosts. In other words the IGP would carry the network prefix P1 and hosts routes for local, foreign hosts and for the absent hosts belonging to network P1. The default routing for prefix P1 would point to the FH router. As an alternative all FH's could detect the active hosts on their LAN, both for hosts that are home at the site as well as hosts that moved into the site's network. The FH routers would inject a host route into the site IGP for every detected host.

The xTR injects the upper and lower half of network P into the IGP and would still register network P1 and the foreign hosts. The IGP would carry the two halves of prefix P and hosts routes for all local hosts. The default routing for prefix P would point to the xTR. Practically a mix of these two options is possible to optimize the number of routes, the number of registrations and/or the number of mapping requests.

The same options exist for site-2 and both sites can choose their internal routing scheme independently. This is possible as the registration scheme is the same: register the home network and the foreign hosts. In all cases a default route pointing to the xTR is completing the routing, to reach all other EID prefixes.

4.2.2. End Host IP Mobility, extended subnet

The discussion of Section 3.1 applies here as well, covering aspects of ARP handled by the First Hop routers and LISP registration, notification and SMRs handled by the xTRs.

For extended subnets it doesn't make sense to talk about the home site or a host returning home. All registrations with the LISP mapping system must be on a per-host basis, based on the locally detected hosts. Additional registration of the network prefix P would be necessary when the setup requires the xTRs of every site to know about all remote hosts. Such registrations of P would have the downside to attract traffic from the LISP network that finally may be dropped when no receiving host is found.

For the routing we can identify again two fundamental cases. The first case is that the FH, when detecting a mobile host, is not only informing the site's xTR but it also redistributing a host route into the site IGP. The xTR would redistribute the upper and lower half of the network prefix P into the site IGP to attract all addresses of P that are not detected in the site. A standard default route pointing to the xTR would complete the site routing. In summary the site IGP would carry host routes for all locally detected hosts and the default routing for prefix P would be towards the xTR.

The other case is that the xTR knows about all the remotely registered hosts and redistributes them as host routes into the site IGP. The FH router would inject a route for network P into the site IGP. A default route pointing to the xTR would complete the routing for the non-mobile EID prefixes. The IGP would carry the host routes of the remotely detected hosts and default routing for P would point to the FH.

Again sites can choose their internal routing independently as the common way to register all locally detected hosts guarantees interoperability of the site routings.

5. Protocol Extension Requirements

This section is not an exhaustive discussion nor description of the LISP protocol extensions used for the use cases. It only briefly mentions the requirements that allow for the design of the use cases in the previous sections.

- o Multiple registrations for the same prefix and parent registrations: registering a host and overriding the previous registration and then informing every site that needs to be updated is the core of how the mapping system synchronizes the sites. Imagine the extended subnet case with 2 sites and a host is activated at site-1. When site-2 uses a routing that requires knowledge of all remote hosts then site-2 needs to be updated. The proposed protocol extension requires site-2 to register the network P and the map server will notify every registerer of P when a host registration falls into the registered network P.
- o Reliable Notifications: notifications are a key aspect of how LISP mapping services updates the sites. A mechanism is proposed to acknowledge received notifications and retry sending them when the ACK is missing.
- o Separation of messages between xTR and FH against map server: an expected setup is to run both map resolver and map server on xTR routers. Messages from FH to xTR must be distinguishable from map registrations to avoid confusing the map server.

6. Acknowledgements

The authors want to thank Patrice Bellagamba, Johnson Leong, Victor Moreno and Fabio Maino for the early review, insightful comments and suggestions.

7. IANA Considerations

This memo includes no request to IANA.

8. Security Considerations

See [I-D.ietf-lisp-sec] for a list of security considerations for LISP.

9. References

9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

9.2. Informative References

- [I-D.farinacci-lisp-mr-signaling]
Farinacci, D. and M. Napierala, "LISP Control-Plane Multicast Signaling", draft-farinacci-lisp-mr-signaling-03 (work in progress), September 2013.
- [I-D.hertoghs-nvo3-lisp-controlplane-unified]
Hertoghs, Y., Maino, F., Moreno, V., Smith, M., Farinacci, D., and L. Iannone, "A Unified LISP Mapping Database for L2 and L3 Network Virtualization Overlays", draft-hertoghs-nvo3-lisp-controlplane-unified-01 (work in progress), February 2014.
- [I-D.ietf-lisp-sec]
Maino, F., Ermagan, V., Cabellos-Aparicio, A., Saucez, D., and O. Bonaventure, "LISP-Security (LISP-SEC)", draft-ietf-lisp-sec-05 (work in progress), October 2013.
- [I-D.maino-nvo3-lisp-cp]
Maino, F., Ermagan, V., Hertoghs, Y., Farinacci, D., and M. Smith, "LISP Control Plane for Network Virtualization Overlays", draft-maino-nvo3-lisp-cp-03 (work in progress), October 2013.
- [I-D.meyer-lisp-mn]
Farinacci, D., Lewis, D., Meyer, D., and C. White, "LISP Mobile Node", draft-meyer-lisp-mn-10 (work in progress), January 2014.
- [I-D.smith-lisp-layer2]
Smith, M., Dutt, D., Farinacci, D., and F. Maino, "Layer 2 (L2) LISP Encapsulation Format", draft-smith-lisp-layer2-03 (work in progress), September 2013.
- [RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, January 2013.

[RFC6831] Farinacci, D., Meyer, D., Zwiebel, J., and S. Venaas, "The Locator/ID Separation Protocol (LISP) for Multicast Environments", RFC 6831, January 2013.

[RFC6832] Lewis, D., Meyer, D., Farinacci, D., and V. Fuller, "Interworking between Locator/ID Separation Protocol (LISP) and Non-LISP Sites", RFC 6832, January 2013.

[draft-moreno-lisp-datacenter-deployment]
Moreno, V., "LISP Deployment Considerations in Data Center Networks.", Work in progress , 2014.

Authors' Addresses

Yves Hertoghs
Cisco Systems
De Kleetlaan 6a
Diegem 1831
BE

Email: yhertogh@cisco.com

Marc Binderberg
Cisco Systems
510 McCarthy Blvd.
Milpitas, California 95035
USA

Email: mbinderb@cisco.com

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: July 21, 2014

L. Jakab
Cisco Systems
A. Cabellos-Aparicio
F. Coras
J. Domingo-Pascual
Technical University of
Catalonia
D. Lewis
Cisco Systems
January 17, 2014

LISP Network Element Deployment Considerations
draft-ietf-lisp-deployment-12.txt

Abstract

This document is a snapshot of different Locator/Identifier Separation Protocol (LISP) deployment scenarios. It discusses the placement of new network elements introduced by the protocol, representing the thinking of the LISP working group as of Summer 2013. LISP deployment scenarios may have evolved since. This memo represents one stable point in that evolution of understanding.

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 <http://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 21, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Tunnel Routers	4
2.1. Deployment Scenarios	4
2.1.1. Customer Edge	4
2.1.2. Provider Edge	6
2.1.3. Tunnel Routers Behind NAT	7
2.1.3.1. ITR	7
2.1.3.2. ETR	8
2.1.3.3. Additional Notes	8
2.2. Functional Models with Tunnel Routers	8
2.2.1. Split ITR/ETR	8
2.2.2. Inter-Service Provider Traffic Engineering	10
2.3. Summary and Feature Matrix	12
3. Map Resolvers and Map Servers	13
3.1. Map Servers	13
3.2. Map Resolvers	15
4. Proxy Tunnel Routers	16
4.1. P-ITR	16
4.2. P-ETR	17
5. Migration to LISP	18
5.1. LISP+BGP	18
5.2. Mapping Service Provider (MSP) P-ITR Service	19
5.3. Proxy-ITR Route Distribution (PITR-RD)	19
5.4. Migration Summary	22
6. Security Considerations	22
7. IANA Considerations	23
8. Acknowledgements	23
9. References	23
9.1. Normative References	23
9.2. Informative References	23
Appendix A. Step-by-Step Example BGP to LISP Migration Procedure	24
A.1. Customer Pre-Install and Pre-Turn-up Checklist	24
A.2. Customer Activating LISP Service	26
A.3. Cut-Over Provider Preparation and Changes	27
Authors' Addresses	27

1. Introduction

The Locator/Identifier Separation Protocol (LISP) is designed to address the scaling issues of the global Internet routing system identified in [RFC4984] by separating the current addressing scheme into Endpoint IDentifiers (EIDs) and Routing LOCators (RLOCs). The main protocol specification [RFC6830] describes how the separation is achieved, which new network elements are introduced, and details the packet formats for the data and control planes.

LISP assumes that such separation is between the edge and core and uses mapping and encapsulation for forwarding. While the boundary between both is not strictly defined, one widely accepted definition places it at the border routers of stub autonomous systems, which may carry a partial or complete default-free zone (DFZ) routing table. The initial design of LISP took this location as a baseline for protocol development. However, the applications of LISP go beyond just decreasing the size of the DFZ routing table, and include improved multihoming and ingress traffic engineering (TE) support for edge networks, and even individual hosts. Throughout the document we will use the term LISP site to refer to these networks/hosts behind a LISP Tunnel Router. We formally define the following two terms:

Network element: Facility or equipment used in the provision of a communications service over the Internet [TELCO96].

LISP site: A single host or a set of network elements in an edge network under the administrative control of a single organization, delimited from other networks by LISP Tunnel Router(s).

Since LISP is a protocol which can be used for different purposes, it is important to identify possible deployment scenarios and the additional requirements they may impose on the protocol specification and other protocols. Additionally, this document is intended as a guide for the operational community for LISP deployments in their networks. It is expected to evolve as LISP deployment progresses, and the described scenarios are better understood or new scenarios are discovered.

Each subsection considers an element type, discussing the impact of deployment scenarios on the protocol specification. For definition of terms, please refer to the appropriate documents (as cited in the respective sections).

This experimental document describing deployment considerations and the LISP specifications have areas that require additional experience and measurement. LISP is not recommended for deployment beyond experimental situations. Results of experimentation may lead to

modifications and enhancements of the LISP protocol mechanisms. Additionally, at the time of this writing there is no standardized security to implement. Beware that there are no counter measures for any of the threads identified in [I-D.ietf-lisp-threats]. See Section 15 [of RFC 6830] for specific, known issues that are in need of further work during development, implementation, and experimentation, and [I-D.ietf-lisp-threats] for recommendations to ameliorate the above-mentioned security threats.

2. Tunnel Routers

The device that is the gateway between the edge and the core is called a Tunnel Router (xTR), performing one or both of two separate functions:

1. Encapsulating packets originating from an end host to be transported over intermediary (transit) networks towards the other end-point of the communication
2. Decapsulating packets entering from intermediary (transit) networks, originated at a remote end host.

The first function is performed by an Ingress Tunnel Router (ITR), the second by an Egress Tunnel Router (ETR).

Section 8 of the main LISP specification [RFC6830] has a short discussion of where Tunnel Routers can be deployed and some of the associated advantages and disadvantages. This section adds more detail to the scenarios presented there, and provides additional scenarios as well. Furthermore this section discusses functional models, that is, network functions that can be achieved by deploying Tunnel Routers in specific ways.

2.1. Deployment Scenarios

2.1.1. Customer Edge

The first scenario we discuss is customer edge, when xTR functionality is placed on the router(s) that connect the LISP site to its upstream(s), but are under its control. As such, this is the most common expected scenario for xTRs, and this document considers it the reference location, comparing the other scenarios to this one.

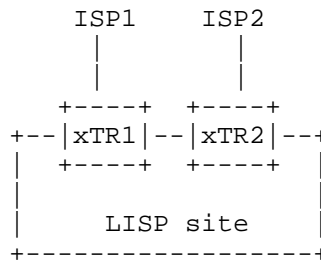


Figure 1: xTRs at the customer edge

From the LISP site perspective the main advantage of this type of deployment (compared to the one described in the next section) is having direct control over its ingress traffic engineering. This makes it easy to set up and maintain active/active, active/backup, or more complex TE policies, adding ISPs and additional xTRs at will, without involving third parties.

Being under the same administrative control, reachability information of all ETRs is easier to synchronize, because the necessary control traffic can be allowed between the locators of the ETRs. A correct synchronous global view of the reachability status is thus available, and the Locator Status Bits (Loc-Status-Bits, defined in [RFC6830]) can be set correctly in the LISP data header of outgoing packets.

By placing the tunnel router at the edge of the site, existing internal network configuration does not need to be modified. Firewall rules, router configurations and address assignments inside the LISP site remain unchanged. This helps with incremental deployment and allows a quick upgrade path to LISP. For larger sites with many external connections, distributed in geographically diverse points of presence (PoPs), and complex internal topology, it may however make more sense to both encapsulate and decapsulate as soon as possible, to benefit from the information in the IGP to choose the best path (see Section 2.2.1 for a discussion of this scenario).

Another thing to consider when placing tunnel routers is MTU issues. Encapsulation increases the amount of overhead associated with each packet. This added overhead decreases the effective end-to-end path MTU (unless fragmentation and reassembly is used). Some transit networks are known to provide larger MTU than the typical value of 1500 bytes of popular access technologies used at end hosts (e.g., IEEE 802.3 and 802.11). However, placing the LISP router connecting to such a network at the customer edge could possibly bring up MTU issues, depending on the link type to the provider as opposed to the following scenario. See [RFC4459] for MTU considerations of tunneling protocols on how to mitigate potential issues. Still, even

with these mitigations, path MTU issues are still possible.

2.1.1.2. Provider Edge

The other location at the core-edge boundary for deploying LISP routers is at the Internet service provider edge. The main incentive for this case is that the customer does not have to upgrade the CE router(s), or change the configuration of any equipment. Encapsulation/decapsulation happens in the provider's network, which may be able to serve several customers with a single device. For large ISPs with many residential/business customers asking for LISP this can lead to important savings, since there is no need to upgrade the software (or hardware, if it's the case) at each client's location. Instead, they can upgrade the software (or hardware) on a few PE routers serving the customers. This scenario is depicted in Figure 2.

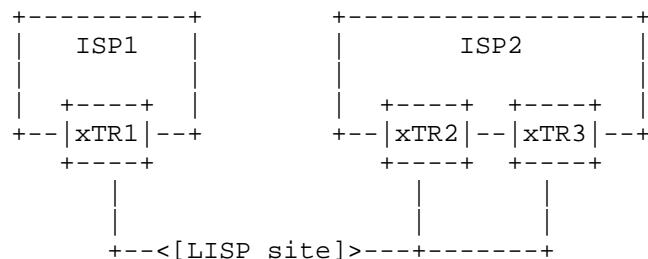


Figure 2: xTR at the PE

While this approach can make transition easy for customers and may be cheaper for providers, the LISP site loses one of the main benefits of LISP: ingress traffic engineering. Since the provider controls the ETRs, additional complexity would be needed to allow customers to modify their mapping entries.

The problem is aggravated when the LISP site is multihomed. Consider the scenario in Figure 2: whenever a change to TE policies is required, the customer contacts both ISP1 and ISP2 to make the necessary changes on the routers (if they provide this possibility). It is however unlikely, that both ISPs will apply changes simultaneously, which may lead to inconsistent state for the mappings of the LISP site. Since the different upstream ISPs are usually competing business entities, the ETRs may even be configured to compete, either to attract all the traffic or to get no traffic. The former will happen if the customer pays per volume, the latter if the connectivity has a fixed price. A solution could be to configure the Map Server(s) to do proxy-replying and have the Mapping Service Provider (MSP) apply policies.

Additionally, since xTR1, xTR2, and xTR3 are in different administrative domains, locator reachability information is unlikely to be exchanged among them, making it difficult to set Loc-Status-Bits (LSB) correctly on encapsulated packets. Because of this, and due to the security concerns about LSB described in [I-D.ietf-lisp-threats] their use is discouraged (set the L-bit to 0). Mapping versioning is another alternative [RFC6834].

Compared to the customer edge scenario, deploying LISP at the provider edge might have the advantage of diminishing potential MTU issues, because the tunnel router is closer to the core, where links typically have higher MTUs than edge network links.

2.1.3. Tunnel Routers Behind NAT

NAT in this section refers to IPv4 network address and port translation.

2.1.3.1. ITR

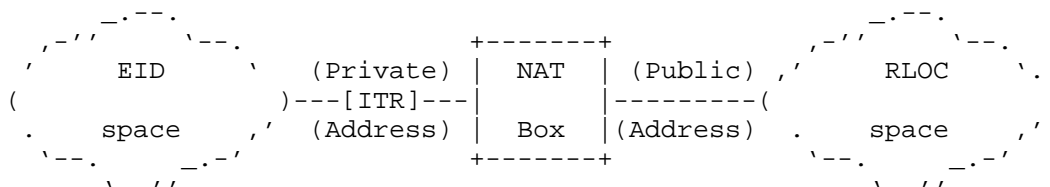


Figure 3: ITR behind NAT

Packets encapsulated by an ITR are just UDP packets from a NAT device's point of view, and they are handled like any UDP packet, there are no additional requirements for LISP data packets.

Map-Requests sent by an ITR, which create the state in the NAT table, have a different 5-tuple in the IP header than the Map-Reply generated by the authoritative ETR. Since the source address of this packet is different from the destination address of the request packet, no state will be matched in the NAT table and the packet will be dropped. To avoid this, the NAT device has to do the following:

- o Send all UDP packets with source port 4342, regardless of the destination port, to the RLOC of the ITR. The most simple way to achieve this is configuring 1:1 NAT mode from the external RLOC of the NAT device to the ITR's RLOC (Called "DMZ" mode in consumer broadband routers).

- o Rewrite the ITR-AFI and "Originating ITR RLOC Address" fields in the payload.

This setup supports only a single ITR behind the NAT device.

2.1.3.2. ETR

An ETR placed behind NAT is reachable from the outside by the Internet-facing locator of the NAT device. It needs to know this locator (and configure a loopback interface with it), so that it can use it in Map-Reply and Map-Register messages. Thus support for dynamic locators for the mapping database is needed in LISP equipment.

Again, only one ETR behind the NAT device is supported.

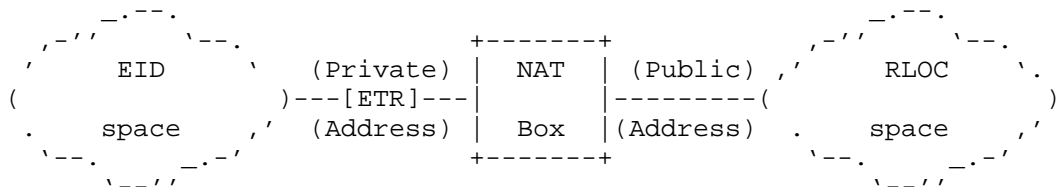


Figure 4: ETR behind NAT

2.1.3.3. Additional Notes

An implication of the issues described above is that LISP sites with xTRs can not be behind carrier based NATs, since two different sites would collide on the port forwarding. An alternative to static hole-punching to explore is the use of the Port Control Protocol (PCP) [RFC6887].

We only include this scenario due to completeness, to show that a LISP site can be deployed behind NAT, should it become necessary. However, LISP deployments behind NAT should be avoided, if possible.

2.2. Functional Models with Tunnel Routers

This section describes how certain LISP deployments can provide network functions.

2.2.1. Split ITR/ETR

In a simple LISP deployment, xTRs are located at the border of the LISP site (see Section 2.1.1). In this scenario packets are routed inside the domain according to the EID. However, more complex

networks may want to route packets according to the destination RLOC. This would enable them to choose the best egress point.

The LISP specification separates the ITR and ETR functionality and allows both entities to be deployed in separated network equipment. ITRs can be deployed closer to the host (i.e., access routers). This way packets are encapsulated as soon as possible, and egress point selection is driven by operational policy. In turn, ETRs can be deployed at the border routers of the network, and packets are decapsulated as soon as possible. Once decapsulated, packets are routed based on destination EID, according to internal routing policy.

In the following figure we can see an example. The Source (S) transmits packets using its EID and in this particular case packets are encapsulated at ITR_1. The encapsulated packets are routed inside the domain according to the destination RLOC, and can egress the network through the best point (i.e., closer to the RLOC's AS). On the other hand, inbound packets are received by ETR_1 which decapsulates them. Then packets are routed towards S according to the EID, again following the best path.

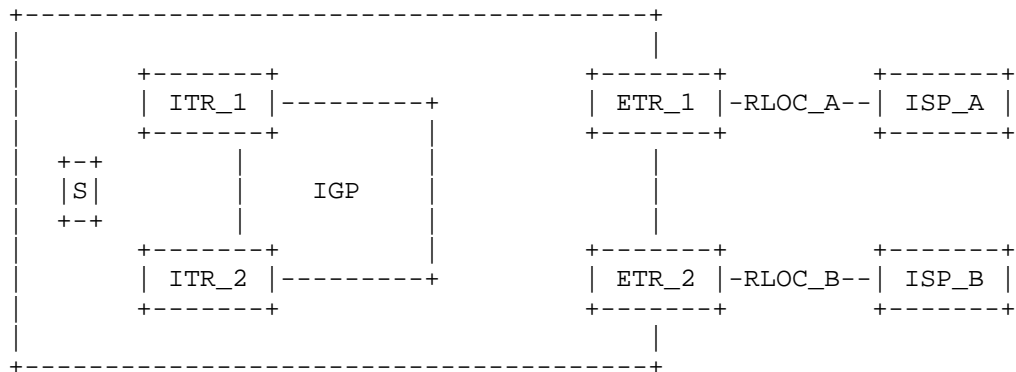


Figure 5: Split ITR/ETR Scenario

This scenario has a set of implications:

- o The site must carry more specific routes in order to choose the best egress point, and typically BGP is used for this, increasing the complexity of the network. However, this is usually already the case for LISP sites that would benefit from this scenario.
- o If the site is multihomed to different ISPs and any of the upstream ISPs are doing uRPF filtering, this scenario may become impractical. ITRs need to determine the exit ETR, for setting the

correct source RLOC in the encapsulation header. This adds complexity and reliability concerns.

- o In LISP, ITRs set the reachability bits when encapsulating data packets. Hence, ITRs need a mechanism to be aware of the liveness of all ETRs serving their site.
- o MTU within the site network must be large enough to accommodate encapsulated packets.
- o In this scenario, each ITR is serving fewer hosts than in the case when it is deployed at the border of the network. It has been shown that cache hit ratio grows logarithmically with the amount of users [CACHE]. Taking this into account, when ITRs are deployed closer to the host the effectiveness of the mapping cache may be lower (i.e., the miss ratio is higher). Another consequence of this is that the site may transmit a higher amount of Map-Requests, increasing the load on the distributed mapping database.
- o By placing the ITRs inside the site, they will still need global RLOCs, and this may add complexity to intra-site routing configuration, and further intra-site issues when there is a change of providers.

2.2.2. Inter-Service Provider Traffic Engineering

At the time of this writing, if two ISPs want to control their ingress TE policies for transit traffic between them, they need to rely on existing BGP mechanisms. This typically means deaggregating prefixes to choose on which upstream link packets should enter. This is either not feasible (if fine-grained per-customer control is required, the very specific prefixes may not be propagated) or increases DFZ table size.

Typically, LISP is seen applicable only to stub networks, however the LISP protocol can be also applied in a recursive manner, providing service provider ingress/egress TE capabilities without impacting the DFZ table size.

In order to implement this functionality with LISP consider the scenario depicted in Figure 6. The two ISPs willing to achieve ingress/egress TE are labeled as ISP_A and ISP_B, they are servicing Stub1 and Stub2 respectively, both are required to be LISP sites with their own xTRs. In this scenario we assume that Stub1 and Stub2 are communicating with each other and thus, ISP_A and ISP_B offer transit for such communications. ISP_A has RLOC_A1 and RLOC_A2 as upstream IP addresses while ISP_B has RLOC_B1 and RLOC_B2. The shared goal

among ISP_A and ISP_B is to control the transit traffic flow between RLOC_A1/A2 and RLOC_B1/B2.

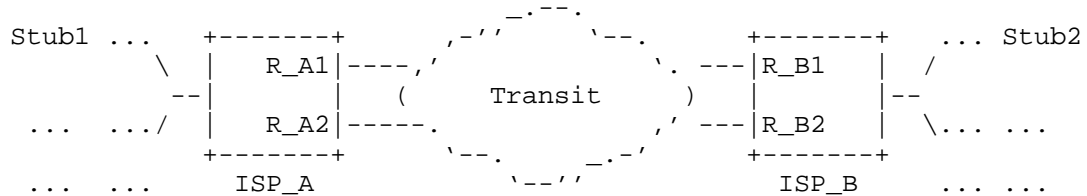


Figure 6: Inter-Service provider TE scenario

Both ISPs deploy xTRs on RLOC_A1/A2 and RLOC_B1/B2 respectively and reach a bilateral agreement to deploy their own private mapping system. This mapping system contains bindings between the RLOCs of Stub1 and Stub2 (owned by ISP_A and ISP_B respectively) and RLOC_A1/A2 and RLOC_B1/B2. Such bindings are in fact the TE policies between both ISPs and the convergence time is expected to be fast, since ISPs only have to update/query a mapping to/from the database.

The packet flow is as follows. First, a packet originated at Stub1 towards Stub2 is LISP encapsulated by Stub1's xTR. The xTR of ISP_A recursively encapsulates it and, according to the TE policies stored in the private mapping system, the ISP_A xTR chooses RLOC_B1 or RLOC_B2 as the outer encapsulation destination. Note that the packet transits between ISP_A and ISP_B double-encapsulated. Upon reception at the xTR of ISP_B the packet is decapsulated and sent towards Stub2 which performs the last decapsulation.

This deployment scenario, which uses recursive LISP, includes three important caveats. First, it is intended to be deployed between only two ISPs. If more than two ISPs use this approach, then the xTRs deployed at the participating ISPs must either query multiple mapping systems, or the ISPs must agree on a common shared mapping system. Furthermore, keeping this deployment scenario restricted to only two ISPs maintains the solution scalable, given that only two entities need to agree on using recursive LISP, and only one private mapping system is involved.

Second, the scenario is only recommended for ISPs providing connectivity to LISP sites, such that source RLOCs of packets to be recursively encapsulated belong to said ISP. Otherwise the participating ISPs must register prefixes they do not own in the above mentioned private mapping system. This results in either requiring complex authentication mechanisms or enabling simple traffic redirection attacks. Failure to follow these recommendations may lead to operational security issues when deploying this scenario.

And third, recursive encapsulation models are typically complex to troubleshoot and debug.

Besides these recommendations, the main disadvantages of this deployment case are:

- o Extra LISP header is needed. This increases the packet size and requires that the MTU between both ISPs accommodates double-encapsulated packets.
- o The ISP ITR must encapsulate packets and therefore must know the RLOC-to-RLOC binding. These bindings are stored in a mapping database and may be cached in the ITR's mapping cache. Cache misses lead to an additional lookup latency, unless a push based mapping system is used for the private mapping system.
- o The operational overhead of maintaining the shared mapping database.

2.3. Summary and Feature Matrix

When looking at the deployment scenarios and functional models above, there are several things to consider when choosing the appropriate one, depending on the type of the organization doing the deployment.

For home users and small site who wish to multihome and have control over their ISP options, the "CE" scenario offers the most advantages: it's simple to deploy, in some cases it only requires a software upgrade of the CPE, getting mapping service, and configuring the router. It retains control of TE and choosing upstreams by the user. It doesn't provide too many advantages to ISPs, due to the lessened dependence on their services in case of multihomed clients. It is also unlikely that ISP wishing to offer LISP to their customers will choose the "CE" placement: they need to send a technician to each customer, and potentially a new CPE. Even if they have remote control over the router, and a software upgrade could add LISP support, the operation is too risky.

For a network operator a good option to deploy is the "PE" scenario, unless a hardware upgrade is required for its edge routers to support LISP (in which case upgrading CPEs may be simpler). It retains control of TE, choice of PETR, and MS/MR. It also lowers potential MTU issues, as discussed above. Network operators should also explore the "Inter-SP TE" (recursive) functional model for their TE needs.

Large organizations can benefit the most from the "Split ITR/ETR" functional model, to optimize their traffic flow.

The following table gives a quick overview of the features supported by each of the deployment scenarios discussed above (marked with an "x") in the appropriate column: "CE" for customer edge, "PE" for provider edge, "Split" for split ITR/ETR, and "Recursive" for inter-service provider traffic engineering. The discussed features include:

Control of ingress TE: The scenario allows the LISP site to easily control LISP ingress traffic engineering policies.

No modifications to existing int. network infrastructure: The scenario doesn't require the LISP site to modify internal network configurations.

Loc-Status-Bits sync: The scenario allows easy synchronization of the Locator Status Bits.

MTU/PMTUD issues minimized: The scenario minimizes potential MTU and Path MTU Discovery issues.

Feature	CE	PE	Split	Recursive	NAT
Control of ingress TE	x	-	x	x	x
No modifications to existing int. network infrastructure	x	x	-	-	x
Loc-Status-Bits sync	x	-	x	x	-
MTU/PMTUD issues minimized	-	x	-	-	-

3. Map Resolvers and Map Servers

Map Resolvers and Map Servers make up the LISP mapping system and provide a means to find authoritative EID-to-RLOC mapping information, conforming to [RFC6833]. They are meant to be deployed in RLOC space, and their operation behind NAT is not supported.

3.1. Map Servers

The Map Server learns EID-to-RLOC mapping entries from an authoritative source and publishes them in the distributed mapping database. These entries are learned through authenticated Map-Register messages sent by authoritative ETRs. Also, upon reception of a Map-Request, the Map Server verifies that the destination EID matches an EID-prefix for which it is authoritative for, and then re-encapsulates and forwards it to a matching ETR. Map Server functionality is described in detail in [RFC6833].

The Map Server is provided by a Mapping Service Provider (MSP). The MSP participates in the global distributed mapping database infrastructure, by setting up connections to other participants, according to the specific mapping system that is employed (e.g., ALT [RFC6836], DDT [I-D.ietf-lisp-ddt]). Participation in the mapping database, and the storing of EID-to-RLOC mapping data is subject to the policies of the "root" operators, who should check ownership rights for the EID prefixes stored in the database by participants. These policies are out of the scope of this document.

The LISP DDT protocol is used by LISP Mapping Service providers to provide reachability between those providers' Map-Resolvers and Map-Servers. The DDT Root is currently operated by a collection of organizations on an open basis. See [DDT-ROOT] for more details. Similarly to the DNS root, it has several different server instances using names of the letters of the Greek alphabet (alpha, delta, etc.), operated by independent organizations. When this document was published, there were 5 such instances, one of them being anycasted. The Root provides the list of server instances on their web site and configuration files for several map server implementations. The DDT Root, and LISP Mapping Providers both rely on and abide by existing allocation policies by Regional Internet Registries to determine prefix ownership for use as EIDs.

It is expected that the DDT root organizations will continue to evolve in response to experimentation with LISP deployments for Internet edge multi-homing and VPN use cases.

In all cases, the MSP configures its Map Server(s) to publish the prefixes of its clients in the distributed mapping database and start encapsulating and forwarding Map-Requests to the ETRs of the AS. These ETRs register their prefix(es) with the Map Server(s) through periodic authenticated Map-Register messages. In this context, for some LISP sites, there is a need for mechanisms to:

- o Automatically distribute EID prefix(es) shared keys between the ETRs and the EID-registrar Map Server.
- o Dynamically obtain the address of the Map Server in the ETR of the AS.

The Map Server plays a key role in the reachability of the EID-prefixes it is serving. On the one hand it is publishing these prefixes into the distributed mapping database and on the other hand it is encapsulating and forwarding Map-Requests to the authoritative ETRs of these prefixes. ITRs encapsulating towards EIDs under the responsibility of a failed Map Server will be unable to look up any of their covering prefixes. The only exception are the ITRs that

already contain the mappings in their local cache. In this case ITRs can reach ETRs until the entry expires (typically 24 hours). For this reason, redundant Map Server deployments are desirable. A set of Map Servers providing high-availability service to the same set of prefixes is called a redundancy group. ETRs are configured to send Map-Register messages to all Map Servers in the redundancy group. The configuration for fail-over (or load-balancing, if desired) among the members of the group depends on the technology behind the mapping system being deployed. Since ALT is based on BGP and DDT was inspired from the Domain Name System (DNS), deployments can leverage current industry best practices for redundancy in BGP and DNS. These best practices are out of the scope of this document.

Additionally, if a Map Server has no reachability for any ETR serving a given EID block, it should not originate that block into the mapping system.

3.2. Map Resolvers

A Map Resolver is a network infrastructure component which accepts LISP encapsulated Map-Requests, typically from an ITR, and finds the appropriate EID-to-RLOC mapping by consulting the distributed mapping database. Map Resolver functionality is described in detail in [RFC6833].

Anyone with access to the distributed mapping database can set up a Map Resolver and provide EID-to-RLOC mapping lookup service. Database access setup is mapping system specific.

For performance reasons, it is recommended that LISP sites use Map Resolvers that are topologically close to their ITRs. ISPs supporting LISP will provide this service to their customers, possibly restricting access to their user base. LISP sites not in this position can use open access Map Resolvers, if available. However, regardless of the availability of open access resolvers, the MSP providing the Map Server(s) for a LISP site should also make available Map Resolver(s) for the use of that site.

In medium to large-size ASes, ITRs must be configured with the RLOC of a Map Resolver, operation which can be done manually. However, in Small Office Home Office (SOHO) scenarios a mechanism for autoconfiguration should be provided.

One solution to avoid manual configuration in LISP sites of any size is the use of anycast RLOCs [RFC4786] for Map Resolvers similar to the DNS root server infrastructure. Since LISP uses UDP encapsulation, the use of anycast would not affect reliability. LISP routers are then shipped with a preconfigured list of well know Map

Resolver RLOCs, which can be edited by the network administrator, if needed.

The use of anycast also helps improve mapping lookup performance. Large MSPs can increase the number and geographical diversity of their Map Resolver infrastructure, using a single anycasted RLOC. Once LISP deployment is advanced enough, very large content providers may also be interested running this kind of setup, to ensure minimal connection setup latency for those connecting to their network from LISP sites.

While Map Servers and Map Resolvers implement different functionalities within the LISP mapping system, they can coexist on the same device. For example, MSPs offering both services, can deploy a single Map Resolver/Map Server in each PoP where they have a presence.

4. Proxy Tunnel Routers

4.1. P-ITR

Proxy Ingress Tunnel Routers (P-ITRs) are part of the non-LISP/LISP transition mechanism, allowing non-LISP sites to reach LISP sites. They announce via BGP certain EID prefixes (aggregated, whenever possible) to attract traffic from non-LISP sites towards EIDs in the covered range. They do the mapping system lookup, and encapsulate received packets towards the appropriate ETR. Note that for the reverse path LISP sites can reach non-LISP sites simply by not encapsulating traffic. See [RFC6832] for a detailed description of P-ITR functionality.

The success of new protocols depends greatly on their ability to maintain backwards compatibility and inter-operate with the protocol(s) they intend to enhance or replace, and on the incentives to deploy the necessary new software or equipment. A LISP site needs an interworking mechanism to be reachable from non-LISP sites. A P-ITR can fulfill this role, enabling early adopters to see the benefits of LISP, similar to tunnel brokers helping the transition from IPv4 to IPv6. A site benefits from new LISP functionality (proportionally with existing global LISP deployment) when going LISP, so it has the incentives to deploy the necessary tunnel routers. In order to be reachable from non-LISP sites it has two options: keep announcing its prefix(es) with BGP, or have a P-ITR announce prefix(es) covering them.

If the goal of reducing the DFZ routing table size is to be reached, the second option is preferred. Moreover, the second option allows

LISP-based ingress traffic engineering from all sites. However, the placement of P-ITRs significantly influences performance and deployment incentives. Section 5 is dedicated to the migration to a LISP-enabled Internet, and includes deployment scenarios for P-ITRs.

4.2. P-ETR

In contrast to P-ITRs, P-ETRs are not required for the correct functioning of all LISP sites. There are two cases, where they can be of great help:

- o LISP sites with unicast reverse path forwarding (uRPF) restrictions, and
- o Communication between sites using different address family RLOCs.

In the first case, uRPF filtering is applied at their upstream PE router. When forwarding traffic to non-LISP sites, an ITR does not encapsulate packets, leaving the original IP headers intact. As a result, packets will have EIDs in their source address. Since we are discussing the transition period, we can assume that a prefix covering the EIDs belonging to the LISP site is advertised to the global routing tables by a P-ITR, and the PE router has a route towards it. However, the next hop will not be on the interface towards the CE router, so non-encapsulated packets will fail uRPF checks.

To avoid this filtering, the affected ITR encapsulates packets towards the locator of the P-ETR for non-LISP destinations. Now the source address of the packets, as seen by the PE router is the ITR's locator, which will not fail the uRPF check. The P-ETR then decapsulates and forwards the packets.

The second use case is IPv4-to-IPv6 transition. Service providers using older access network hardware, which only supports IPv4 can still offer IPv6 to their clients, by providing a CPE device running LISP, and P-ETR(s) for accessing IPv6-only non-LISP sites and LISP sites, with IPv6-only locators. Packets originating from the client LISP site for these destinations would be encapsulated towards the P-ETR's IPv4 locator. The P-ETR is in a native IPv6 network, decapsulating and forwarding packets. For non-LISP destination, the packet travels natively from the P-ETR. For LISP destinations with IPv6-only locators, the packet will go through a P-ITR, in order to reach its destination.

For more details on P-ETRs see [RFC6832].

P-ETRs can be deployed by ISPs wishing to offer value-added services

to their customers. As is the case with P-ITRs, P-ETRs too may introduce path stretch (the ratio between the cost of the selected path and that of the optimal path). Because of this the ISP needs to consider the tradeoff of using several devices, close to the customers, to minimize it, or few devices, farther away from the customers, minimizing cost instead.

Since the deployment incentives for P-ITRs and P-ETRs are different, it is likely they will be deployed in separate devices, except for the CDN case, which may deploy both in a single device.

In all cases, the existence of a P-ETR involves another step in the configuration of a LISP router. CPE routers, which are typically configured by DHCP, stand to benefit most from P-ETRs. Autoconfiguration of the P-ETR locator could be achieved by a DHCP option, or adding a P-ETR field to either Map-Notifys or Map-Replies.

5. Migration to LISP

This section discusses a deployment architecture to support the migration to a LISP-enabled Internet. The loosely defined terms of "early transition phase", "late transition phase", and "LISP Internet phase" refer to time periods when LISP sites are a minority, a majority, or represent all edge networks respectively.

5.1. LISP+BGP

For sites wishing to go LISP with their PI prefix the least disruptive way is to upgrade their border routers to support LISP, register the prefix into the LISP mapping system, but keep announcing it with BGP as well. This way LISP sites will reach them over LISP, while legacy sites will be unaffected by the change. The main disadvantage of this approach is that no decrease in the DFZ routing table size is achieved. Still, just increasing the number of LISP sites is an important gain, as an increasing LISP/non-LISP site ratio may decrease the need for BGP-based traffic engineering that leads to prefix deaggregation. That, in turn, may lead to a decrease in the DFZ size and churn in the late transition phase.

This scenario is not limited to sites that already have their prefixes announced with BGP. Newly allocated EID blocks could follow this strategy as well during the early LISP deployment phase, depending on the cost/benefit analysis of the individual networks. Since this leads to an increase in the DFZ size, the following architecture should be preferred for new allocations.

5.2. Mapping Service Provider (MSP) P-ITR Service

In addition to publishing their clients' registered prefixes in the mapping system, MSPs with enough transit capacity can offer them P-ITR service as a separate service. This service is especially useful for new PI allocations, to sites without existing BGP infrastructure, that wish to avoid BGP altogether. The MSP announces the prefix into the DFZ, and the client benefits from ingress traffic engineering without prefix deaggregation. The downside of this scenario is adding path stretch.

Routing all non-LISP ingress traffic through a third party which is not one of its ISPs is only feasible for sites with modest amounts of traffic (like those using the IPv6 tunnel broker services today), especially in the first stage of the transition to LISP, with a significant number of legacy sites. This is because the handling of said traffic is likely to result in additional costs, which would be passed down to the client. When the LISP/non-LISP site ratio becomes high enough, this approach can prove increasingly attractive.

Compared to LISP+BGP, this approach avoids DFZ bloat caused by prefix deaggregation for traffic engineering purposes, resulting in slower routing table increase in the case of new allocations and potential decrease for existing ones. Moreover, MSPs serving different clients with adjacent aggregatable prefixes may lead to additional decrease, but quantifying this decrease is subject to future research study.

5.3. Proxy-ITR Route Distribution (PITR-RD)

Instead of a LISP site, or the MSP, announcing their EIDs with BGP to the DFZ, this function can be outsourced to a third party, a P-ITR Service Provider (PSP). This will result in a decrease of the operational complexity both at the site and at the MSP.

The PSP manages a set of distributed P-ITR(s) that will advertise the corresponding EID prefixes through BGP to the DFZ. These P-ITR(s) will then encapsulate the traffic they receive for those EIDs towards the RLOCs of the LISP site, ensuring their reachability from non-LISP sites.

While it is possible for a PSP to manually configure each client's EID routes to be announced, this approach offers little flexibility and is not scalable. This section presents a scalable architecture that offers automatic distribution of EID routes to LISP sites and service providers.

The architecture requires no modification to existing LISP network elements, but it introduces a new (conceptual) network element, the

EID Route Server, defined as a router that either propagates routes learned from other EID Route Servers, or it originates EID Routes. The EID-Routes that it originates are those that it is authoritative for. It propagates these routes to Proxy-ITRs within the AS of the EID Route Server. It is worth to note that a BGP capable router can be also considered as an EID Route Server.

Further, an EID-Route is defined as a prefix originated via the Route Server of the mapping service provider, which should be aggregated if the MSP has multiple customers inside a single large continuous prefix. This prefix is propagated to other P-ITRs both within the MSP and to other P-ITR operators it peers with. EID Route Servers are operated either by the LISP site, MSPs or PSPs, and they may be collocated with a Map Server or P-ITR, but are a functionally discrete entity. They distribute EID-Routes, using BGP, to other domains, according to policies set by participants.

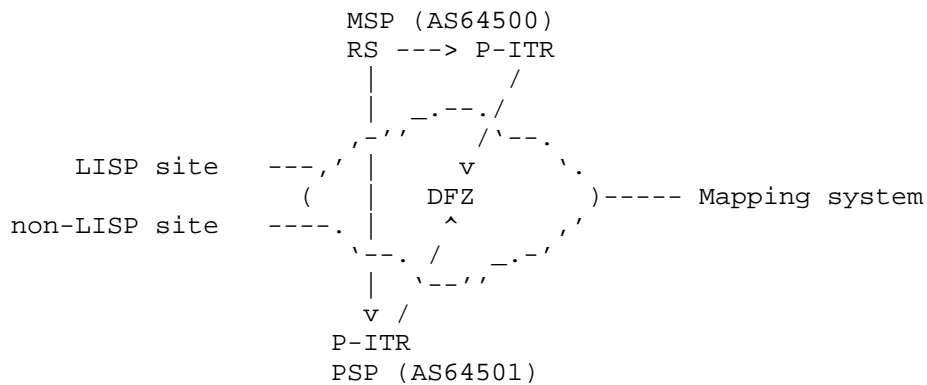


Figure 7: The P-ITR Route Distribution architecture

The architecture described above decouples EID origination from route propagation, with the following benefits:

- o Can accurately represent business relationships between P-ITR operators
- o More mapping system agnostic
- o Minor changes to P-ITR implementation, no changes to other components

In the example in the figure we have a MSP providing services to the LISP site. The LISP site does not run BGP, and gets an EID allocation directly from a RIR, or from the MSP, who may be a LIR. Existing PI allocations can be migrated as well. The MSP ensures the

presence of the prefix in the mapping system, and runs an EID Route Server to distribute it to P-ITR service providers. Since the LISP site does not run BGP, the prefix will be originated with the AS number of the MSP.

In the simple case depicted in Figure 7 the EID-Route of LISP site will be originated by the Route Server, and announced to the DFZ by the PSP's P-ITRs with AS path 64501 64500. From that point on, the usual BGP dynamics apply. This way, routes announced by P-ITR are still originated by the authoritative Route Server. Note that the peering relationships between MSP/PSPs and those in the underlying forwarding plane may not be congruent, making the AS path to a P-ITR shorter than it is in reality.

The non-LISP site will select the best path towards the EID-prefix, according to its local BGP policies. Since AS-path length is usually an important metric for selecting paths, a careful placement of P-ITR could significantly reduce path-stretch between LISP and non-LISP sites.

The architecture allows for flexible policies between MSP/PSPs. Consider the EID Route Server networks as control plane overlays, facilitating the implementation of policies necessary to reflect the business relationships between participants. The results are then injected to the common underlying forwarding plane. For example, some MSP/PSPs may agree to exchange EID-Prefixes and only announce them to each of their forwarding plane customers. Global reachability of an EID-prefix depends on the MSP the LISP site buys service from, and is also subject to agreement between the mentioned parties.

In terms of impact on the DFZ, this architecture results in a slower routing table increase for new allocations, since traffic engineering will be done at the LISP level. For existing allocations migrating to LISP, the DFZ may decrease since MSPs may be able to aggregate the prefixes announced.

Compared to LISP+BGP, this approach avoids DFZ bloat caused by prefix deaggregation for traffic engineering purposes, resulting in slower routing table increase in the case of new allocations and potential decrease for existing ones. Moreover, MSPs serving different clients with adjacent aggregatable prefixes may lead to additional decrease, but quantifying this decrease is subject to future research study.

The flexibility and scalability of this architecture does not come without a cost however: A PSP operator has to establish either transit or peering relationships to improve their connectivity.

5.4. Migration Summary

Registering a domain name typically entails an annual fee that should cover the operating expenses for publishing the domain in the global DNS. The situation is similar with several other registration services. A LISP mapping service provider (MSR) client publishing an EID prefix in the LISP mapping system has the option of signing up for P-ITR services as well, for an extra fee. These services may be offered by the MSP itself, but it is expected that specialized P-ITR service providers (PSPs) will do it. Clients not signing up become responsible for getting non-LISP traffic to their EIDs (using the LISP+BGP scenario).

Additionally, Tier 1 ISPs have incentives to offer P-ITR services to non-subscribers in strategic places just to attract more traffic from competitors, thus more revenue.

The following table presents the expected effects of the different transition scenarios during a certain phase on the DFZ routing table size:

Phase	LISP+BGP	MSP P-ITR	P-ITR-RD
Early transition	no change	slower increase	slower increase
Late transition	may decrease	slower increase	slower increase
LISP Internet	considerable decrease		

It is expected that P-ITR-RD will co-exist with LISP+BGP during the migration, with the latter being more popular in the early transition phase. As the transition progresses and the MSP P-ITR and P-ITR-RD ecosystem gets more ubiquitous, LISP+BGP should become less attractive, slowing down the increase of the number of routes in the DFZ.

Note that throughout Section 5 we focused on the effects of LISP deployment on the DFZ route table size. Other metrics may be impacted as well, but to the best of our knowledge have not been measured as of yet.

6. Security Considerations

All security implications of LISP deployments are to be discussed in separate documents. [I-D.ietf-lisp-threats] gives an overview of LISP threat models, including ETR operators attracting traffic by overclaiming an EID-prefix (Section 4.4.3). Securing mapping lookups is discussed in [I-D.ietf-lisp-sec].

7. IANA Considerations

This memo includes no request to IANA.

8. Acknowledgements

Many thanks to Margaret Wasserman for her contribution to the IETF76 presentation that kickstarted this work. The authors would also like to thank Damien Saucez, Luigi Iannone, Joel Halpern, Vince Fuller, Dino Farinacci, Terry Manderson, Noel Chiappa, Hannu Flinck, Paul Vinciguerra, Fred Templin, Brian Haberman, and everyone else who provided input.

9. References

9.1. Normative References

- [RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, January 2013.
- [RFC6832] Lewis, D., Meyer, D., Farinacci, D., and V. Fuller, "Interworking between Locator/ID Separation Protocol (LISP) and Non-LISP Sites", RFC 6832, January 2013.
- [RFC6833] Fuller, V. and D. Farinacci, "Locator/ID Separation Protocol (LISP) Map-Server Interface", RFC 6833, January 2013.

9.2. Informative References

- [CACHE] Jung, J., Sit, E., Balakrishnan, H., and R. Morris, "DNS performance and the effectiveness of caching", 2002.
- [DDT-ROOT]
"DDT Root", <<http://ddt-root.org/>>.
- [I-D.ietf-lisp-ddt]
Fuller, V., Lewis, D., Ermagan, V., and A. Jain, "LISP Delegated Database Tree", draft-ietf-lisp-ddt-01 (work in progress), March 2013.
- [I-D.ietf-lisp-sec]
Maino, F., Ermagan, V., Cabellos-Aparicio, A., Saucez, D., and O. Bonaventure, "LISP-Security (LISP-SEC)", draft-ietf-lisp-sec-05 (work in progress), October 2013.

- [I-D.ietf-lisp-threats]
Saucez, D., Iannone, L., and O. Bonaventure, "LISP Threats Analysis", draft-ietf-lisp-threats-08 (work in progress), October 2013.
- [RFC4459] Savola, P., "MTU and Fragmentation Issues with In-the-Network Tunneling", RFC 4459, April 2006.
- [RFC4786] Abley, J. and K. Lindqvist, "Operation of Anycast Services", BCP 126, RFC 4786, December 2006.
- [RFC4984] Meyer, D., Zhang, L., and K. Fall, "Report from the IAB Workshop on Routing and Addressing", RFC 4984, September 2007.
- [RFC6834] Iannone, L., Saucez, D., and O. Bonaventure, "Locator/ID Separation Protocol (LISP) Map-Versioning", RFC 6834, January 2013.
- [RFC6836] Fuller, V., Farinacci, D., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol Alternative Logical Topology (LISP+ALT)", RFC 6836, January 2013.
- [RFC6887] Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, April 2013.
- [TELC096] "Telecommunications Act of 1996", 1996.

Appendix A. Step-by-Step Example BGP to LISP Migration Procedure

To help the operational community deploy LISP, this informative section offers a step-by-step guide for migrating a BGP based Internet presence to a LISP site. It includes a pre-install/pre-turn-up checklist, and customer and provider activation procedures.

A.1. Customer Pre-Install and Pre-Turn-up Checklist

1. Determine how many current physical service provider connections the customer has and their existing bandwidth and traffic engineering requirements.

This information will determine the number of routing locators, and the priorities and weights that should be configured on the xTRs.

2. Make sure customer router has LISP capabilities.

- * Check OS version of the CE router. If LISP is an add-on, check if it is installed.

This information can be used to determine if the platform is appropriate to support LISP, in order to determine if a software and/or hardware upgrade is required.

- * Have customer upgrade (if necessary, software and/or hardware) to be LISP capable.

3. Obtain current running configuration of CE router. A suggested LISP router configuration example can be customized to the customer's existing environment.

4. Verify MTU Handling

- * Request increase in MTU to 1556 or more on service provider connections. Prior to MTU change verify that 1500 byte packet from P-xTR to RLOC with do not fragment (DF-bit) bit set.
- * Ensure they are not filtering ICMP unreachable or time-exceeded on their firewall or router.

LISP, like any tunneling protocol, will increase the size of packets when the LISP header is appended. If increasing the MTU of the access links is not possible, care must be taken that ICMP is not being filtered in order to allow for Path MTU Discovery to take place.

5. Validate member prefix allocation.

This step is to check if the prefix used by the customer is a direct (Provider Independent), or if it is a prefix assigned by a physical service provider (Provider Aggregatable). If the prefixes are assigned by other service providers then a Letter of Agreement is required to announce prefixes through the Proxy Service Provider.

6. Verify the member RLOCs and their reachability.

This step ensures that the RLOCs configured on the CE router are in fact reachable and working.

7. Prepare for cut-over.

- * If possible, have a host outside of all security and filtering policies connected to the console port of the edge router or switch.
- * Make sure customer has access to the router in order to configure it.

A.2. Customer Activating LISP Service

1. Customer configures LISP on CE router(s) from service provider recommended configuration.

The LISP configuration consists of the EID prefix, the locators, and the weights and priorities of the mapping between the two values. In addition, the xTR must be configured with Map Resolver(s), Map Server(s) and the shared key for registering to Map Server(s). If required, Proxy-ETR(s) may be configured as well.

In addition to the LISP configuration, the following:

- * Ensure default route(s) to next-hop external neighbors are included and RLOCs are present in configuration.
 - * If two or more routers are used, ensure all RLOCs are included in the LISP configuration on all routers.
 - * It will be necessary to redistribute default route via IGP between the external routers.
2. When transition is ready perform a soft shutdown on existing eBGP peer session(s)
 - * From CE router, use LIG to ensure registration is successful.
 - * To verify LISP connectivity, find and ping LISP connected sites. If possible, find ping destinations that are not covered by a prefix in the global BGP routing system, because PITRs may deliver the packets even if LISP connectivity is not working. Traceroutes may help discover if this is the case.
 - * To verify connectivity to non-LISP sites, try accessing a landmark (e.g., a major Internet site) via a web browser.

A.3. Cut-Over Provider Preparation and Changes

1. Verify site configuration and then active registration on Map Server(s)
 - * Authentication key
 - * EID prefix
2. Add EID space to map-cache on proxies
3. Add networks to BGP advertisement on proxies
 - * Modify route-maps/policies on P-xTRs
 - * Modify route policies on core routers (if non-connected member)
 - * Modify ingress policers on core routers
 - * Ensure route announcement in looking glass servers, RouteViews
4. Perform traffic verification test
 - * Ensure MTU handling is as expected (PMTUD working)
 - * Ensure proxy-ITR map-cache population
 - * Ensure access from traceroute/ping servers around Internet
 - * Use a looking glass, to check for external visibility of registration via several Map Resolvers

Authors' Addresses

Lorand Jakab
Cisco Systems
170 Tasman Drive
San Jose, CA 95134
USA

Email: lojakab@cisco.com

Albert Cabellos-Aparicio
Technical University of Catalonia
C/Jordi Girona, s/n
BARCELONA 08034
Spain

Email: acabello@ac.upc.edu

Florin Coras
Technical University of Catalonia
C/Jordi Girona, s/n
BARCELONA 08034
Spain

Email: fcoras@ac.upc.edu

Jordi Domingo-Pascual
Technical University of Catalonia
C/Jordi Girona, s/n
BARCELONA 08034
Spain

Email: jordi.domingo@ac.upc.edu

Darrel Lewis
Cisco Systems
170 Tasman Drive
San Jose, CA 95134
USA

Email: darlewis@cisco.com

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: August 29, 2016

L. Iannone
Telecom ParisTech
D. Lewis
Cisco Systems, Inc.
D. Meyer
Brocade
V. Fuller
February 26, 2016

LISP EID Block
draft-ietf-lisp-eid-block-13.txt

Abstract

This is a direction to IANA to allocate a /32 IPv6 prefix for use with the Locator/ID Separation Protocol (LISP). The prefix will be used for local intra-domain routing and global endpoint identification, by sites deploying LISP as EID (Endpoint Identifier) addressing space.

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 <http://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 August 29, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Definition of Terms	3
3. Rationale and Intent	3
4. Expected use	4
5. Block Dimension	5
6. 3+3 Allocation Plan	6
7. Allocation Lifetime	7
8. Routing Considerations	7
9. Security Considerations	8
10. IANA Considerations	8
11. Acknowledgments	9
12. References	10
12.1. Normative References	10
12.2. Informative References	11
Appendix A. Document Change Log	12
Authors' Addresses	15

1. Introduction

This document directs the IANA to allocate a /32 IPv6 prefix for use with the Locator/ID Separation Protocol (LISP - [RFC6830]), LISP Map Server ([RFC6833]), LISP Alternative Topology (LISP+ALT - [RFC6836]) (or other) mapping systems, and LISP Interworking ([RFC6832]).

This block will be used as global Endpoint IDentifier (EID) space.

2. Definition of Terms

The present document does not introduce any new term with respect to the set of LISP Specifications ([RFC6830], [RFC6831], [RFC6832], [RFC6833], [RFC6834], [RFC6835], [RFC6836], [RFC6837]), but assumes that the reader is familiar with the LISP terminology. [I-D.ietf-lisp-introduction] provides an introduction to the LISP technology, including its terminology.

3. Rationale and Intent

Discussion within the LISP Working Group led to identify several scenarios in which the existence of a LISP specific address block brings technical benefits. Hereafter the most relevant scenarios are described:

Early LISP destination detection: With the current specifications, there is no direct way to detect whether or not a certain destination is in a LISP domain or not without performing a LISP mapping lookup. For instance, if an ITR is sending to all types of destinations (i.e., non-LISP destinations, LISP destinations not in the IPv6 EID block, and LISP destinations in the IPv6 EID block) the only way to understand whether or not to encapsulate the traffic is to perform a cache lookup and, in case of a LISP Cache miss, send a Map-Request to the mapping system. In the meanwhile (waiting the Map-Reply), packets may be dropped in order to avoid excessive buffering.

Avoid penalizing non-LISP traffic: In certain circumstances it might be desirable to configure a router using LISP features to natively forward all packets that have not a destination address in the block, hence, no lookup whatsoever is performed and packets destined to non-LISP sites are not penalized in any manner.

Traffic Engineering: In some deployment scenarios it might be desirable to apply different traffic engineering policies for LISP and non-LISP traffic. A LISP specific EID block would allow improved traffic engineering capabilities with respect to LISP vs. non-LISP traffic. In particular, LISP traffic might be identified without having to use DPI techniques in order to parse the encapsulated packet, performing instead a simple inspection of the outer header is sufficient.

Transition Mechanism: The existence of a LISP specific EID block may prove useful in transition scenarios. A non-LISP domain would ask for an allocation in the LISP EID block and use it to deploy LISP in its network. Such allocation will not be announced in the BGP routing infrastructure (cf., Section 4). This approach will allow non-LISP domains to avoid fragmenting their already allocated non-LISP addressing space, which may lead to BGP routing table inflation since it may (rightfully) be announced in the BGP routing infrastructure.

Limit the impact on BGP routing infrastructure: As described in the previous scenario, LISP adopters will avoid fragmenting their addressing space, since fragmentation would negatively impact the BGP routing infrastructure. Adopters will use addressing space from the EID block, which might be announced in large aggregates and in a tightly controlled manner only by proxy xTRs.

Is worth mentioning that new use cases can arise in the future, due to new and unforeseen scenarios.

Furthermore, the use of a dedicated address block will give a tighter control, especially filtering, over the traffic in the initial experimental phase, while facilitating its large-scale deployment.

[RFC3692] considers assigning experimental and testing numbers useful, and the request of a reserved IPv6 prefix is a perfect match of such practice. The present document follows the guidelines provided in [RFC3692], with one exception. [RFC3692] suggests the use of values similar to those called "Private Use" in [RFC5226], which by definition are not unique. One of the purposes of the present request to IANA is to guarantee uniqueness to the EID block. The lack thereof would result in a lack of real utility of a reserved IPv6 prefix.

4. Expected use

Sites planning to deploy LISP may request a prefix in the IPv6 EID

block. Such prefixes will be used for routing and endpoint identification inside the site requesting it. Mappings related to such prefix, or part of it, will be made available through the mapping system in use and registered to one or more Map Server(s).

The EID block must be used for LISP experimentation and must not be advertised in the form of more specific route advertisements in the non-LISP inter-domain routing environment. Interworking between the EID block sub-prefixes and the non-LISP Internet is done according to [RFC6832] and [RFC7215].

As the LISP adoption progresses, the EID block may potentially have a reduced impact on the BGP routing infrastructure, compared to the case of having the same number of adopters using global unicast space allocated by RIRs ([MobiArch2007]). From a short-term perspective, the EID block offers potentially large aggregation capabilities since it is announced by PxTRs possibly concentrating several contiguous prefixes. This trend should continue with even lower impact from a long-term perspective, since more aggressive aggregation can be used, potentially leading at using few PxTRs announcing the whole EID block ([FIABook2010]).

The EID block will be used only at configuration level, it is recommended not to hard-code in any way the IPv6 EID block in the router hardware. This allows avoiding locking out sites that may want to switch to LISP while keeping their own IPv6 prefix, which is not in the IPv6 EID block. Furthermore, in the case of a future permanent allocation, the allocated prefix may differ from the experimental temporary prefix allocated during the experimentation phase.

With the exception of Pitr case (described in Section 8) prefixes out of the EID block must not be announced in the BGP routing infrastructure.

5. Block Dimension

The working group reached consensus on an initial allocation of a /32 prefix. The reason of such consensus is manifold:

- o The working group agreed that /32 prefix is sufficiently large to cover initial allocation and requests for prefixes in the EID space in the next few years for very large-scale experimentation and deployment.
- o As a comparison, it is worth mentioning that the current LISP Beta Network ([BETA]) is using a /32 prefix, with more than 250 sites

using a /48 sub prefix. Hence, a /32 prefix appears sufficiently large to allow the current deployment to scale up and be open for interoperation with independent deployments using EIDs in the new /32 prefix.

- o A /32 prefix is sufficiently large to allow deployment of independent (commercial) LISP enabled networks by third parties, but may as well boost LISP experimentation and deployment.
- o The use of a /32 prefix is in line with previous similar prefix allocation for tunneling protocols ([RFC3056]).

6. 3+3 Allocation Plan

This document requests IANA to initially assign a /32 prefix out of the IPv6 addressing space for use as EID in LISP (Locator/ID Separation Protocol).

IANA allocates the requested address space by MMMM/YYYY0 for a duration of 3 (three) initial years (through MMMM/YYYY3), with an option to extend this period by 3 (three) more years (until MMMM/YYYY6). By the end of the first period, the IETF will provide a decision on whether to transform the prefix in a permanent assignment or to put it back in the free pool (see Section 7 for more information).

[RFC Editor: please replace MMMM and all its occurrences in the document with the month of publication as RFC.]

[RFC Editor: please replace YYYY0 and all its occurrences in the document with the year of publication as RFC.]

[RFC Editor: please replace YYYY3 and all its occurrences in the document with the year of publication as RFC plus 3 years, e.g., if published in 2016 then put 2019.]

[RFC Editor: please replace YYYY6 and all its occurrences in the document with the year of publication as RFC plus 6 years, e.g., if published in 2016 then put 2022.]

In the first case, i.e., if the IETF decides to transform the block in a permanent allocation, the EID block allocation period will be extended for three years (until MMMM/YYYY6) so to give time to the IETF to define the final size of the EID block and create a transition plan. The transition of the EID block into a permanent allocation has the potential to pose policy issues (as recognized in [RFC2860], section 4.3) and hence discussion with the IANA, the RIR

communities, and the IETF community will be necessary to determine appropriate policy for permanent EID block allocation and management. Note as well that the final permanent allocation may differ from the initial experimental assignment, hence, it is recommended not to hard-code in any way the experimental EID block on LISP-capable devices.

In the latter case, i.e., if the IETF decides to stop the EID block experimental use, by MMMM/YYYY3 all temporary prefix allocations in such address range must expire and be released, so that the entire /32 is returned to the free pool.

The allocation and management of the EID block for the initial 3 years period (and the optional 3 more years) is detailed in [I-D.ietf-lisp-eid-block-mgmt].

7. Allocation Lifetime

If no explicit action is carried out by the end of the experiment (by MMMM/YYYY3) it is automatically considered that there was no sufficient interest in having a permanent allocation and the address block will be returned to the free pool.

Otherwise, if the LISP Working Group recognizes that there is value in having a permanent allocation then explicit action is needed.

In order to trigger the process for a permanent allocation a document is required. Such document has to articulate the rationale why a permanent allocation would be beneficial. More specifically, the document has to detail the experience gained during experimentation and all of the technical benefits provided by the use of a LISP specific prefix. Such technical benefits are expected to lay in the scenarios described in Section 3, however, new unforeseen benefits may appear during experimentation. The description should be sufficiently articulate so to allow to provide an estimation of what should be the size of the permanent allocation. Note however that, as explained in Section 6, it is up to IANA to decide which address block will be used as permanent allocation and that such block may be different from the temporary experimental allocation.

8. Routing Considerations

In order to provide connectivity between the Legacy Internet and LISP sites, PITRs announcing large aggregates (ideally one single large aggregate) of the IPv6 EID block could be deployed. By doing so, PITRs will attract traffic destined to LISP sites in order to

encapsulate and forward it toward the specific destination LISP site. Routers in the Legacy Internet must treat announcements of prefixes from the IPv6 EID block as normal announcements, applying best current practice for traffic engineering and security.

Even in a LISP site, not all routers need to run LISP elements. In particular, routers that are not at the border of the local domain, used only for intra-domain routing, do not need to provide any specific LISP functionality but must be able to route traffic using addresses in the IPv6 EID block.

For the above-mentioned reasons, routers that do not run any LISP element, must not include any special handling code or hardware for addresses in the IPv6 EID block. In particular, it is recommended that the default router configuration does not handle such addresses in any special way. Doing differently could prevent communication between the Legacy Internet and LISP sites or even break local intra-domain connectivity.

9. Security Considerations

This document does not introduce new security threats in the LISP architecture nor in the legacy Internet architecture.

10. IANA Considerations

This document instructs the IANA to assign a /32 IPv6 prefix for use as the global LISP EID space using a hierarchical allocation as outlined in [RFC5226] and summarized in Table 1.

This document does not specify any specific value for the requested address block but suggests that should come from the 2000::/3 Global Unicast Space. IANA is not requested to issue an AS0 ROA (Route Origin Attestation [RFC6491]), since the Global EID Space will be used for routing purposes.

Attribute	Value
Address Block	2001:5::/32
Name	EID Space for LISP
RFC	[This Document]
Allocation Date	2015
Termination Date	MMMM/YYYY3 [1]
Source	True [2]
Destination	True
Forwardable	True
Global	True
Reserved-by-protocol	True [3]

[1] According to the 3+3 Plan outlined in this document termination date can be postponed to MMMM/YYYY6. [2] Can be used as a multicast source as well. [3] To be used as EID space by LISP [RFC6830] enabled routers.

Table 1: Global EID Space

[IANA: Please update the Termination Date and footnote [1] in the Special-Purpose Address Registry when the I-D is published as RFC.]

The reserved address space is requested for a period of time of three initial years starting in MMMM/YYYY0 (until MMMM/YYYY3), with an option to extend it by three years (until MMMM/YYYY6) up on decision of the IETF (see Section 6 and Section 7). Following the policies outlined in [RFC5226], upon IETF Review, by MMMM/YYYY3 decision should be made on whether to have a permanent EID block assignment. If no explicit action is taken or if the IETF review outcome will be that is not worth to have a reserved prefix as global EID space, the whole /32 will be taken out from the IPv6 Special Purpose Address Registry and put back in the free pool managed by IANA.

Allocation and management of the Global EID Space is detailed in a different document. Nevertheless, all prefix allocations out of this space must be temporary and no allocation must go beyond MMMM/YYYY3 unless the IETF Review decides for a permanent Global EID Space assignment.

11. Acknowledgments

Special thanks to Roque Gagliano for his suggestions and pointers. Thanks to Alvaro Retana, Deborah Brungard, Ron Bonica, Damien Saucez, David Conrad, Scott Bradner, John Curran, Paul Wilson, Geoff Huston,

Wes George, Arturo Servin, Sander Steffann, Brian Carpenter, Roger Jorgensen, Terry Manderson, Brian Haberman, Adrian Farrel, Job Snijders, Marla Azinger, Chris Morrow, and Peter Schoenmaker, for their insightful comments. Thanks as well to all participants to the fruitful discussions on the IETF mailing list.

The work of Luigi Iannone has been partially supported by the ANR-13-INFR-0009 LISP-Lab Project (www.lisp-lab.org) and the EIT KIC ICT-Labs SOFNETS Project.

12. References

12.1. Normative References

- [I-D.ietf-lisp-eid-block-mgmt] Iannone, L., Jorgensen, R., Conrad, D., and G. Huston, "LISP EID Block Management Guidelines", draft-ietf-lisp-eid-block-mgmt-06 (work in progress), August 2015.
- [RFC2860] Carpenter, B., Baker, F., and M. Roberts, "Memorandum of Understanding Concerning the Technical Work of the Internet Assigned Numbers Authority", RFC 2860, DOI 10.17487/RFC2860, June 2000, <<http://www.rfc-editor.org/info/rfc2860>>.
- [RFC3692] Narten, T., "Assigning Experimental and Testing Numbers Considered Useful", BCP 82, RFC 3692, DOI 10.17487/RFC3692, January 2004, <<http://www.rfc-editor.org/info/rfc3692>>.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <<http://www.rfc-editor.org/info/rfc5226>>.
- [RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <<http://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, <<http://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, <<http://www.rfc-editor.org/info/rfc6832>>.

- [RFC6833] Fuller, V. and D. Farinacci, "Locator/ID Separation Protocol (LISP) Map-Server Interface", RFC 6833, DOI 10.17487/RFC6833, January 2013, <<http://www.rfc-editor.org/info/rfc6833>>.
- [RFC6834] Iannone, L., Saucez, D., and O. Bonaventure, "Locator/ID Separation Protocol (LISP) Map-Versioning", RFC 6834, DOI 10.17487/RFC6834, January 2013, <<http://www.rfc-editor.org/info/rfc6834>>.
- [RFC6835] Farinacci, D. and D. Meyer, "The Locator/ID Separation Protocol Internet Groper (LIG)", RFC 6835, DOI 10.17487/RFC6835, January 2013, <<http://www.rfc-editor.org/info/rfc6835>>.
- [RFC6836] Fuller, V., Farinacci, D., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol Alternative Logical Topology (LISP+ALT)", RFC 6836, DOI 10.17487/RFC6836, January 2013, <<http://www.rfc-editor.org/info/rfc6836>>.
- [RFC6837] Lear, E., "NERD: A Not-so-novel Endpoint ID (EID) to Routing Locator (RLOC) Database", RFC 6837, DOI 10.17487/RFC6837, January 2013, <<http://www.rfc-editor.org/info/rfc6837>>.

12.2. Informative References

- [BETA] LISP Beta Network, "<http://www.lisp4.net>".
- [FIABook2010] L. Iannone, T. Leva, "Modeling the economics of Loc/ID Separation for the Future Internet.", Towards the Future Internet - Emerging Trends from the European Research, Pages 11-20, ISBN: 9781607505389, IOS Press , May 2010.
- [I-D.ietf-lisp-introduction] Cabellos-Aparicio, A. and D. Saucez, "An Architectural Introduction to the Locator/ID Separation Protocol (LISP)", draft-ietf-lisp-introduction-13 (work in progress), April 2015.
- [MobiArch2007] B. Quoitin, L. Iannone, C. de Launois, O. Bonaventure,

"Evaluating the Benefits of the Locator/Identifier Separation", The 2nd ACM-SIGCOMM International Workshop on Mobility in the Evolving Internet Architecture (MobiArch'07) , August 2007.

- [RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, DOI 10.17487/RFC3056, February 2001, <<http://www.rfc-editor.org/info/rfc3056>>.
- [RFC6491] Manderson, T., Vegoda, L., and S. Kent, "Resource Public Key Infrastructure (RPKI) Objects Issued by IANA", RFC 6491, DOI 10.17487/RFC6491, February 2012, <<http://www.rfc-editor.org/info/rfc6491>>.
- [RFC7215] Jakab, L., Cabellos-Aparicio, A., Coras, F., Domingo-Pascual, J., and D. Lewis, "Locator/Identifier Separation Protocol (LISP) Network Element Deployment Considerations", RFC 7215, DOI 10.17487/RFC7215, April 2014, <<http://www.rfc-editor.org/info/rfc7215>>.

Appendix A. Document Change Log

[RFC Editor: Please remove this section on publication as RFC]

Version 13 Posted MMMM 2016.

- o Changed I-D type from "Informational" to "Experimental" as requested by A. Retana during IESG review.
- o Dropped the appendix "LISP Terminology"; replaced by pointer to the LISP Introduction document.
- o Added Section 7 to clarify the process after the 3 years experimental allocation.
- o Modified the dates, introducing variables, so to allow RFC Editor to easily update dates by publication as RFC.

Version 12 Posted May 2015.

- o Fixed typos and references as suggested by the Gen-ART and OPS-DIR review.

Version 11 Posted April 2015.

- o In Section 4, deleted contradictory text on EID prefix advertisement in non-LISP inter-domain routing environments.

- o In Section 3 deleted the "Avoid excessive stretch" bullet, because confusing.
- o Deleted last bullet of the list in Section 3 because redundant w.r.t. global content of the document.

Version 10 Posted January 2015.

- o Keep alive version

Version 09 Posted July 2014.

- o Few Editorial modifications as requested by D. Saucez, as shepherd, during the write up of the document.
- o Allocation date postponed to beginning 2015, as suggested by D. Saucez.

Version 08 Posted January 2014.

- o Modified Section 4 as suggested by G. Houston.

Version 07 Posted November 2013.

- o Modified the document so to request a /32 allocation, as for the consensus reached during IETF 88th.

Version 06 Posted October 2013.

- o Clarified the rationale and intent of the EID block request with respect to [RFC3692], as suggested by S. Bradner and J. Curran.
- o Extended Section 3 by adding the transition scenario (as suggested by J. Curran) and the TE scenario. The other scenarios have been also edited.
- o Section 6 has been re-written to introduce the 3+3 allocation plan as suggested by B. Haberman and discussed during 86th IETF.
- o Section 10 has also been updated to the 3+3 years allocation plan.
- o Moved Section 11 at the end of the document.
- o Changed the original Definition of terms to an appendix.

Version 05 Posted September 2013.

- o No changes.

Version 04 Posted February 2013.

- o Added Table 1 as requested by IANA.
- o Transformed the prefix request in a temporary request as suggested by various comments during IETF Last Call.
- o Added discussion about short/long term impact on BGP in Section 4 as requested by B. Carpenter.

Version 03 Posted November 2012.

- o General review of Section 5 as requested by T. Manderson and B. Haberman.
- o Dropped RFC 2119 Notation, as requested by A. Farrel and B. Haberman.
- o Changed "IETF Consensus" to "IETF Review" as pointed out by Roque Gagliano.
- o Changed every occurrence of "Map-Server" and "Map-Resolver" with "Map Server" and "Map Resolver" to make the document consistent with [RFC6833]. Thanks to Job Snijders for pointing out the issue.

Version 02 Posted April 2012.

- o Fixed typos, nits, references.
- o Deleted reference to IANA allocation policies.

Version 01 Posted October 2011.

- o Added Section 5.

Version 00 Posted July 2011.

- o Updated section "IANA Considerations"
- o Added section "Rationale and Intent" explaining why the EID block allocation is useful.
- o Added section "Expected Use" explaining how sites can request and use a prefix in the IPv6 EID Block.

- o Added section "Action Plan" suggesting IANA to avoid allocating address space adjacent the allocated EID block in order to accommodate future EID space requests.
- o Added section "Routing Consideration" describing how routers not running LISP deal with the requested address block.
- o Added the present section to keep track of changes.
- o Rename of draft-meyer-lisp-eid-block-02.txt.

Authors' Addresses

Luigi Iannone
Telecom ParisTech

Email: ggx@gigix.net

Darrel Lewis
Cisco Systems, Inc.

Email: darlewis@cisco.com

David Meyer
Brocade

Email: dmm@1-4-5.net

Vince Fuller

Email: vaf@vaf.net

LISP Working Group
Internet-Draft
Intended status: Informational
Expires: April 24, 2014

J. N. Chiappa
Yorktown Museum of Asian Art
October 21, 2013

An Architectural Introduction to the LISP
Location-Identity Separation System
draft-ietf-lisp-introduction-03

Abstract

LISP is an upgrade to the architecture of the IP internetworking system, one which separates location and identity properties (previously intermingled in IP addresses). This document is an introductory overview of the entire LISP system, and focuses on describing the major concepts and functional sub-systems of LISP, and the interactions between them.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79. This document may not be modified, and derivative works of it may not be created, except to format it for publication as an RFC or to translate it into languages other than English.

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 <http://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 April 24, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Prefatory Note
2. Part I
3. Initial Glossary

4. Background
5. Deployment Philosophy
 - 5.1. Economics
 - 5.2. Maximize Re-use of Existing Mechanism
6. LISP Overview
 - 6.1. Basic Approach
 - 6.2. Basic Functionality
 - 6.3. Mapping from EIDs to RLOCs
 - 6.4. Interworking With Non-LISP-Capable Endpoints
 - 6.5. Security in LISP
7. Initial Applications
 - 7.1. Provider Independence
 - 7.2. Multi-Homing
 - 7.3. Traffic Engineering
 - 7.4. Routing
 - 7.5. Mobility
 - 7.6. Traversal Across Alternate IP Versions
 - 7.7. Virtual Private Networks
 - 7.8. Local Uses
8. Major Functional Subsystems
 - 8.1. Data Plane - xTRs Overview
 - 8.1.1. Mapping Cache Performance
 - 8.2. Control Plane - Mapping System Overview
 - 8.2.1. Mapping System Organization
 - 8.2.2. Interface to the Mapping System
 - 8.2.3. Indexing Sub-system
9. Examples of Operation
 - 9.1. An Ordinary Packet's Processing
 - 9.2. A Mapping Cache Miss
10. Part II
11. Design Approach
12. xTRs
 - 12.1. When to Encapsulate
 - 12.2. UDP Encapsulation Details
 - 12.3. Header Control Channel
 - 12.3.1. Mapping Versioning
 - 12.3.2. Echo Nonces
 - 12.3.3. Instances
 - 12.4. Probing
 - 12.5. Mapping Lifetimes and Timeouts
 - 12.6. Mapping Gleaning in ETRs
 - 12.7. MTU Issues
 - 12.8. Security of Mapping Lookups
 - 12.9. xTR Mapping Cache Performance
13. The Mapping System
 - 13.1. The Mapping System Interface
 - 13.1.1. Map-Request Messages
 - 13.1.2. Map-Reply Messages
 - 13.1.3. Map-Register and Map-Notify Messages
 - 13.2. The DDT Indexing Sub-system
 - 13.2.1. Map-Referral Messages
 - 13.3. Reliability via Replication
 - 13.4. Security of the DDT Indexing Sub-system
 - 13.5. Extended Capabilities
 - 13.6. Performance of the Mapping System
14. Multicast Support in LISP
 - 14.1. Basic Concepts of Multicast Support in LISP
 - 14.2. Initial Multicast Support in LISP
15. Deployment Issues and Mechanisms
 - 15.1. LISP Deployment Needs
 - 15.2. Interworking Mechanisms
 - 15.2.1. Proxy LISP Routers

- 15.2.2. LISP-NAT
- 15.3. Use Through NAT Devices
- 15.4. LISP and Core Internet Routing
- 16. Fault Discovery/Handling
 - 16.1. Handling Missing Mappings
 - 16.2. Outdated Mappings
 - 16.2.1. Outdated Mappings - Updated Mapping
 - 16.2.2. Outdated Mappings - Wrong ETR
 - 16.2.3. Outdated Mappings - No Longer an ETR
 - 16.3. Erroneous Mappings
 - 16.4. Verifying ETR Liveness
 - 16.5. Verifying ETR Reachability
- 17. Acknowledgments
- 18. IANA Considerations
- 19. Security Considerations
- 20. References
 - 20.1. Normative References
 - 20.2. Informative References
- Appendix A. Glossary/Definition of Terms
- Appendix B. Other Appendices
 - B.1. A Brief History of Location/Identity Separation
 - B.2. A Brief History of the LISP Project
 - B.3. Old LISP 'Models'
 - B.4. The ALT Mapping Indexing Sub-system
 - B.5. Early NAT Support

1. Prefatory Note

This document is the first of a pair which, together, form what one would think of as the 'architecture document' for LISP (the 'Location-Identity Separation Protocol'). Much of what would normally be in an architecture document (e.g. the architectural design principles used in LISP, and the design considerations behind various components and aspects of the LISP system) is in the second document, the 'Architectural Perspective on LISP' document. [Perspective]

This 'Architectural Introduction' document is primarily intended for those who unfamiliar with LISP, and want to start learning about it. It is intended primarily for those working on LISP, but those working with LISP, and more generally anyone who wants to know more about LISP, may also find this document useful.

This document is intended to both be easy to follow, and also to give the reader a choice as to how much they wish to know about LISP. It is structured as a series of phases, each covering the entire system, but with ever-increasing detail. Reading only the first part of the document will give a good high-level view of the system; reading the complete document should provide a fairly detailed understanding of the entire system.

People who just want to get an idea of how LISP works might only read the first part; they can stop reading either just before, or just after, Section 9, "Examples of Operation". People who are going to go on and read the protocol specifications (perhaps to implement LISP) should read the entire document.

Note: This document is a descriptive document, not a protocol specification. Should it differ in any detail from any of the LISP protocol specification documents, they take precedence for the actual operation of the protocol.

2. Part I

3. Initial Glossary

This initial glossary defines a few general terms which will be useful to have in hand when commencing reading this document. A complete glossary is available in Appendix A.

A note about style: initial usage of a term defined in the glossary is denoted with double quotation marks ("). Other uses of quotations (e.g. for quotations, euphemisms, etc) use single quotation marks (').

- Name: In this document, and in much of computer science, a 'name' simply refers to an identifier for an object or entity. Names have both semantics (meaning) and syntax (form). [RFC1498]
- Namespace: A group of "names" with matching semantics and syntax; they usually, but not always, refer to members of a class of identical objects.
- Mapping: In this document, a connection (or binding, to use the computer science term) between two names, one in each of two namespaces.
- Delegation Hierarchy: an abstract rooted tree (in the graph theory sense of the term) which is a virtual representation of the delegation of a "namespace" into smaller and smaller blocks, in a recursive process.
- Node: The general term used to describe any sort of communicating entity; it might be a physical or a virtual host, or a mobile device of some sort. It includes both entities which forward packets, and entities which create or consume packets. It was deliberately chosen for use in this document precisely because its definition is not fixed, and therefore unlikely to cause erroneous images in the minds of readers.
- Switch, Packet Switch: A packet switch, in the general meaning of that term. A device which takes in packets from its interfaces and forwards them on, either to a next-hop switch, or to the final destination. They may operate at either the network layer (e.g. ARPANET), or internetwork layer. [Baran][Heart][RFC1812]
- Endpoint, end-end communication entity: The fate-sharing region at one end of an end-end communication; the collection of state related to both the reliable end-end communication channel, and the applications running there. [Chiappa]
- IPvN: IPv4 ([RFC791]) or IPv6 ([RFC2460]); the two are so similar, in fundamental architecture, that in much discussion about their capabilities, limitations, etc statements about the apply equally to both, and to continually say 'IPv4 and IPv6' quickly becomes tedious.
- Address: In this document, and in current "IPvN" and similar networking suites, a "name" which has mixed semantics, in that it includes both identity ('who') and location ('where') semantics. [Atkinson]
- Address Block, Block: A contiguous section of a namespace, usually IPvN addresses; for the latter, it will normally be on a bit boundary, using the standard 'prefix/length' selection indication.
- Identifier: Here, and in current networking discussions, a "name" which has purely identity semantics.
- Locator: Originally defined as a "name" with only location semantics, and one that was not necessarily carried in every packet (as was widely assumed of "addresses") [RFC1992], it is now generally taken, including here, to mean a "name" with purely location semantics.
- Site: A collection of hosts, routers and networks under a single

- administrative control.
- LISP site: A single node, or a set of network elements in an edge network under the administrative control of a single organization; they are separated from the rest of the network by "LISP routers".
- LISP node: A IPvN "node" which has been enhanced with LISP functionality; generally this means it can process some subset of LISP control plane traffic.
- LISP router: A IPvN "switch" which has been enhanced with LISP functionality; a LISP node which can forward user traffic.
- LISP host: A IPvN host which is 'behind' (from the point of view of the rest of the network) a "LISP router".

4. Background

It has gradually been realized in the networking community that networks, especially large networks, should deal quite separately with the 'identity' and 'location' of an "endpoint" - basically, 'who' an endpoint is, and 'where' it is. ([RFC1498]) (A more detailed history of this evolution is in Appendix B.1, "A Brief History of Location/Identity Separation".)

At the moment, in both IPv4 and IPv6, IP "addresses" indicate both where the named "node" is, as well as identify it for purposes of end-end communication; i.e. it has both location and identity properties. However, the separation of those two properties is a step which has recently been identified by the IRTF as a necessary evolutionary architectural step for the Internet. [RFC6115]

The on-going LISP project is an attempt to provide a viable path towards this separation. (A brief history of the LISP project can be found in Appendix B.2, "A Brief History of the LISP Project".)

As an add-on to a large existing system, it has had to make certain compromises. (For a good example, see [Perspective], Section "Residual Location Functionality in EIDs".) However, if it reaches near-ubiquitous deployment, it will have two important consequences.

First, in effectively providing separation of location and identity, along with providing a distributed directory of the "mappings" between them, 'Wheeler's Law' ('All problems in computer science can be solved by another level of indirection') will come into play, and the Internet technical community will have a new, immensely powerful, tool at its disposal. The fact that the namespaces on both sides of the mapping are global ones maximizes the power of that tool. (See [Perspective], Section "Need for a Mapping System", for more on this.)

Second, because of a combination of the flexible capability built into LISP, and the breaking of the unification of location and identity names, further architectural evolution of the Internet becomes easily available; for example, new namespaces for location could be designed and deployed. In other words, LISP is not a point solution to meet a particular need, but hopefully an 'escape hatch' which will allow further significant enhancement to the Internet's overall architecture. (See [Future] for more on this.)

5. Deployment Philosophy

The deployment philosophy was a major driver for much of the design of LISP: to some degree of the architecture, and to a very large measure, the engineering.

Experience over the last several decades has shown that having a viable 'deployment model' for a new design is absolutely key to the success of that design. In general, it is comparatively easy to conceive of new network designs, but much harder to devise approaches which will actually get deployed throughout the global network. A new design may be fantastic - but if it can not or will not be successfully deployed (for whatever factors), it is useless.

This absolute primacy of what is hoped is a viable deployment model is what has lead to some painful compromises in the design; and the extreme focus on a viable deployment model (including economics) is one of the key design guides of LISP.

LISP aims to achieve the near-ubiquitous deployment necessary for maximum exploitation of an architectural upgrade by i) minimizing the amount of change needed (most existing hosts and routers can operate unmodified); and ii) by providing significant benefits to early adopters.

5.1. Economics

A key factor in successful adoption is economics: does the new design have benefits which outweigh its costs?

More importantly, this balance needs to hold for early adopters - because if they do not receive benefits to their adoption, the sphere of earliest adopters will not expand, and it will never get to widespread deployment.

This is particularly true of architectural enhancements, which are far less likely to be an addition which one can 'bolt onto the side' of existing mechanisms, and often offer their greatest benefits only when widely (or ubiquitously) deployed.

Maximizing the cost-benefit ratio obviously has two aspects. First, on the cost side, by making the design as inexpensive as possible, which means in part making the deployment as easy as possible. Second, on the benefit side, by providing many new capabilities, which is best done not by loading the design up with lots of features or options (which adds complexity), but by making the addition powerful through deeper flexibility. The LISP community believes LISP has met both of these goals.

5.2. Maximize Re-use of Existing Mechanism

One key part of reducing the cost of a new design is to absolutely minimize the amount of change required to existing, deployed, devices: the fewer devices need to be changed, and the smaller the change to those that do, the lower the pain (and thus the greater the likelihood) of deployment.

Designs which absolutely require 'forklift upgrades' to large amounts of existing gear are far less likely to succeed - because they have to have extremely large benefits to make their very substantial costs worthwhile.

It is for this reason that LISP, in most cases, initially requires no changes to almost all existing devices in the Internet (both hosts and routers); LISP functionality needs to be added in only a few places (see Section 15.1, "LISP Deployment Needs", for more).

LISP also initially re-uses, where-ever possible, existing protocols.

The 'initially' must be stressed - careful attention has also long been paid to the long-term future (see [Future]), and larger changes become feasible as deployment increases.

6. LISP Overview

LISP is an incrementally deployable architectural upgrade to the existing Internet infrastructure, one which provides separation of location and identity. It thus starts to separate the names used for identity and location of nodes, which are currently unified in "IPvN" "addresses".

The separation into names with purely location and purely identity semantics is usually - but not necessarily - not perfect, for reasons which are driven by the deployment philosophy (above), and explored in more detail elsewhere (in [Perspective], Section "Namespaces-EIDs-Residual").

6.1. Basic Approach

In LISP, the first key concept is that nodes have both an 'identifier' (a name which serves only to provide a persistent handle for the node), called an "EID" (short for 'endpoint identifier'), and an associated 'locator' (a name which says where the node is, in the network's connectivity structure), called an "RLOC" (short for 'routing locator').

A node may be associated with more than one RLOC, or the RLOC may change over time (e.g. if the node is mobile), but it would normally always have the same EID.

The second key concept is that if one wants to be as forward-looking as possible, conceptually one should think of the two kinds of names (EIDs and RLOCs) as naming different classes of entities.

EIDs name nodes - or rather, their end-end communication entities (see [Chiappa] for more). RLOC(s), on the other hand, name interfaces, i.e. places to which the system of routers sends packets. (These will usually be on the "LISP routers", in the early stages of LISP deployment; see below for more.)

This distinction, the formal recognition of different kinds of entities ("endpoints" and interfaces), and their association with the two different classes of names, is also important. Clearly recognizing interfaces and endpoints as distinctly separate classes of objects is another improvement to the existing Internet architecture.

An important insight in LISP is that it initially uses existing IPvN addresses for both of these kinds of names, as opposed to some similar earlier deployment proposals for separation of location and identity (e.g. [RFC1992]), which proposed using a new namespace for locators. This choice minimized LISP's deployment cost, as well as providing the ability to easily interact with un-modified hosts and routers.

The capability to use namespaces other than IPvN addresses for both kinds of names is already built in, which is expected to greatly increase the long-term benefits, flexibility, and power of the LISP "mapping" layer. [AFI][LCAF]

6.2. Basic Functionality

The basic operation of LISP, as it currently stands, is quite simple. LISP augmented packet switches, "LISP routers", near the source and destination of packets intercept traffic, and 'enhance' the packets for the trip between the LISP switches.

The LISP router near the original source (the Ingress Tunnel Router, or "ITR") looks up additional information about the destination of the packet, and then wraps the packet in an outer header, one which contains some of that additional information.

The LISP router near the destination, the (the Egress Tunnel Router, or "ETR") removes that header, leaving the original, un-modified, packet to be sent on to the original destination node.

The overall processing is shown below, in Figure 1:

(to be added)

Figure 1: Basic LISP Packet Flow

To retrieve that additional information, the ITR uses the information in the original packet about the identity of its ultimate destination, i.e. the destination address; in LISP, this is the EID of the ultimate destination. It uses the destination EID to look up the current location (the RLOC) of that EID.

The lookup is performed through a "mapping system", which is the heart of LISP: it is a distributed directory of "mappings" from EIDs to RLOCs. The destination RLOC(s) will normally be the address(es) of the ETR(s) near the ultimate destination.

The ITR then generates a new outer header for the original packet, with that header containing the ETR's RLOC as the wrapped packet's destination, and the ITR's own address (i.e. the RLOC usually associated with the original source) as the wrapped packet's source, and sends it off.

When the packet arrives at the ETR, that outer header is stripped off, and the original packet is forwarded to the original ultimate destination for normal processing.

Return traffic is handled similarly, often (depending on the network's configuration) with the original ITR and ETR switching roles. The ETR and ITR functionality is usually co-located in a single LISP router; these are normally denominated as "xTRs".

6.3. Mapping from EIDs to RLOCs

The "mappings" from EIDs to RLOCs are provided by a distributed, and potentially replicated, database, the "mapping database", which is the heart of LISP. (Here, and in other places in LISP, the replication is not a deep architectural concept, simply an engineering device to obtain reliability via potential redundancy.)

Entities which need mappings get them from the "mapping system", which is a collection of sub-systems through which clients can find and obtain mappings. (The mapping system will be discussed in more detail below, in Section 8.2, "Control Plane - Mapping System Overview" and Section 13, "The Mapping System".)

Mappings are normally distributed via a 'pull' mechanism; in

other words, they are generally not pre-loaded, but requested on demand. Once obtained by an ITR, they are cached by the ITR, for performance reasons.

Extensive studies, including large-scale simulations driven by lengthy recordings of actual traffic at several major sites, have been performed to verify that this 'pull and cache' approach is viable, in practical engineering terms. (This subject will be discussed in more detail in Section 12.9, "xTR Mapping Cache Performance", below, including references to the studies.)

6.4. Interworking With Non-LISP-Capable Endpoints

It is clearly crucial to provide the capability for 'easy' interoperation between "LISP hosts" - i.e. they are behind xTRs, and their EIDs are in the mapping database - and existing non-LISP-using hosts (often called 'legacy' hosts) or legacy "sites".

To allow such interoperation, a number of mechanisms have been designed. One approach uses proxy LISP routers, called "PITRs" (proxy ITRs) and "PETRs" (proxy ETRs), to provide LISP functionality during interaction with legacy hosts. Another approach uses a router with combined LISP and NAT ([RFC1631]) functionality, named a LISP-NAT.

(See Section 15.2.1, "Proxy LISP Routers", and Section 15.2.2, "LISP-NAT", respectively, for details of each, and their respective advantages and disadvantages.)

6.5. Security in LISP

To provide a brief overview of security in LISP, it is definitely understood that LISP needs to be highly securable, especially in the long term; over time, the attacks mounted by 'bad guys' are becoming more and more sophisticated. So LISP, like DNS, needs to be capable of providing 'the very best' security there is.

At the same time, there is a conflicting goal: it must be deployable at a viable cost. That means two things: First, as an experiment, we cannot expect to create the complete security apparatus which we might see in the finished product, including both design and implementation. Second, security needs to be flexible, so that we don't overload the users with more security than they need at any point.

To accomplish these divergent goals, the approach taken is to first analyze what LISP needs for security. [Threats]. Then, steps can be taken to ensure that the appropriate 'hooks' (such as packet fields) are included at an early stage, when doing so is still easy. Over time, additional mechanisms will be fully specified, implemented, and deployed.

LISP does already include a number of security mechanisms; in particular, requesting mappings can be secured (see Section 12.8, "Security of Mapping Lookups"), as can registering of xTRs (see Section 13.1.3, "Map-Register and Map-Notify Messages"); the key database of the mapping system is also secured (see Section 13.4, "Security of the DDT Indexing Sub-system").

The existing security mechanisms, and their configuration (which is mostly manual at this point) currently in LISP are felt to be adequate for the needs of the on-going early stages of deployment;

experience will indicate when improvements are required (within the constraints of the conflicting goal given above).

For more on LISP's security philosophy; see [Perspective], Section "Security", where it is laid out in some detail.

7. Initial Applications

{{Reorder the whole section in popularity order?}}

As previously mentioned, it is felt that LISP will provide even the earliest adopters with some useful capabilities, and that these capabilities will drive early LISP deployment.

It is very important to note that even when used only for interoperation with existing un-modified hosts, use of LISP can still provide benefits to the site which has deployed it - and, perhaps even more importantly, can do so to both sides. This characteristic acts to further enhance the utility for early adopters of LISP. .

Note also that this section only lists some early applications and benefits. See [Perspective], in the Section "Goals of LISP", for a more extensive discussion of some of what LISP might ultimately provide.

7.1. Provider Independence

Provider independence (i.e. the ability to easily change one's Internet Service Provider) is a good example of the utility of separating location and identity.

The problem is simple: for the global routing to scale, addresses need to be aggregated; i.e. things which are close in the overall network's connectivity need to have closely related addresses (so-called "provider aggregatable" addresses). [RFC4116] However, if this principle is followed, it means that when an entity switches providers (i.e. it moves to a different 'place' in the network), it has to re-number, a painful undertaking. [RFC5887]

Having separate namespaces for location and identity greatly reduces the problems involved with re-numbering; an organization which moves retains its EIDs (which are how most other parties refer to its nodes), but is allocated new RLOCs, and the mapping system can quickly provide the updated mapping from the EIDs to the new RLOCs.

7.2. Multi-Homing

Multi-homing is another place where the value of separation of location and identity became apparent. There are several different sub-flavours of the multi-homing problem - e.g. depending on whether one wants open TCP connections to keep working, etc - and other axes as well (e.g. site multi-homing versus host multi-homing).

In particular, for the 'keep open connections up' case, without separation of location and identity, with most currently deployed implementations, the only currently feasible approach is to use provider-independent addresses - which moves the problem into the global routing system, with attendant costs. This approach is also not really feasible for host multi-homing.

7.3. Traffic Engineering

{{Needs a fix - not sure what.}}

Traffic engineering (TE) [RFC3272], desirable though this capability is in a global network, is currently somewhat problematic to provide in the Internet. The problem, fundamentally, is that this capability was not foreseen when the Internet was designed, so the support for it via 'hacks' is neither clean, nor flexible.

TE is, fundamentally, a routing issue. However, the current Internet routing architecture, which is basically the Baran design of fifty years ago [Baran] (a single large, distributed computation), is ill-suited to provide TE. The Internet seems a long way from adopting a more-advanced routing architecture, although the basic concepts for such have been known for some time. [RFC1992]

Although the identity-location mapping layer is thus a poor place, architecturally, to provide TE capabilities, it is still an improvement over the current routing tools available for this purpose (e.g. injection of more-specific routes into the global routing table).

In addition, instead of the entire network incurring the costs (through the routing system overhead), when using a mapping layer to provide TE, the overhead is limited to those who are actually communicating with that particular destination.

LISP includes a number of features in the mapping system to support TE. (described in Section 8.2, "Control Plane - Mapping System Overview", below); more details about using LISP for TE can be found in [LISP-TE].

Also, a number of academic papers have explored how LISP can be used to do TE, and how effective it can be. See the online LISP Bibliography ([Bibliography]) for information about them.

7.4. Routing

Multi-homing and Traffic Engineering are both, in some sense, uses of LISP for routing, but there are many other routing-related uses for LISP.

One of the major original motivations for the separation of location and identity in general, and thus LISP, was to reduce the growth of the routing tables in the "Internet core", the part where routes to _all_ ultimate destinations must be available. LISP is expected to help with this; for more detail, see Section 15.4, "LISP and Core Internet Routing", below.

LISP may also have more local applications in which it can help with routing; see, for instance, [CorasBGP].

7.5. Mobility

Mobility is yet another place where separation of location and identity is obviously a key part of a clean, efficient and high-functionality solution. Considerable experimentation has been completed on doing mobility with LISP.

The mobility provided by LISP allows active sessions to survive moves (provided of course that there is not a period of inaccessibility which exceeds a timeout). LISP mobility also will typically have

better packet 'stretch' (i.e. increase in path length) compared to traditional mobility schemes, which use a 'home agent'.

7.6. Traversal Across Alternate IP Versions

Note that LISP inherently supports intermixing of various IP versions for packet carriage; IPv4 packets might well be carried in IPv6, or vice versa, depending on the network's configuration.

This capability allows an 'island' of operation of one type to be automatically tunneled over a stretch of infrastructure which only supports the other type.

While the machinery of LISP may seem too heavy-weight to be good for such a mundane use, this is not intended as a 'sole use' case for deployment of LISP. Rather, it is something which, if LISP is being deployed anyway (for its other advantages), is an added benefit that one gets 'for free'.

7.7. Virtual Private Networks

L2 and L3 {{Need to add text here - This used to be part of 'Local' below, but we decided this was so important it deserved its own section. Maybe move this up further, as it seems to be the most important 'early adopter' application?}}

This includes support of VPN's for segmentation and multi-tenancy (i.e. a spatially separated private VPN whose components are joined together using the public Internet as a backbone).

7.8. Local Uses

LISP has a number of use cases which are within purely organizationally-local contexts, i.e. not in the larger Internet. These fall into two categories: uses seen on the Internet (above), but here on a private (and usually small scale) setting; and applications which do not have a direct analog in the larger Internet, and which apply only to local deployments.

Among the former are multi-homing and IP version traversal. {{This was marked to be deleted - why? The next part doesn't make sense without this first?}}

Among the latter class, non-Internet applications which have no analog on the Internet, are the following example applications: virtual machine mobility in data centers; other non-IP EID types such as local network MAC addresses, or application specific data.

Several of the applications listed in this section are the ones which have been most popular for LISP in practise; these include virtual networks, and virtual machine mobility.

These often show a synergistic tendency, in that a site which installs LISP to do one, often finds that then becomes a small matter to use it for the second. Given all the things which LISP can do, it is hoped that this synergistic effect will continue to expand LISP's uses.

{{Preceeding paragraphs should probably get moved up into VPN section?}}

8. Major Functional Subsystems

LISP has only two major functional sub-systems - the collection of LISP "packet switches" (the xTRs), which form the 'data plane' of LISP; and the "mapping system", the most important part of the 'control plane', which manages the "mapping database".

The purpose and operation of each is described at a high level below, and then, later on, in a fair amount of detail, in separate sections on each (Sections Section 12, "xTRs", and Section 13, "The Mapping System", respectively).

8.1. Data Plane - xTRs Overview

xTRs are packet switches which have been augmented with extra functionality in both the data and control planes. The data plane functions in ITRs include deciding which packets need to be given LISP processing (since packets to non-LISP hosts may be sent as they are); i.e. looking up the mapping; encapsulating (wrapping) the packet; and sending it to the ETR.

This encapsulation is done using UDP [RFC768] (for reasons to be explained below, in Section 12.2, "UDP Encapsulation Details"), along with an additional outer IPvN header (to hold the source and destination RLOCs). To the extent that traffic engineering features are in use for a particular EID, the ITRs implement them as well.

In the ETR, the data plane simply decapsulates (unwraps) the packets, and forwards the now-normal packets to the ultimate destination.

Control plane functions in ITRs include: asking for {EID->RLOC} mappings via request control messages (Map-Request packets); handling the returning reply control messages (Map-Reply packets), which contain the requested information; managing the local "mapping cache" of "mappings"; checking for the "reachability" and "liveness" of their neighbour ETRs; and checking for outdated mappings and requesting updates.

In the ETR, control plane functions include participating in the reachability and liveness function (see Section 16.4, "Verifying ETR Liveness"); interacting with the mapping sub-system to let it know what mapping this ETR can provide (see Section 8.2.2, "Interface to the Mapping System"); and answering requests from ITRs for those mappings (ditto).

8.1.1. Mapping Cache Performance

As mentioned, studies have been performed to verify that caching mappings in ITRs is viable, in practical engineering terms. These studies not only verified that such caching is feasible, but also provided some insight for designing ITR "mapping caches".

Briefly, they took lengthy traces of all packets leaving a large site, over a period of a week or so, and used those to drive simulations which showed how many mappings would be required. It also allowed analysis of how much control traffic (for loading needed mappings) would result, using various cache sizes and replacement algorithms.

A more extended look at the results is given below, in Section 12.9, "xTR Mapping Cache Performance".

Obviously, these studies are all snapshots of a particular point in

time, and as the Internet continues its life-cycle they will increasingly become out-dated. However, they are useful because they provide an insight into how well LISP can be expected to perform, and scale, over time.

8.2. Control Plane - Mapping System Overview

The mapping system's entire purpose is to give ITRs on-demand access to the mapping database, which is a distributed, and potentially replicated, database which holds mappings between EIDs (identity) and RLOCs (location), along with needed ancillary data (e.g. lifetimes).

To be exact, it contains mappings between EID "blocks" and RLOCs (the block size is given explicitly, as part of the syntax). Support for blocks is both for minimizing the administrative configuration overhead, as well as for operational efficiency; e.g. when a group of EIDs are behind a single xTR.

However, the block may be, and sometimes is, as small as a single EID. However, since mappings are only loaded upon demand, if smaller blocks become predominant, then the increased size of the overall database is far less problematic than if the Internet's routing tables came to be dominated by such small entries.

A particular EID (or EID block) may have more than one RLOC, or may change its RLOC(s), while keeping its basic identity.

Also, in general, throughout LISP, anyplace a name (EID, RLOC, etc) appears in a control packet, the packet format also includes an Address Family Identifier (AFI) for that name. [AFI] The inclusion of the AFI allows LISP (and in particular, the mapping system interface, as embodied in those control packets) a great deal of flexibility. (See [Perspective], Section "Namespaces" for more on this.)

Finally, the mapping from an EID (or EID block) contains not just the RLOC(s), but also (for each RLOC for any given EID entry) priority and weight fields (to allow allocation of load between several RLOCs at a given priority); this allows a certain amount of traffic engineering to be accomplished with LISP.

8.2.1. Mapping System Organization

The "mapping system" is actually split into what are effectively three major functional sub-systems (although the latter two are closely integrated, and appear to most entities in the LISP system as a single sub-system).

The first is the actual mappings themselves, collectively the "mapping database"; they are held by the ETRs, and an ITR which needs a mapping gets it (effectively) directly from the ETR. This co-location of the authoritative version of the mappings, and the forwarding functionality which it describes, is an instance of fate-sharing. [Clark]

To find the appropriate ETR(s) to query for the mapping, the second two sub-systems form an 'indexing system', itself also based on a distributed, potentially replicated database. It provides information on which ETR(s) are authoritative sources for the various {EID -> RLOC} mappings which are available. The two sub-systems which form it are the client interface sub-system, and "indexing sub-system" (which holds and provides the actual information).

8.2.2. Interface to the Mapping System

The client interface to the indexing system from an ITR's point of view is not with the indexing sub-system directly; rather, it is through the client-interface sub-system, which is provided by LISP nodes called Map-Resolvers (MRs) and Map-Servers (MSs).

ITRs send request control messages (Map-Request packets) to an MR. (This interface is probably the most important standardized interface in LISP - it is the key to the entire system.)

The MR then uses the indexing sub-system to allow it to forward the Map-Request to an appropriate Map-Server (MS), which in turn sends the Map-Request on to the appropriate ETR. The latter is authoritative for the actual contents of all mappings for those EID namespace blocks which have been delegated to it.

The ETR then formulates reply control messages (Map-Reply packets), which are sent to the ITR. The details of the indexing sub-system are thus hidden from the ITRs.

(Note that in some cases, it is desirable for the MS to reply on behalf of the ETR, in so-called 'proxy' mode. This behaviour can be selected when the ETR registers with the MR, described immediately below.)

Similarly, the client interface to the indexing system from an ETR's point of view is through LISP nodes called Map-Servers (MSs). ETRs send registration control messages (Map-Register packets) to an MS, which makes the information about the mappings which the ETR indicates it is authoritative for available to the indexing sub-system.

The MS formulates a reply control message (the Map-Notify packet), which confirms the registration, and is returned to the ETR. The details of the indexing sub-system are thus likewise hidden from the 'ordinary' ETRs.

The fact that the details of the indexing sub-system are entirely hidden from xTRs gives considerably flexibility to this aspect of LISP. As long as any potential indexing sub-system can track where mappings are, it could potentially be used; this would allow the actual indexing sub-system to be replaced without needing to modify the clients - as has happened once already (see below).

8.2.3. Indexing Sub-system

The current indexing sub-system is the Delegated Database Tree (DDT), which is very similar to DNS ([DDT], [RFC1034]). Unlike DNS, the actual mappings are not handled by DDT; DDT, as the indexing sub-system, merely identifies the ETRs which hold the actual mappings.

DDT replaces an earlier indexing sub-system, ALT (Appendix B.4, "The ALT Mapping Indexing Sub-system"); this swap validated the concept of having a client-interface sub-system between the indexing sub-system, and the clients.

8.2.3.1. DDT Overview

Conceptually, DDT is fairly simple: like DNS, in DDT the delegation of the EID namespace ([Perspective], Section "Namespaces-XEIDs") is

instantiated as a "delegation hierarchy", a tree of "DDT vertices", starting with the 'root' DDT vertex. Each vertex is responsible for a "block" of the EID namespace.

The 'root' vertex is responsible for the entire namespace; any DDT vertex can 'delegate' part(s) of its block of the namespace to child DDT vertex(s). The child vertex(s) can in turn further delegate (necessarily smaller) blocks of namespace to their children, through as many levels as are needed (for operational, administrative, etc, needs).

Just as with DNS, any particular vertex in the DDT delegation tree may be instantiated in one or more "DDT servers". Multiple (redundant) servers for a given vertex would be used for reasons of performance, reliability and robustness. Obviously, all the servers which instantiate a particular vertex in the tree have to have identical data about that vertex; if they do not, when a Map-Request is sent to one that does not have consistent information with its other sibling(s), incorrect results will be returned.

Also, although the delegation hierarchy is a strict tree, a single DDT server could be authoritative for more than one block of the EID namespace (i.e. it could be a server for more than one vertex).

Eventually, leaf vertices in the delegation hierarchy statically delegate EID namespace blocks to MS's, which are DDT terminal servers; i.e. a leaf of the tree is reached when the delegation points to an MS instead of to another DDT vertex. {{Straighten out.}}

The MS is in direct communication with the ETR(s) which both i) are authoritative for the mappings for that block, and ii) handle traffic to all nodes in that block of EID namespace.

8.2.3.2. Use of DDT by MRs

An MR which wants to find a mapping for a particular EID first interacts with the "DDT servers" which instantiate the "vertices" of the LISP "delegation hierarchy" tree, discovering (by querying the servers for information about DDT vertices) the chain of delegations which cover that EID. Eventually it is directed to an MS, which is the 'door' to an ETR which is authoritative for that EID.

Also, again like DNS, MRs cache information they receive about the delegations in the delegation tree. This means that once an MR has been in operation for while, it will usually have much of the delegation information cached locally (especially the top levels of the delegation tree). This allows them, when passed a request for a mapping by an ITR, to usually forward the mapping request to the appropriate MS without having to interact with all the DDT servers on the path down the delegation tree, in order to find any particular mapping.

Thus, a typical resolution cycle would usually involve looking at some locally cached delegation information, perhaps loading some missing delegation entries into their delegation cache, and finally sending the Map-Request to the appropriate MS.

It should also be noted that the delegation tree is fairly static, since it reflects namespace allocations, which are themselves fairly static. This stability has several important consequences. First, it increases the performance of the mapping system, since the sub-system almost never needs to be re-queried for information about

intermediate vertices. Second, it is not necessary to include a mechanism to find out-dated delegations. [LISP-TREE]

This contrasts with the `_mappings_`, which may change at a high rate - changes which have no impact on the indexing sub-system. LISP is designed to make sure that changes in the mappings are detected and acted upon fairly quickly; this allows LISP to provide a number of capabilities, such as mobility.

9. Examples of Operation

To aid in comprehension, a few examples are given of user packets traversing the LISP system. The first shows the processing of a typical user packet which is LISP forwarded, i.e. what the vast majority of user packets will see. The second shows what happens when the first packet to a previously-unseen ultimate destination (at a particular ITR) is to be processed by LISP.

9.1. An Ordinary Packet's Processing

This case follows the processing of a typical user packet (for instance, a normal TCP data or acknowledgment packet associated with an already-open TCP connection) - i.e. not the first packet sent from a given source to a given destination - as it makes its way from the original source host to the ultimate destination.

When the packet has made its way through the local site to an ITR, which in this case is a border router for the site, the border router looks up the destination address - an EID - in its local "mapping cache". For EIDs which are IPvN addresses, this lookup usually uses the usual IPvN 'longest prefix match' algorithm.

It finds a mapping, which instructs it to wrap the packet in an outer header - an IP packet, containing a UDP packet which contains a LISP header - and then the user's original packet (see Section 12.2, "UDP Encapsulation Details", for the reasons for this particular choice). The destination address in the outer header is set by the ITR to the RLOC of the destination ETR.

The encapsulated packet is then sent off through the Internet, using normal Internet routing.

On arrival at the destination ETR, the ETR will notice that it is listed as the destination in the outer header. It will examine the packet, detect that it is a LISP packet, and unwrap it. It will then examine the header of the user's original packet, and forward it internally, through the local site, to the ultimate destination.

At the ultimate destination, the packet will be processed, and may produce a return packet, which follows the exact same process in reverse - with the exception that the roles of the ITR and ETR are swapped.

9.2. A Mapping Cache Miss

If a host sends a packet, and it gets to the ITR, and the ITR determines that it does not yet have a "mapping cache" entry which covers that destination EID, then additional processing ensues; it has to look up the mapping in the mapping system (as previously described in Section 6.2, "Basic Functionality").

The overall processing is shown below, in Figure 2:

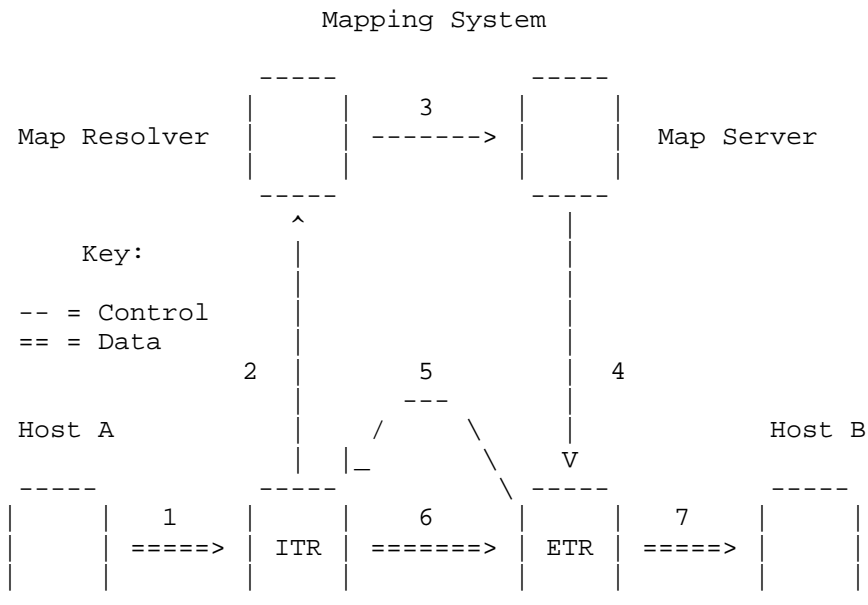


Figure 2: Packet Flow With Missing Mapping

1. Source-EID sends packet (to Dest-EID) to ITR
2. ITR sends Map-Request to Map Resolver
3. Map-Resolver delivers Map-Request to Map-Server
4. Map-Server delivers Map-Request to ETR
5. ETR returns Map-Reply to ITR; ITR caches EID-to-RLLOC(s) mapping
6. ITR uses mapping to encapsulate to ETR; sends user packet to ETR
7. ETR decapsulates packet, delivers to Dest-EID

The ITR first sends a Map-Request packet, giving the destination EID it needs a mapping for, to its MR. The MR will look in its cache of delegation information to find the vertex which is the most specific in the delegation tree for that destination EID. If it does not have the address of an appropriate MS, it will query the DDT system, recursively if need be, in order to eventually find the address of such an MS.

When it has the MS's address, it will send the Map-Request on to the MS, which then usually sends it on to an appropriate ETR. The ETR sends a Map-Reply to the ITR which needs the mapping; from then on, processing of user packets through that ITR to that ultimate destination proceeds as above.

Often the original user packet will have been discarded, and not queued waiting for the mapping to be returned. When the host retransmits such a packet, the mapping will be there, and the packet will be forwarded. Alternatively, it might have been queued, or perhaps it was forwarded using a PITR. (Section 6.4, "Interworking With Non-LISP-Capable Endpoints")

10. Part II

11. Design Approach

Before describing LISP's components in more detail below, it is worth pointing out that what may seem, in some cases, like odd (or poor) design approaches do in fact result from the application of a thought-through, and consistent, design philosophy used in creating

them. {{Subjective: maybe JMH, Dino can help with better words?}}

This design philosophy is covered in detail in in [Perspective], Section "Design"), and readers who are interested in the 'why' of various mechanisms should consult that; reading it may make clearer the reasons for some engineering choices in the mechanisms given here.

12. xTRs

As mentioned above (in Section 8.1, "Data Plane - xTRs Overview"), xTRs are the basic data-handling nodes in LISP, and, as such, form the LISP data plane - although of necessity they are also involved in some control plane functions. This section explores some advanced topics related to xTRs.

Careful rules have been specified for both TTL and ECN [RFC3168] to ensure that passage through xTRs does not interfere with the operation of these mechanisms. In addition, care has been taken to ensure that 'traceroute' works when xTRs are involved.

12.1. When to Encapsulate

An ITR knows that an ultimate destination is 'running' LISP (remember that the actual destination machine itself probably knows nothing about LISP), and thus that it should perform LISP processing on a packet (including potential encapsulation) if it has an entry in its local "mapping cache" that covers the destination EID.

Conversely, if the cache contains a 'negative' entry (indicating that the ITR has previously attempted to find a mapping that covers this EID, and it has been informed by the mapping system that no such mapping exists), it knows the ultimate destination is not running LISP, and the packet can be forwarded natively (i.e. not LISP-encapsulated).

Note that the ITR cannot simply depend on the appearance, or non-appearance, of the destination in the routing tables in the "Internet core", as a way to tell if an ultimate destination is a LISP node or not. That is because mechanisms to allow interoperation of LISP sites and 'legacy' sites necessarily involve advertising LISP sites' EIDs into the Internet core; in other words, LISP sites which need to interoperate with 'legacy' nodes will appear in the Internet core routing tables, along with non-LISP sites.

12.2. UDP Encapsulation Details

Use of UDP (instead of, say, a LISP-specific protocol number) was driven by the fact that many routers filter out 'unknown' protocols, so adopting a non-UDP encapsulation would have made the initial deployment of LISP harder.

The UDP source port in the encapsulated packet is a 5-way hash of the original source and ultimate destination in the inner header, along with the ports, and the protocol.

This is because many ISPs use multiple parallel paths (so-called 'Equal Cost Multi-Path'), and load-share across them. Using such a hash in the source-port in the outer header both allows LISP traffic to be load-shared, and also ensures that packets from individual connections are delivered in order (since most ISPs try to ensure that packets for a particular {source, source port, destination,

destination port} tuple flow along a single path, and do not become disordered).

The UDP checksum is zero because the inner packet usually already has a end-end checksum, and the outer checksum adds no value. [Saltzer] In most existing hardware, computing such a checksum (and checking it at the other end) would also present a major load, for no benefit.

12.3. Header Control Channel

LISP provides a multiplexed channel in the encapsulation header. It is mostly (but not entirely) used for control purposes. (See [Perspective], Section "Architecture-Piggyback" for a longer discussion of the architectural implications of performing control functions with data traffic.)

The general concept is that the header starts with an 'flags' field, and it also includes two data fields, the contents and meaning of which vary, depending on which flags are set. This allows these fields to be multiplexed among a number of different low-duty-cycle functions, while minimizing the space overhead of the LISP encapsulation header.

12.3.1. Mapping Versioning

One important use of the multiplexed control channel is mapping versioning; i.e. the discovery of when the mapping cached in an ITR is outdated. To allow an ITR to discover this, identifying sequence numbers are applied to different versions of a mapping. [RFC6834] This allows an ITR to easily discover when a cached mapping has been updated by a more recent variant.

Version numbers are available in control messages (Map-Replies), but the initial concept is that to limit control message overhead, the versioning mechanism should primarily use the multiplexed user data header control channel.

Versioning can operate in both directions: an ITR can advise an ETR what version of a mapping it is currently using (so the ETR can notify it if there is a more recent version), and ETRs can let ITRs know what the current mapping version is (so the ITRs can request an update, if their copy is outdated).

At the moment version numbers are manually assigned, and ordered.

12.3.2. Echo Nonces

Another important use of the header control channel is for a mechanism known as the Nonce Echo, which is used as an efficient method for ITRs to check the reachability of "neighbour ETRs".

Basically, an ITR which wishes to ensure that an ETR is up, and "reachable", sends a nonce to that ETR, carried in the encapsulation header; when that ETR (acting as an ITR) sends some other user data packet back to the ITR (acting in turn as an ETR), that nonce is carried in the header of that packet, allowing the original ITR to confirm that its packets are reaching that ETR.

Note that a lack of a response is not necessarily proof that something has gone wrong - but it strongly suggests that something has, so other actions (e.g. a switch to an alternative ETR, if one is listed; a direct probe; etc) are advised.

(See Section 16.5, "Verifying ETR Reachability", for more about Echo Nonces.)

12.3.3. Instances

Another use of these header fields is for 'Instances' - basically, support for VPN's across backbones. [RFC4026] Since there is only one destination UDP port used for carriage of user data packets, and the source port is used for multiplexing (above), there is no other way to differentiate among different destination address namespaces (which are often overlapped in VPNs).

12.4. Probing

RLOC-Probing (see [RFC6830], Section 6.3.2. "RLOC-Probing Algorithm" for details) is a mechanism method that an ITR can use to determine with certainty that an ETR is up and reachable from the ITR. As a side-benefit, it gives a rough RTT estimates.

It is quite a simple mechanism - an ITR simply sends a specially marked Map-Request directly to the ETR it wishes information about; that ETR sends back a specially marked Map-Reply. A Map-Request and Map-Reply are used, rather than a special probing control-message pair, because as a side-benefit the ITR can discover if the mapping has been updated since it cached it.

The probing mechanism is rather heavy-weight and expensive (compared to mechanisms like the Echo-Nonce), since it costs a control message from each side, so it should only be used sparingly. However, it has the advantages of providing information quickly (a single RTT), and being a simple, direct, robust way of doing so.

If the number of active neighbour ETRs of the ITR is large, use of RLOC-Probing to check on their reachability will result in considerable control traffic; such control traffic has to be spread out to prevent a load peak.

Obviously, if RLOC-Probing is the only mechanism being used to detect unreachable neighbour ETRs, the rate at which RLOC-Probing is done will control the timeliness of the detection of loss of reachability. There is thus a tradeoff between overhead and responsiveness, particular when an ITR has a large fanout of neighbour ETRs.

A further observation is that unless what are likely unreasonable amounts of RLOC Probing are being done, Echo Nonce will generally provide faster notification of loss of reachability (unless there is little or no bi-directional traffic between the ITR and ETR). {{ENS help reduce the amount of probing when both are in use}}

12.5. Mapping Lifetimes and Timeouts

Mappings come with a Time-To-Live, which indicate how long the creator of the mapping expects them to be useful for. The TTL may also indicate that the mapping should not be cached at all, or it can indicate that it has no particular lifetime, and the recipient can chose how long to store it.

Mappings might also be discarded before the TTL expires, depending on what strategies the ITR is using to maintain its cache; if the maximum cache size is fixed, or the ITR needs to reclaim memory, mappings which have not been used 'recently' may be discarded.

(After all, there is no harm in so doing; a future reference will merely cause that mapping to be reloaded.)

{{Contents may change before TTL expires?}}

12.6. Mapping Gleaning in ETRs

As an optimization to the mapping acquisition process, ETRs are allowed to 'glean' mappings from incoming user data packets, and also from incoming Map-Request control messages. This is not secure, and so any such mapping must be 'verified' by sending a Map-Request to get an authoritative mapping. (See further discussion of the security implications of this in [Perspective], Section "Security-xTRs".)

The value of gleaning is that most communications are two-way, and so if host A is sending packets to host B (therefore needing B's EID->RLOC mapping), very likely B will soon be sending packets back to A (and thus needing A's EID->RLOC mapping). Without gleaning, this would sometimes result in a delay, and the dropping of the first return packet; this is felt to be very undesirable.

12.7. MTU Issues

Several mechanisms have been proposed for dealing with packets which are too large to transit the path from a particular ITR to a given ETR.

In one, called the 'stateful' approach, the ITR keeps a per-ETR record of the maximum size allowed, and sends an ICMP Too Big message to the original source host when a packet which is too large is seen.

In the other, referred to as the 'stateless' approach, for IPv4 packets without the 'DF' bit set, too-large packets are fragmented, and then the fragments are forwarded; all other packets are discarded, and an ICMP Too Big message returned.

12.8. Security of Mapping Lookups

LISP provides an optional mechanism to secure the obtaining of mappings by an ITR. [LISP-SEC] It provides protection against attackers generating spurious Map-Reply messages (including replaying old Map-Replies), and also against 'over-claiming' attacks (where a malicious ETR by claims EID-prefixes which are larger than what have been actually delegated to it).

In summary, the ITR provides a One-Time Key with its Map-Request; this key is used by both the MS (to sign an affirmation that it has delegated that EID block to that ETR), and indirectly by the ETR (to sign the mapping that it is returning to the ITR).

The specification for LISP-SEC suggests that the ITR-MR stage be cryptographically protected, and indicates that the existing mechanisms for securing the ETR-MS stage are used to protect Map-Requests also. It does assume that the channel from the MR to the MS is secure (otherwise an attacker could obtain the OTK from the Map-Request and use it to forge a reply).

12.9. xTR Mapping Cache Performance

As mentioned previously (Section 8.1.1 "Mapping Cache Performance"), a substantial amount of simulation work has been performed to

predict, and understand, the performance of the "mapping cache" in xTRs.

For a comprehensive survey of this work, see [Perspective], Section "Mapping Cache Performance", and the references; full details are too lengthy to include here.

Briefly, however, the first, [Iannone], was performed in the very early stages of the LISP effort, to verify that that caching approach was feasible.

Packet traces of all traffic over the external connection of a large university over a week-long period were collected; simulations driven by these recording were then performed. A variety of control settings on the cache were used, to study the effects of varying the settings.

First, the simulation gave the cache sizes that would result from such a cache design: it showed that the resulting cache sizes ranged from 7,500 entries, up to about 100,000 (depending on factors such as traffic and entry retention time). Using some estimations as to how much memory mapping entries would use, this indicated cache sizes of between roughly 100 Kbytes and a few Mbytes.

Of more interest, in a way, were the results regarding two important measurements of the effectiveness of the cache: i) the hit ratio (i.e. the share of references which could be satisfied by the cache), and ii) the miss_rate_ (since control traffic overhead is one of the chief concerns when using a cache). These results were also encouraging: miss (and hence lookup) rates ranged from 30 per minute, up to 3,000 per minute.

Significantly, this was substantially lower than the amount of observed DNS traffic, which ranged from 1,800 packets per minute up to 15,000 per minute. The results overall showed that using a demand-loaded cache was an entirely plausible design approach: both cache size, and the control plane traffic load, were definitely feasible.

The second, [Kim], was in general terms similar, except that it used data from a large ISP, one with about three times as many users as the previous study. It used the same cache design philosophy (the cache size was not fixed), but slightly different, lower, retention time values.

The results were similar: cache sizes ranges from 20,000 entries to roughly 60,000; the miss rate ranged from very roughly 400 per minute to very roughly 7,000 per minute, similar to the previous results.

Finally, a third study, [CorasCache], examined the effect of using a fixed size cache, and a purely Least Recently Used (LRU) cache eviction algorithm (i.e. no timeouts). It also tried to verify that models of the performance of such a cache (using previous theoretical work on caches) produced results that conformed with actual empirical measurements.

It used yet another set of packet traces; using a cache size of around 50,000 entries produced a miss rate of around 1×10^{-4} ; again, definitely viable, and in line with the results of the other studies.

As discussed already in Section 8.2, "Control Plane - Mapping System Overview", the LISP "mapping system" is an important part of LISP's control plane: it i) maintains the database of "mappings" between EIDs, and the RLOCs at which they are to be found, and ii) provides those mappings to ITRs which request them, so that the ITRs can send traffic for a given EID to the correct RLOC(s) for that EID.

RFC 1034 ("DNS Concepts and Facilities") has this to say about the DNS name to IP address database and mapping system:

"The sheer size of the database and frequency of updates suggest that it must be maintained in a distributed manner, with local caching to improve performance. Approaches that attempt to collect a consistent copy of the entire database will become more and more expensive and difficult, and hence should be avoided."

and this observation applies equally to the LISP mapping database and mapping system.

To briefly recap, the mapping system is split into three parts: i) an "indexing sub-system", which keeps track of where all the mappings are kept; ii) the interface to the indexing system (which remains the same, even if the actual indexing system is changed); and iii) the mappings themselves (collectively, the "mapping database"), the authoritative copies of which are always held by ETRs.

13.1. The Mapping System Interface

As mentioned in Section 8.2.2, "Interface to the Mapping System", both of the interfaces to the mapping system (from ITRs, and ETRs) are standardized, so that the more numerous xTRs do not have to be modified when the mapping indexing sub-system is changed.

(This precaution has already allowed the mapping system to be upgraded during LISP's evolution, when ALT was replaced by DDT.)

This section describes the interfaces in a little more detail; for details, see [RFC6833].

13.1.1. Map-Request Messages

The Map-Request message contains a number of fields, the two most important of which are the requested EID block identifier (remember that individual mappings may cover a block of EIDs, not just a single EID), and the Address Family Identifier (AFI) for that EID block.

Other important fields are the source EID (and its AFI), and one or more RLOCs for the source EID, along with their AFIs. {{Not quite right, Dino will clarify. - Also two sets of RLOCs.}} Multiple RLOCs are included to ensure that at least one is in a form which will allow the reply to be returned to the requesting ITR, and the source EID is used for a variety of functions, including 'gleaning' (see Section 12.6, " Mapping Gleaning in ETRs").

Finally, the message includes a long nonce, for simple, efficient protection against offpath attackers (see [Perspective], Section "Security-xTRs" for more), and a variety of other fields and control flag bits.

13.1.2. Map-Reply Messages

The Map-Reply message looks similar, except it includes the mapping

entry for the requested EID(s), which contains one or more RLOCs and their associated data. (Note that the reply may cover a larger block of the EID namespace than the request; most requests will be for a single EID, the one which prompted the query.)

If there are no mappings available at all for the EID(s) requested, a 'Negative Map-Reply' message will be returned. This is a Map-Reply message with flag bits set to indicate that fact.

For each RLOC in the entry, there is the RLOC, its AFI, priority and weight fields (see Section 8.2, "Control Plane - Mapping System Overview"), and multicast priority and weight fields (see Section 14, "Multicast Support in LISP" for more about multicast support in LISP).

13.1.2.1. Solicit-Map-Request Messages

"Solicit-Map-Request" (SMR) messages are actually not another message type, but a variant of Map-Request messages. {{Look at how probe is handled, do similar here - take out 'not xxx', say what they are.}} They include a special flag which indicates to the recipient that it should send a new Map-Request message, to refresh its mapping, because the ETR has detected that the one it is using is out-dated.

SMR's, like most other control traffic, is rate-limited.

13.1.3. Map-Register and Map-Notify Messages

The Map-Register message contains authentication information, and a number of mapping records, each with an individual Time-To-Live (TTL). Each of the records contains an EID (potentially, a block of EIDs) and its AFI, a version number for this mapping (see Section 12.3.1, "Mapping Versioning"), and a number of RLOCs and their AFIs.

Each RLOC entry also includes the same data as in the Map-Replies (i.e. priority and weight); this is because in some circumstances it is advantageous to allow the MS to proxy reply on the ETR's behalf to Map-Request messages, and the MS needs this information when it does so (see [Mobility]).

Map-Notify messages have the exact same contents as Map-Register messages; they are purely acknowledgements (although planned LISP functionality extensions may give them other functions as well).

The entire interaction can be authenticated by use of a shared key, configured in the MS and ETR. Although the protocol does already allow for replacement of the encryption algorithm, it does not support automated key management (although it appears to fall under the exclusions in [RFC4107]).

{{Deregistering??}}

13.2. The DDT Indexing Sub-system

As previously mentioned in Section 8.2.3, "Indexing Sub-system", the "indexing sub-system" in LISP is currently the DDT system.

The overall functioning is conceptually fairly simple; an MR which needs a "mapping" starts at a server for the root "DDT vertex" (there will normally be more than one such server available, for both performance and robustness reasons), and through a combination of

cached delegation information, and repetitive querying of a sequence of DDT servers, works its way down the delegation tree until it arrives at an MS which is authoritative (responsible?) for the block of EID namespace which holds the destination EID in question.

The interaction between MRs and DDT servers is as follow. The MR sends to the DDT server a Map-Request control message. The DDT server uses its data (which is configured, and static) to see whether it is directly peered to an MS which can answer the request, or if it has a child (or children, if replicated) which is responsible for that portion of the EID namespace.

If it has children configured which are responsible, it will reply to the MR with another kind of LISP control message, a Map-Referral message, which provides information about the delegation of the block containing the requested EID. This step is secured; see Section 13.4, "Security of the DDT Indexing Sub-system", for more.

The Map-Referral also gives the addresses of DDT servers for that block. and the MR can then send Map-Requests to any one (or all) of them. In addition, the Map-Referral includes keying material for the children, which allows any information provided by them to be cryptographically verified.

Control flags in the Map-Referral indicate to the querying MR whether the referral is to another DDT server, an MS, or an ETR. {{All three? Check}} If the former, the MR then sends the Map-Request to the child DDT server, repeating the process.

If the second, the MR then interacts with that MS, and usually the block's ETR(s) as well, to cause a mapping to be sent to the ITR which queried the MR for it. (Recall that some MS's provide Map-Replies on behalf of an associated ETR, in so-called 'proxy mode', so in such cases the Map-Reply will come from the MS, not the ETR.)

Delegations are cached in the MRs, so that once an MR has received information about a delegation, it usually will not need to look that up again. Once it has been in operation for a short while, there will usually only be a limited amount of delegation information which is has not yet asked about - probably only the last stage in a delegation to a 'leaf' MS.

As describe below (Section 13.6, "Performance of the Mapping System"), an extensive modeling and performance evaluation has verified that DDT provides acceptable performance, as well as scalability. [LISP-TREE]

13.2.1. Map-Referral Messages

Map-Referral messages look almost identical to Map-Reply messages, except that the RLOCs potentially name either i) the DDT servers for other DDT vertices (children in the delegation tree), or ii) terminal MSs.

13.3. Reliability via Replication

Everywhere throughout the mapping system, robustness to operational failures is obtained by replicating data in multiple instances of any particular node (of whatever type). Map-Resolvers, Map-Servers, DDT nodes, ETRs - all of them can be replicated, and the protocol supports this replication.

{{About replication - we don't talk about how that rep occurs}}
{{Reliability through rep is much sturdier - provide good ref}}

There are generally no mechanisms specified yet to ensure coherence between multiple copies of any particular data item (e.g. the copies of delegation data for a particular block of namespace, in DDT sibling servers) - this is currently a manual responsibility.

If and when LISP protocol adoption proceeds, an automated layer to perform this functionality can 'easily' be layered on top of the existing mechanisms.

The deployed DDT system actually uses anycast [RFC4786], along with replicated servers, to improve both performance and robustness. {{Not just DDT, other places as well.}}

13.4. Security of the DDT Indexing Sub-system

In summary, securing the mapping indexing system is divided into two parts: the interface between the clients of the system (MR's) and the mapping indexing system itself, and the interaction between the DDT servers which make it up.

The client interface provides only a single model, using the 'canonical' public-private key system (starting from a trust anchor), in which the child's public key is provided by the parent, along with the delegation. When the child returns any data, it can sign the data, and the requestor can use that signature to verify the data. This requires very little configuration in the clients.

The interface between the DDT servers allows for choices between a number of different options, allowing the operators to trade off among configuration complexity, security level, etc. This is based on experience with DNSSEC ([RFC4033]), where configuration complexity has been a major stumbling block to deployment.

See [Perspective], Section "Security-Mappings" for more.

13.5. Extended Capabilities

In addition to the priority and weight data items in mappings, LISP offers other tools to enhance functionality, particularly in the traffic engineering area.

One is 'requestor-specific mappings', i.e. the ETR may return different mappings to the enquiring ITR, depending on the identity of the ITR. This allows very fine-tuned traffic engineering, far more powerful than routing-based TE. {{Policy-based?}}

13.6. Performance of the Mapping System

Prior to the creation of DDT, a large study of the performance of the previous mapping system, ALT ([ALT]), along with a proposed new design called TREE (which used DNS to hold delegation information) provided considerable insight into the likely performance of the mapping systems at larger scale. (See [LISP-TREE], in particular Section V, "Mapping System Comparison".)

The basic structure and concepts of DDT are identical to those of TREE, so the performance simulation work done for that design applies equally to DDT.

In that study, as with earlier LISP performance analyses, extensive large-scale simulations were driven by lengthy recordings of actual traffic at several major sites; one was the site in the first study ([Iannone]), and the other was an even large university, with roughly 35,000 users.

The results showed that a system like DDT, which caches information about delegations, and allows the MR to communicate directly with the servers for the lower vertices on the delegation hierarchy based on cached delegation information, would have good performance, with average resolution times on the order of the MR to MS RTT. This verified the effectiveness of this particular type of indexing system.

A more recent study, [Saucez], has measured actual resolution times in the deployed LISP network; it took measurements from a variety of locations in the Internet, with respect to a number of different target EIDs. Average measured resolution delays ranged from roughly 175 msec to 225 msec, depending on the location.

14. Multicast Support in LISP

Multicast ([RFC3170], [RFC5110]) , since LISP is all about separating identity from location, and although a multicast group in some sense has an identity, it certainly does not have a location.

{{Say something about sources.}}

Multicast is an important requirement, for a number of reasons: doing multiple unicast streams is inefficient, as it is easy to use up all the upstream bandwidth; without multicast a server can also be saturated fairly easily in doing the unicast replication; etc.

Since it is important for LISP to work well with multicast; doing so has been a significant focus in LISP throughout its entire development.

Further very significant improvements to multicast support in LISP are in progress; see [Improvements], Section "Multicast" for more on them.

14.1. Basic Concepts of Multicast Support in LISP

This section introduces some of the basic principles of multicast support in LISP.

Since group addresses name distributed collective entities, in general they cannot have a single RLOC (although they may, after future improvements in multicast support in LISP, have multiple RLOCs); also, since they usually refer to collections of entities, they aren't really EIDs either.

A multicast source at a LISP site may not be able to become the root of a distribution tree in the core if it uses its EID as its identity for that distribution tree (i.e. a distribution tree (S-EID, G)); that is because there may not be a route to its EID in the core (assuming that its section of the core even supports multicast; not all parts of the core do).

Therefore, outside the LISP site, multicast state for the distribution tree (S-RLOC, G) needs to be built instead, where S-RLOC is the RLOC of the ITR that the multicast source inside the LISP site

will be sending its traffic through.

Multicast LISP requires no packet format changes to existing multicast packets (both control, and user data). The initial multicast support in LISP uses existing multicast control mechanisms exclusively; improvements currently being worked on provide LISP-specific control mechanisms (see [Improvements], Section "Multicast", for more).

14.2. Initial Multicast Support in LISP

Readers who wish to fully understand multicast support need to consult the appropriate specifications: LISP multicast issues are discussed in [RFC6830], Section 11; and see [RFC6831] for the full details of current multicast support in LISP.

In the current simple operating mode (covered in [RFC6831]), destination group addresses are not mapped; only the source address (when the original source is inside a LISP site) needs to be mapped, both during distribution tree setup, as well as actual traffic delivery.

In other words, while LISP's mapping capability is used, at this stage it is only applied to the source, not the destination (as with most LISP activity). Thus, in LISP-encapsulated multicast packets in this mode, the inner source is the EID, and the outer source is the ITR's RLOC; both inner and outer destinations are the group's multicast address.

Note that this does mean that if the group is using separate source-specific trees for distribution, there isn't a separate distribution tree outside the LISP site for each different source of traffic to the group from inside the LISP site; they are all lumped together under a single source, the RLOC.

The issue of encapsulation is complex, because if the rest of the group outside the LISP site includes some members which are at other LISP sites (i.e. packets to them have to be encapsulated), and some members at legacy sites (i.e. encapsulated packets would not be understood), there is no simple answer. (The situation becomes even more complex when one considers that as hosts leave and join the group, it may switch back and forth between 'mixed' and 'homogenous'.)

This issue is too complex to fully cover here; see Section 9.2., "LISP Sites with Mixed Address Families", in [RFC6831], for complete coverage of this issue.

Basically, there are multicast equivalents of some of the legacy interoperability mechanisms used for unicast; mPITRs and mPETRs (multicast-capable PITRs and PETRs) etc. When 'mixed' groups are a possibility, two choices are available: i) send two copies (one encapsulated, and one not) of all traffic, or ii) employ mPETRs to distribute non-encapsulated copies to 'legacy' group members.

15. Deployment Issues and Mechanisms

This section discusses several deployment issues in more detail. With LISP's heavy emphasis on practicality, much work has gone into making sure it works well in the real-world environments most people have to deal with.

15.1. LISP Deployment Needs

As mentioned earlier (Section 5.2, "Maximize Re-use of Existing Mechanism"), LISP requires no change to almost all existing hosts and routers. Obviously, however, one must deploy `_something_` to run LISP! Exactly what that has to be will depend greatly on the details of the site's existing networking gear, and choices it makes for how to achieve LISP deployment.

The primary requirement is for one or more xTRs. These may be existing routers, just with new software loads, or it may require the deployment of new devices.

LISP also requires a certain amount of LISP-specific support infrastructure, such as MRs, MSs, the DDT hierarchy, etc. However, much of this will either i) `{{for the case where you are adding a new site using existing LISP infrastructure}}` already be deployed, and if the new site can make arrangements to use it, it need do nothing else; or ii) those functions the site must provide may be co-located in other LISP devices (again, either new devices, or new software on existing ones).

15.2. Interworking Mechanisms

One aspect which has received a lot of attention are the mechanisms previously referred to (in Section 6.4, "Interworking With Non-LISP-Capable Endpoints") to allow interoperation of LISP sites with so-called 'legacy' sites which are not running LISP (yet).

There are two main approaches to such interworking: proxy routers (PITRs and PETRs), and an alternative mechanism using a router with combined NAT and LISP functionality; these are described in more detail here.

15.2.1. Proxy LISP Routers

PITRs (proxy ITRs) serve as ITRs for traffic `_from_` legacy hosts to nodes in LISP sites. PETRs (proxy ETRs) serve as ETRs for LISP traffic `_to_` legacy hosts (for cases where a LISP node cannot send packets directly to such hosts, without encapsulation).

Note that return traffic `_to_` a legacy host from a LISP-using node does not necessarily have to pass through an ITR/PETR pair - the original packets can usually just be sent directly to the ultimate destination. However, for some kinds of LISP operation (e.g. mobile nodes), this is not possible; in these situations, the PETR is needed.

15.2.1.1. PITRs

To serve as ITRs for traffic `_from_` legacy hosts to nodes in LISP sites, PITRs they have to advertise into the existing legacy backbone Internet routing the availability of whatever ranges of EIDs (i.e. of nodes using LISP) they are proxying for, so that legacy hosts will know where to send traffic to those LISP nodes.

This technique obviously has an impact on routing table in the "Internet core", but it is not clear yet exactly what that impact will be; it is very dependent on the collected details of many individual deployment decisions. `{{Check on text elsewhere for effects on routing table size, specifically advertizement of large blocks.}}`

A PITR may cover a group of EID blocks with a single EID advertisement to the core, in order to reduce the number of routing table entries added. (In fact, at the moment, aggressive aggregation of EID announcements is performed, precisely to minimize the number of new announced routes added by this technique.) {{BGP tools can be used to restrict the direction and scope of these advertisements.}}

At the same time, if a site does traffic engineering with LISP instead of fine-grained BGP announcement, that will help keep table sizes down (and this is true even in the early stages of LISP deployment). The same is true for multi-homing. {{Maybe mixing two concepts? LISP TE tools will still apply to traffic between PITR and LISP site.}}

{{Maybe reword, as we changed the target section.}} As mentioned previously (Section 12.1, "When to Encapsulate"), an ITR at another LISP site can avoid using a PITR (i.e. it can detect that a given ultimate destination is not a legacy host, if a PITR is advertising it into the "Internet core") by checking to see if a LISP mapping exists for that ultimate destination.

15.2.1.2. PETRs

PETRs (proxy ETRs) serve as ETRs for LISP traffic to legacy hosts, for cases where a LISP node cannot send packets to such hosts without encapsulation. That typically happens for one of two reasons.

First, it will happen in places where some device is implementing Unicast Reverse Path Forwarding (uRPF), to prevent a variety of negative behaviour; originating packets with the original source's EID in the source address field will result in them being filtered out and discarded.

Second, it will happen when a LISP site wishes to send packets to a non-LISP site, and the path in between does not support the particular IP protocol version used by the original source along its entire length. Use of a PETR on the other side of the 'gap' will allow the LISP site's packet to 'hop over' the gap, by utilizing LISP's built-in support for mixed protocol encapsulation.

PETRs are generally used by specific ITRs, which have the location of their PETRs configured into them. In other words, unlike normal ETRs, PETRs do not have to register themselves in the mapping database, on behalf of any legacy sites they serve.

Also, allowing an ITR to always send traffic leaving a site to a PETR does avoid having to choose whether or not to encapsulate packets; it can just always encapsulate packets, sending them to the PETR if it has no specific mapping for the ultimate destination. However, this is not advised: as mentioned, it is easy to tell if something is a legacy destination.

15.2.2. LISP-NAT

A LISP-NAT router, as previously mentioned, combines LISP and NAT functionality, in order to allow a LISP site which is internally using addresses which cannot be globally routed to communicate with non-LISP sites elsewhere in the Internet. (In other words, the technique used by the PITR approach simply cannot be used in this case.)

To do this, a LISP-NAT performs the usual NAT functionality, and translates a host's source address(es) in packets passing through it from an 'inner' value to an 'outer' value, and storing that translation in a table, which it can use to similarly process subsequent packets (both outgoing and incoming). [RFC6832]

There are two main cases where this might apply:

- Sites using non-routable global addresses
- Sites using private addresses [RFC1918]

15.3. Use Through NAT Devices

NATs are both ubiquitous, and here to stay for a long time to come. [RFC1631] Thus, in the actual Internet of today, having any new mechanisms function well in the presence of NATs (i.e. with LISP xTRs behind a NAT device) is absolutely necessary.

LISP has produced a variety of mechanisms to do this. An experimental mechanism to support them had major limitations; it, and its limitations, are described in Appendix B.5, "Early NAT Support". A more recent proposed mechanism, which avoids those limitations, is described in [Improvements], Section "Improved NAT Support".

15.4. LISP and Core Internet Routing

One of LISP's original motivations was to try and control the growth of the size of routing tables in the Internet core, the part where routes to all destinations must be available. As LISP becomes more widely deployed, it can help with this issue, in a variety of ways. {{Give ref for why large rout tables bad.}}

{{Does applications make forward ref to this section?}}

In covering this topic, one must recognize that conditions in various stages of LISP deployment (in terms of ubiquity) will have a large influence. [Deployment] introduced useful terminology for this progression, in addition to some coverage of the topic (see Section 5, "Migration to LISP"):

The loosely defined terms of "early transition phase", "late transition phase", and "LISP Internet phase" refer to time periods when LISP sites are a minority, a majority, or represent all edge networks respectively.

In the early phases of deployment, two primary effects will allow LISP to have a positive impact on the routing table growth:

- Using LISP for traffic engineering instead of BGP
- Aggregation of smaller PI sites into a single PIR advertisement

The first is fairly obvious (doing TE with BGP requires injecting more-specific routes into the "Internet core" routing tables, something doing TE with LISP avoids); the second is not guaranteed to happen (since it requires coordination among a number of different parties), and only time will tell if it does happen.

{{Add xref to text moved to "Improvements" document.}}

16. Fault Discovery/Handling

The structure of LISP gives rise to a moderate number of of failure

modes.

16.1. Handling Missing Mappings

To handling missing mappings, the ITR calls for the mapping, and in the meantime can either discard traffic to that ultimate destination (as many ARP implementations do) [RFC826], or, if dropping the traffic is deemed undesirable, it can forward them via a PITR.

16.2. Outdated Mappings

If a mapping changes once an ITR has retrieved it, that may result in traffic to the EIDs covered by that mapping failing. There are three cases to consider:

- When the ETR to which traffic is being sent is still a valid ETR for that EID, but the mapping has been updated (e.g. to change the priority of various ETRs)
- When the ETR traffic is being sent to is still an ETR, but no longer a valid ETR for that EID
- When the ETR traffic is being sent to is no longer an ETR
- {{No longer an ETR, but still a LISP node - another case to consider.}}

16.2.1. Outdated Mappings - Updated Mapping

A 'mapping versioning' system, whereby mappings have version numbers, and ITRs are notified when their mapping is out of date, has been added to detect this, and the ITR responds by refreshing the mapping. [RFC6834]

16.2.2. Outdated Mappings - Wrong ETR

If an ITR is holding an outdated cached mapping, it may send packets to an ETR which is no longer an ETR for that EID.

It might be argued that if the ETR is properly managing the lifetimes on its mapping entries, this 'cannot happen', but it is a wise design methodology to assume that 'cannot happen' events will in fact happen (as they do, due to software errors, or, on rare occasions, hardware faults), and ensure that the system will handle them properly (if, perhaps not in the most expeditious, or 'clean' way - they are, after all, very unlikely to happen). {{Make less run on, easier to understand.}}

ETRs can easily detect cases where this happens, after they have unwrapped a user data packet; in response, they send a Solicit-Map-Request to the source ITR to cause it to refresh its mapping.

16.2.3. Outdated Mappings - No Longer an ETR

In another case for what can happen if an ITR uses an outdated mapping, the destination of traffic from an ITR might no longer be a LISP node at all. In such cases, one might get an ICMP Destination Unreachable (Port Unreachable subtype) error message. However, one cannot depend on that - and in any event, that would provide an attack vector, so it should be used with care. (See [RFC6830], Section 6.3, "Routing Locator Reachability" for more about this.)

The following mechanism will work, though. Since the destination is not an ETR, the echoing reachability detection mechanism (see Section 12.3.2, "Echo Nonces") will detect a problem. At that point,

the backstop mechanism, Probing, will kick in. Since the destination is still not an ETR, that will fail, too.

At that point, traffic will be switched to a different ETR, or, if none are available, a reload of the mapping may be initiated.

16.3. Erroneous Mappings

Again, this 'should not happen', but a good system should deal with it. However, in practise, should this happen, it will produce one of the prior two cases (the wrong ETR, or something that is not an ETR), and will be handled as described there.

16.4. Verifying ETR Liveness

The ITR, like all packet switches, needs to detect, and react, when its neighbour ceases operation. As LISP traffic is effectively always uni-directional (from ITR to ETR), this could be somewhat problematic.

Solving a related problem, "neighbour ETR" "reachability" below) subsumes handling this fault mode, however.

Note that the two terms - "liveness" and "reachability" - are not synonymous (although they are often confused). Liveness is a property of a node - it is either up and functioning, or it is not. Reachability is only a property of a particular pair of nodes. {{Really property of path - if only one path, property of pair, otherwise of path.}}

If packets sent from a first node to a second are successfully received at the second, it is 'reachable' from the first. However, the second node may at the very same time not be reachable from some other node. Reachability is always a ordered pairwise property, and of a specified ordered pair.

16.5. Verifying ETR Reachability

A more significant issue than whether a particular ETR is up or not is, as mentioned above, that although the ETR may be up, attached to the network, etc, an issue in the network, between a source ITR, and the ETR, may prevent traffic from the ITR from getting to the ETR. (Perhaps a routing problem, or perhaps some sort of access control setting.)

The one-way nature of LISP traffic makes this situation hard to detect in a way which is economic, robust and fast. Two out of the three are usually not too hard, but all three at the same time - as is highly desirable for this particular issue - are harder.

In line with the LISP design philosophy ([Perspective], Section "Design-Theoretical"), this problem is attacked not with a single mechanism (which would have a hard time meeting all those three goals simultaneously), but with a collection of simpler, cheaper mechanisms, which collectively will usually meet all three.

They are reliance on the underlying routing system (which can of course only reliably provide a negative reachability indication, not a positive one), the echo nonce (which depends on some return traffic from the destination xTR back to the source xTR), and finally direct 'pinging', in the case where no positive echo is returned.

(The last is not the first choice, as due to the large fan-out expected of LISP router, reliance on it as a sole mechanism would produce a fair amount of overhead.)

17. Acknowledgments

The author would like to start by thanking all the members of the core LISP group for their willingness to allow him to add himself to their effort, and for their enthusiasm for whatever assistance he has been able to provide.

He would also like to thank (in alphabetical order) Michiel Blokzijl, Peter Chiappa, Vina Ermagan, Dino Farinacci, Vince Fuller and Vasileios Lakafosis for their review of, and helpful suggestions for, this document. (If I have missed anyone in this list, I apologize most profusely.)

A special thank you goes to Joel Halpern, who almost invariably, when asked, promptly returned comments on intermediate versions of this document. Grateful thanks go also to Darrel Lewis for his help with material on non-Internet uses of LISP, and to Dino Farinacci and Vince Fuller for answering detailed questions about some obscure LISP topics.

A final thanks is due to John Wrocklawski for the author's organizational affiliation, and to Vince Fuller for help with XML. This memo was created using the xml2rfc tool.

I would like to dedicate this document to the memory of my parents, who gave me so much, and whom I can no longer thank in person, as I would have so much liked to be able to.

18. IANA Considerations

This document makes no request of the IANA.

19. Security Considerations

This memo does not define any protocol and therefore creates no new security issues.

20. References

20.1. Normative References

- | | |
|-----------|--|
| [AFI] | IANA, "Address Family Indicators (AFIs)", Address Family Numbers, January 2011, < http://www.iana.org/assignments/address-family-numbers >. |
| [RFC768] | J. Postel, "User Datagram Protocol", RFC 768, August 1980. |
| [RFC791] | J. Postel, "Internet Protocol", RFC 791, September 1981. |
| [RFC2460] | S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998. |
| [RFC6830] | D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, January 2013. |

- [RFC6831] D. Farinacci, D. Meyer, J. Zwiebel, and S. Venaas, "The Locator/ID Separation Protocol (LISP) for Multicast Environments", RFC 6831, January 2013.
- [RFC6832] D. Lewis, D. Meyer, D. Farinacci, and V. Fuller, "Interworking between Locator/ID Separation Protocol (LISP) and Non-LISP Sites", RFC 6832, January 2013.
- [RFC6833] V. Fuller and D. Farinacci, "Locator/ID Separation Protocol (LISP) Map-Server Interface", RFC 6833, January 2013.
- [RFC6834] L. Iannone, D. Saucez, and O. Bonaventure, "Locator/ID Separation Protocol (LISP) Map-Versioning", RFC 6834, January 2013.
- [Perspective] J. N. Chiappa, "An Architectural Perspective on the LISP Location-Identity Separation System", draft-ietf-lisp-perspective-00 (work in progress), February 2013.
- [Improvements] J. N. Chiappa, "An Overview of On-Going Improvements to the LISP Location-Identity Separation System", draft-chiappa-lisp-improvements-00 (work in progress), September 2013.
- [DDT] V. Fuller, D. Lewis, and D. Farinacci, "LISP Delegated Database Tree", draft-ietf-lisp-ddt-01 (work in progress), March 2013.
- [LISP-SEC] F. Maino, V. Ermagan, A. Cabellos-Aparicio, D. Saucez, and O. Bonaventure, "LISP-Security (LISP-SEC)", draft-ietf-lisp-sec-04 (work in progress), October 2012.
- [NAT-Traversal] V. Ermagan, D. Farinacci, D. Lewis, J. Skriver, F. Maino, and C. White, "NAT traversal for LISP", draft-ermagan-lisp-nat-traversal-03 (work in progress), March 2013.
- [Mobility] D. Farinacci, V. Fuller, D. Lewis, and D. Meyer, "LISP Mobility Architecture", draft-meyer-lisp-mn-08 (work in progress), April 2012.
- [Deployment] L. Jakab, A. Cabellos-Aparicio, F. Coras, J. Domingo-Pascual, and D. Lewis, "LISP Network Element Deployment Considerations", draft-ietf-lisp-deployment-09 (work in progress), July 2013.
- [Threats] D. Saucez, L. Iannone, and O. Bonaventure, "LISP Threats Analysis", draft-ietf-lisp-threats-08 (work in progress), October 2013.
- [LCAF] D. Farinacci, D. Meyer, and J. Snijders, "LISP Canonical Address Format (LCAF)", draft-ietf-lisp-lcaf-03 (work in progress), September 2013.
- [LISP-TE] D. Farinacci, P. Lahiri, and M. Kowal, "LISP Traffic Engineering Use-Cases", draft-farinacci-lisp-te-03

(work in progress), July 2013.

20.2. Informative References

- [NIC8246] A. McKenzie and J. Postel, "Host-to-Host Protocol for the ARPANET", NIC 8246, Network Information Center, SRI International, Menlo Park, CA, October 1977.
- [NSAP] International Organization for Standardization, "Information Processing Systems - Open Systems Interconnection - Basic Reference Model", ISO Standard 7489.1984, 1984.
- [IEN19] J. F. Shoch, "Inter-Network Naming, Addressing, and Routing", IEN (Internet Experiment Note) 19, January 1978.
- [RFC826] D. Plummer, "Ethernet Address Resolution Protocol", RFC 826, November 1982.
- [RFC1034] P. V. Mockapetris, "Domain Names - Concepts and Facilities", RFC 1034, November 1987.
- [RFC1498] J. H. Saltzer, "On the Naming and Binding of Network Destinations", RFC 1498, (Originally published in: 'Local Computer Networks', edited by P. Ravasio et al., North-Holland Publishing Company, Amsterdam, 1982, pp. 311-317.), August 1993.
- [RFC1631] K. Egevang and P. Francis, "The IP Network Address Translator (NAT)", RFC 1631, May 1994.
- [RFC1812] F. Baker, "Requirements for IP Version 4 Routers", RFC 1812, June 1995.
- [RFC1918] Y. Rekhter, R. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear, "Address Allocation for Private Internets", RFC 1918, February 1996.
- [RFC1992] I. Castineyra, J. N. Chiappa, and M. Steenstrup, "The Nimrod Routing Architecture", RFC 1992, August 1996.
- [RFC3168] K. Ramakrishnan, S. Floyd, and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.
- [RFC3170] B. Quinn and K. Almeroth, "IP Multicast Applications: Challenges and Solutions", RFC 3170, September 2001.
- [RFC3272] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and Principles of Internet Traffic Engineering", RFC 3272, May 2002.
- [RFC4026] L. Andersson and T. Madsen, "Provider Provisioned Virtual Private Network (VPN) Terminology", RFC 4026, March 2005.
- [RFC4033] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "DNS Security Introduction and

- Requirements", RFC 4033, March 2005.
- [RFC4107] S. Bellovin and R. Housley, "Guidelines for Cryptographic Key Management", RFC 4107, June 2005.
- [RFC4116] J. Abley, K. Lindqvist, E. Davies, B. Black, and V. Gill, "IPv4 Multihoming Practices and Limitations", RFC 4116, July 2005.
- [RFC4786] J. Abley and K. Lindqvist, "Operation of Anycast Services", RFC 4786, December 2006.
- [RFC4984] D. Meyer, L. Zhang, and K. Fall, "Report from the IAB Workshop on Routing and Addressing", RFC 4984, September 2007.
- [RFC5110] P. Savola, "Overview of the Internet Multicast Routing Architecture", RFC 5110, January 2008.
- [RFC5887] B. Carpenter, R. Atkinson, and H. Flinck, "Renumbering Still Needs Work", RFC 5887, May 2010.
- [RFC6115] T. Li, Ed., "Recommendation for a Routing Architecture", RFC 6115, February 2011.
- (Perhaps the most ill-named RFC of all time; it contains nothing that could truly be called a 'routing architecture'.)
- [ALT] V. Fuller, D. Farinacci, D. Meyer, and D. Lewis, "Locator/ID Separation Protocol Alternative Logical Topology (LISP+ALT)", RFC 6836, January 2013.
- [LISP0] D. Farinacci, V. Fuller, and D. Oran, "Locator/ID Separation Protocol (LISP)", draft-farinacci-lisp-00 (work in progress), January 2007.
- [Future] J. N. Chiappa, "Potential Long-Term Developments With the LISP System", draft-chiappa-lisp-evolution-00 (work in progress), October 2012.
- [Baran] P. Baran, "On Distributed Communications Networks", IEEE Transactions on Communications Systems Vol. CS-12 No. 1, pp. 1-9, March 1964.
- [Chiappa] J. N. Chiappa, "Endpoints and Endpoint Names: A Proposed Enhancement to the Internet Architecture", Personal draft (work in progress), 1999, <<http://www.chiappa.net/~jnc/tech/endpoints.txt>>.
- [Clark] D. D. Clark, "The Design Philosophy of the DARPA Internet Protocols", in 'Proceedings of the Symposium on Communications Architectures and Protocols SIGCOMM '88', pp. 106-114, 1988.
- [Saltzer] J. H. Saltzer, D. P. Reed, and D. D. Clark, "End-To-End Arguments in System Design", ACM TOCS, Vol 2, No. 4, pp 277-288, November 1984.
- [Heart] F. E. Heart, R. E. Kahn, S. M. Ornstein, W. R. Crowther, and D. C. Walden, "The Interface

- Message Processor for the ARPA Computer Network",
Proceedings AFIPS 1970 SJCC, Vol. 36, pp. 551-567.
- [Iannone] L. Iannone and O. Bonaventure, "On the Cost of
Caching Locator/ID Mappings", in 'Proceedings of the
3rd International Conference on emerging Networking
EXperiments and Technologies (CoNEXT'07)', ACM, pp.
1-12, December 2007.
- [Kim] J. Kim, L. Iannone, and A. Feldmann, "A Deep Dive
Into the LISP Cache and What ISPs Should Know About
It", in 'Proceedings of the 10th International IFIP
TC 6 Conference on Networking - Volume Part I
(NETWORKING '11)', IFIP, pp. 367-378, May 2011.
- [CorasCache] F. Coras, A. Cabellos-Aparicio, and J. Domingo-
Pascual, "An Analytical Model for the LISP Cache
Size", in 'Proceedings of the 11th International
IFIP TC 6 Networking Conference: Part I', IFIP, pp.
409-420, May 2012.
- [LISP-TREE] L. Jakab, A. Cabellos-Aparicio, F. Coras, D. Saucez,
and O. Bonaventure, "LISP-TREE: A DNS Hierarchy to
Support the LISP Mapping System", in 'IEEE Journal
on Selected Areas in Communications', Vol. 28, No.
8, pp. 1332-1343, October 2010.
- [Saucez] D. Saucez, L. Iannone, and B. Donnet, "A First
Measurement Look at the Deployment and Evolution of
the Locator/ID Separation Protocol", in 'ACM SIGCOMM
Computer Communication Review', Vol. 43 No. 2, pp.
37-43, April 2013.
- [CorasBGP] F. Coras, D. Saucez, L. Jakab, A. Cabellos-Aparicio,
and J. Domingo-Pascual, "Implementing a BGP-free ISP
Core with LISP", in 'Proceedings of the Global
Communications Conference (GlobeCom)', IEEE, pp.
2772-2778, December 2012.
- [Atkinson] R. Atkinson, "Revised draft proposed definitions",
RRG list message, Message-Id: 808E6500-97B4-4107-
8A2F-36BC913BE196@extremenetworks.com, 11 June 2007,
<[http://www.ietf.org/mail-archive/web/ram/current/
msg01470.html](http://www.ietf.org/mail-archive/web/ram/current/msg01470.html)>.
- [Bibliography] J. N. Chiappa (editor), "LISP (Location/Identity
Separation Protocol) Bibliography", Personal
site (work in progress), July 2013, <[http://
www.chiappa.net/~jnc/tech/lisp/LISPbiblio.html](http://www.chiappa.net/~jnc/tech/lisp/LISPbiblio.html)>.

Appendix A. Glossary/Definition of Terms

- EID, Endpoint Identifier: Originally defined as a name for an
"endpoint", one with purely identity semantics, and globally
unique, and with syntax of relatively short fixed length.
[Chiappa] It is used in the LISP work to mean the "identifier" of
a "node"; it is the input to an EID->RLOC lookup in the "mapping
system"; it is usually an "IPvN" "address". The source and
destination addresses of the `_innermost_` header in a LISP packet
are usually EIDs.
- RLOC, Routing Locator: a LISP-specific term meaning the "locator"
associated with an entity identified by an EID; as such, it is

often the output of an EID->RLOC lookup in the "mapping system"; it is usually an "IPvN" address, and of an "ETR". The source and destination addresses of the `_outermost_` header in a LISP packet are usually RLOCs.

- ITR, Ingress Tunnel Router: a "LISP router" at the border of a "LISP site" which takes user packets sent to it from inside the LISP site, encapsulates in a LISP header, and then sends them across the Internet to an "ETR"; in other words, the start of a 'tunnel' from the ITR to an ETR.
- ETR: Egress Tunnel Router: a "LISP router" at the border of a "LISP site" which decapsulates user packets which arrive at it encapsulated in a LISP header, and sends them on towards their ultimate destination; in other words, the end of the 'tunnel' from an "ITR" to the ETR.
- Neighbour ETR: Although an "ITR" and "ETR" may be separated by many actual physical hops, `_at the LISP level_`, they are direct neighbours; so any ETR which an ITR sends traffic to is a 'neighbour ETR' of that ITR.
- xTR: An xTR refers to a "LISP router" which functions both as an "ITR" and an "ETR" (which is typical), when the discussion involves packet flows in both directions through the router, which results in it alternately functioning as an ITR and then as an ETR.
- Reachable; Reachability; Neighbour ETR Reachability: The ability of an "ITR" to be able to send packets to a "neighbour ETR", or the property of an ITR to be able to send such packets.
- Liveness: Whether a LISP "node" of any kind is 'up' and operating, or not; or the property of a LISP node to be in such a state.
- MR, Map Resolver: A LISP "node" to which "ITRs" send requests for "mappings". See Section 8.2.2, "Interface to the Mapping System", for more.
- MS, Map Server: A LISP "node" with which "ETRs" register "mappings", to indicate their availability to handle incoming traffic to the "EIDs" covered in those mappings. See Section 8.2.2, "Interface to the Mapping System" for more.
- Mapping System: The entire ensemble of data and mechanisms which allow clients - usually "ITRs" - to find "mappings" (from EIDs to RLOCs). It includes both the "mapping database", and also everything used to gain access to it - the MRs, the "indexing sub-system", etc. See Section 8.2.1, "Mapping System Organization" for more.
- Mapping Database: The term 'mapping database' refers to the entire collection of {EID->RLOC} "mappings" spread throughout the entire LISP system. It is a subset of the "mapping system". See Section 8.2, "Control Plane - Mapping System Overview", for more.
- Mapping Cache: A collection of copies of {EID->RLOC} "mappings" retained in an ITR; not the entire "mapping database", but just the subset of it that an ITR needs in order to be able to properly handle the user data traffic which is flowing through it.
- Indexing Sub-system: the entire ensemble of data and mechanisms which allows "MRs" to find out which "ETR(s)" hold the mapping for a given "EID" or "EID block". It includes both the data on "namespace" delegations, as well as the nodes which hold that data, and the protocols used to interact with those nodes. See Section 8.2.1, "Mapping System Organization" for more.
- DDT Vertex; Vertex: a node (in the graph theory sense of the term) in the (abstract) LISP namespace "delegation hierarchy".
- DDT Server: an actual machine, which one can send packets to, in the DDT server hierarchy - which is, hopefully, a one-to-one projection of the LISP address "delegation hierarchy" (although of course a single "DDT vertex" may turn into several sibling servers). Some documents refer to these as 'DDT nodes' but this

document does not use that term, to prevent confusion with "DDT vertex".

- PITR: Proxy ITR; an "ITR" which is used for interworking between a LISP-speaking "node" or "site", and legacy nodes or sites; in general, it acts like a normal ITR, but does so on behalf of LISP nodes which are receiving packets from a legacy node. See Section 15.2.1.1, "PITRs", for more.
- PETR: Proxy ETR; an "ETR" which is used for interworking between a LISP-speaking "node" or "site", and legacy nodes or sites; in general, it acts like a normal ETR, but does so on behalf of LISP nodes which are sending packets to a legacy node. See Section 15.2.1.2, "PETRs" for more.
- RTR: Re-encapsulating Tunnel Router; a data plane 'anchor point' used by a LISP-speaking node to perform functions that can only be performed in the core of the network. One use is for LISP-speaking node behind a NAT device to send and receive traffic through the NAT device; see [Improvements], Section "Improved NAT Support" for more.
- Internet core: That part of the Internet in which there are no 'default' entries in routing tables, but where the routing tables hold entries for every single reachable destination in the Internet. (Sometimes referred to colloquially as the 'DFZ', or 'Default Free Zone'.)

Appendix B. Other Appendices

B.1. A Brief History of Location/Identity Separation

It was only gradually realized in the networking community that networks (especially large networks) should deal quite separately with the identity and location of a node; the distinction between the two was more than a little hazy at first.

The ARPANET had no real acknowledgment of the difference between the two. [Heart] [NIC8246] The early Internet also co-mingled the two ([RFC791]), although there was recognition in the early Internet work that there were two different things going on. [IEN19]

This likely resulted not just from lack of insight, but also the fact that extra mechanism is needed to support this separation (and in the early days there were no resources to spare), as well as the lack of need for it in the smaller networks of the time. (It is a truism of system design that small systems can get away with doing two things with one mechanism, in a way that usually will not work when the system gets much larger.)

The ISO protocol architecture took steps in this direction [NSAP], but to the Internet community the necessity of a clear separation was definitively shown by Saltzer. [RFC1498] Later work expanded on Saltzer's, and tied his separation concepts into the fate-sharing concepts of Clark. [Clark], [Chiappa]

The separation of location and identity is a step which has recently been identified by the IRTF as a critically necessary evolutionary architectural step for the Internet. [RFC6115] However, it has taken quite some time for this requirement to be generally accepted by the Internet engineering community at large, although it seems that this may finally be happening.

Unfortunately, although the development of IPv6 presented a golden opportunity to learn from this particular failing of IPv4, that design failed to recognize the need for separation of location and

identity.

B.2. A Brief History of the LISP Project

The LISP system for separation of location and identity resulted from the discussions of this topic at the Amsterdam IAB Routing and Addressing Workshop, which took place in October 2006. [RFC4984]

A small group of like-minded personnel from various scattered locations within Cisco, spontaneously formed immediately after that workshop, to work on an idea that came out of informal discussions at the workshop. The first Internet-Draft on LISP appeared in January, 2007, along with a LISP mailing list at the IETF. [LISP0]

Trial implementations started at that time, with initial trial deployments underway since June 2007; the results of early experience have been fed back into the design in a continuous, ongoing process over several years. LISP at this point represents a moderately mature system, having undergone a long organic series of changes and updates.

LISP transitioned from an IRTF activity to an IETF WG in March 2009, and after numerous revisions, the basic specifications moved to becoming RFCs at the start of 2013 (although work to expand and improve it, and find new uses for it, continues, and undoubtedly will for a long time to come).

B.3. Old LISP 'Models'

LISP, as initially conceived, had a number of potential operating modes, named 'models'. Although they are now obsolete, one occasionally sees mention of them, so they are briefly described here.

- LISP 1: EIDs all appear in the normal routing and forwarding tables of the network (i.e. they are 'routable'); this property is used to 'bootstrap' operation, by using this to load EID->RLOC mappings. Packets were sent with the EID as the destination in the outer wrapper; when an ETR saw such a packet, it would send a Map-Reply to the source ITR, giving the full mapping.
- LISP 1.5: Similar to LISP 1, but the routability of EIDs happens on a separate network.
- LISP 2: EIDs are not routable; EID->RLOC mappings are available from the DNS.
- LISP 3: EIDs are not routable; and have to be looked up in a new EID->RLOC mapping database (in the initial concept, a system using Distributed Hash Tables). Two variants were possible: a 'push' system, in which all mappings were distributed to all ITRs, and a 'pull' system in which ITRs load the mappings they need, as needed.

B.4. The ALT Mapping Indexing Sub-system

LISP initially used an indexing sub-system called ALT. [ALT] ALT repurposed a number of existing mechanisms to provide an indexing system, which allowed an experimental LISP initial deployment to become operational without having to write a lot of code, ALT was relatively easily constructed from basically unmodified existing mechanisms; it used BGP running over virtual tunnels using GRE.

ALT proved to have a number of issues which made it unsuitable for large-scale use, and it has now been superseded by DDT. A complete

list of these is not possible here, but the issues mostly were of two kinds: technical issues which would have arisen at large scale, and practical operational issues which appeared even in the experimental deployment.

The biggest operational issues was the effort involved in configuring, and maintain the configuration, of the virtual tunnels over which ALT ran (including assigning the addresses for the ends, etc); also, managing the multiple disjoint routing tables required was difficult and confusing (even for those who were very familiar with ALT). Debugging faults in ALT was also difficult; and finally, because of ALT's nature, administrative issues (who pays for what, who controls what, etc) were problematic.

However, ALT would have had significant technical issues had it been used at a larger scale.

The most severe (and fundamental) issue was that since all traffic on the ALT had to transit the 'root' of the ALT tree, those locations would have become traffic 'hot-spots' in a large scale deployment.

In addition, optimal performance would have required that the ALT overall topology be restrained to follow the EID namespace allocation; however, it was not clear that this was feasible. In any event, even optimal performance was still less than that in alternatives. The ALT was also very vulnerable to misconfiguration.

See [LISP-TREE] for more about these issues: the basic structure and operation of DDT is identical to that of TREE, so the conclusions drawn there about TREE's superiority to ALT apply equally to DDT.

In particular, the big advantage of DDT over the ALT, in performance terms, is that it allows MRs to interact directly with distant DDT servers (as opposed to the ALT, which always required mediation through intermediate servers); caching of information about those distant servers allows DDT to make extremely effective use of this capability.

The ALT did have some useful properties which its replacement, DDT, did not, e.g. the ability to forward data directly to the destination, over the ALT, when no mapping was available yet for the destination. However, these were minor, and heavily outweighed by its problems.

A recent study, [Saucez], measured actual resolution times in the deployed LISP network during the changeover from ALT to DDT, allowing direct comparison of the performance of the two systems. The study took measurements from a variety of locations in the Internet, with respect to a number of different target EIDs. The results indicate that the performance was almost identical; there was more variance with DDT (perhaps due to the effects of caching), but the greatly improved scalability of DDT as compared to ALT made that effect acceptable.

B.5. Early NAT Support

The first mechanism used by LISP to support operation through a NAT device, described here, has now been superseded by the more general mechanism proposed in [NAT-Traversal]. That mechanism is, however, based heavily on this mechanism. The initial mechanism had some serious limitations, which is why that particular form of it has been dropped.

First, it only worked with some NATs, those which were configurable to allow inbound packet traffic to reach a configured host. The NAT had to be configured to know of the ETR.

Second, since NATs share addresses by using ports, it was only possible to have a single LISP node behind any given NAT device. That is because LISP expects all incoming data traffic to be on a specific port, so it was not possible to have multiple ETRs behind a single NAT (which normally would have only one global IP address to share). Even looking at the source host and port would not necessarily help, because some source ITR could be sending packets to both ETRs, so packets to either ETR could also have the identical source host/port. In short, there was no way for a NAT with multiple ETRs behind it to know which ETR the packet was for.

To support operation behind a NAT, there was a pair of new LISP control messages, LISP Echo-Request and Echo-Reply, which allowed the ETR to discover its temporary global address. The Echo-Request was sent to the configured Map-Server, and it replied with an Echo-Reply which included the source address from which the Echo Request was received (i.e. the public global address assigned to the ETR by the NAT). The ETR could then insert that address in any Map-Reply control messages which it sent to correspondent ITRs.

Echo-Request and Echo-Reply have been replaced by Info-Request and Info-Reply in the replacement, [NAT-Traversal], where they perform very similar functions; the main change is the addition of the {{xxx - probably the port, etc to allow multiple XTRs behind a NAT}}.

Author's Address

J. Noel Chiappa
Yorktown Museum of Asian Art
Yorktown, Virginia
USA

EMail: jnc@mit.edu

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 1, 2016

D. Saucez
INRIA
L. Iannone
Telecom ParisTech
O. Bonaventure
Universite catholique de Louvain
January 29, 2016

LISP Threats Analysis
draft-ietf-lisp-threats-15.txt

Abstract

This document provides a threat analysis of the Locator/Identifier Separation Protocol (LISP).

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 <http://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 August 1, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Threat model	3
2.1. Attacker's Operation Modes	4
2.1.1. On-path vs. Off-path Attackers	4
2.1.2. Internal vs. External Attackers	4
2.1.3. Live vs. Time-shifted attackers	5
2.1.4. Control-plane vs. Data-plane attackers	5
2.1.5. Cross mode attackers	5
2.2. Threat categories	5
2.2.1. Replay attack	5
2.2.2. Packet manipulation	6
2.2.3. Packet interception and suppression	6
2.2.4. Spoofing	6
2.2.5. Rogue attack	7
2.2.6. Denial of Service (DoS) attack	7
2.2.7. Performance attack	7
2.2.8. Intrusion attack	7
2.2.9. Amplification attack	7
2.2.10. Passive Monitoring Attacks	8
2.2.11. Multi-category attacks	8
3. Attack vectors	8
3.1. Gleaning	8
3.2. Locator Status Bits	9
3.3. Map-Version	10
3.4. Routing Locator Reachability	11
3.5. Instance ID	12
3.6. Interworking	12
3.7. Map-Request messages	12
3.8. Map-Reply messages	13
3.9. Map-Register messages	15
3.10. Map-Notify messages	15
4. Note on Privacy	15
5. Threats Mitigation	16
6. Security Considerations	17
7. IANA Considerations	17
8. Acknowledgments	17
9. References	17
9.1. Normative References	17
9.2. Informative References	18
Appendix A. Document Change Log (to be removed on publication) .	19
Authors' Addresses	21

1. Introduction

The Locator/ID Separation Protocol (LISP) is specified in [RFC6830]. This document provides an assessment of the potential security threats for the current LISP specifications if LISP is deployed in the Internet (i.e., a public non-trustable environment).

The document is composed of three main parts: the first defines a general threat model that attackers use to mount attacks. The second part, using this threat model, describes the techniques based on the LISP protocol and LISP architecture that attackers may use to construct attacks. The third part discusses mitigation techniques and general solutions to protect the LISP protocol and architecture from attacks.

This document does not consider all the possible uses of LISP as discussed in [RFC6830] and [RFC7215] and does not cover threats due to specific implementations. The document focuses on LISP unicast, including as well LISP Interworking [RFC6832], LISP Map-Server [RFC6833], and LISP Map-Versioning [RFC6834]. Additional threats may be discovered in the future while deployment continues. The reader is assumed to be familiar with these documents for understanding the present document.

This document assumes a generic IP service and does not discuss the difference, from a security viewpoint, between using IPv4 or IPv6.

2. Threat model

This document assumes that attackers can be located anywhere in the Internet (either in LISP sites or outside LISP sites) and that attacks can be mounted either by a single attacker or by the collusion of several attackers.

An attacker is a malicious entity that performs the action of attacking a target in a network where LISP is (partially) deployed by leveraging the LISP protocol and/or architecture.

An attack is the action of performing an illegitimate action on a target in a network where LISP is (partially) deployed.

The target of an attack is the entity (i.e., a device connected to the network or a network) that is aimed to undergo the consequences of an attack. Other entities can potentially undergo side effects of an attack, even though they are not directly targeted by the attack. The target of an attack can be selected specifically, i.e., a particular entity, or arbitrarily, i.e., any entity. Finally, an

attacker can aim at attacking one or several targets with a single attack.

Section 2.1 specifies the different modes of operation that attackers can follow to mount attacks and Section 2.2 specifies the different categories of attacks that attackers can build.

2.1. Attacker's Operation Modes

In this document attackers are classified according to their modes of operation, i.e., the temporal and spacial diversity of the attacker. These modes are not mutually exclusive, they can be used by attackers in any combination, and other modes may be discovered in the future. Further, attackers are not at all bound by our classification scheme, so implementers and those deploying will always need to do additional risk analysis for themselves.

2.1.1. On-path vs. Off-path Attackers

On-path attackers, also known as Men-in-the-Middle, are able to intercept and modify packets between legitimate communicating entities. On-path attackers are located either directly on the normal communication path (either by gaining access to a node on the path or by placing themselves directly on the path) or outside the location path but manage to deviate (or gain a copy of) packets sent between the communication entities. On-path attackers hence mount their attacks by modifying packets initially sent legitimately between communication entities.

An attacker is called off-path attacker if it does not have access to packets exchanged during the communication or if there is no communication. In order for their attacks to succeed, off-path attackers must hence generate packets and inject them in the network.

2.1.2. Internal vs. External Attackers

An internal attacker launches its attack from a node located within a legitimate LISP site. Such an attacker is either a legitimate node of the site or it exploits a vulnerability to gain access to a legitimate node in the site. Because of their location, internal attackers are trusted by the site they are in.

On the contrary, an external attacker launches its attacks from the outside of a legitimate LISP site.

2.1.3. Live vs. Time-shifted attackers

A live attacker mounts attacks for which it must remain connected as long as the attack is mounted. In other words, the attacker must remain active for the whole duration of the attack. Consequently, the attack ends as soon as the attacker (or the used attack vector) is neutralized.

On the contrary, a time-shifted attacker mounts attacks that remain active after it disconnects from the Internet.

2.1.4. Control-plane vs. Data-plane attackers

A control-plane attacker mounts its attack by using control-plane functionalities, typically the mapping system.

A data-plane attacker mounts its attack by using data-plane functionalities.

As there is no complete isolation between the control-plane and the data-plane, an attacker can operate in the control-plane (or data-plane) to mount attacks targeting the data-plane (or control-plane) or keep the attacked and targeted planes at the same layer (i.e., from control-plane to control-plane or from data-plane to data-plane).

2.1.5. Cross mode attackers

The attacker modes of operation are not mutually exclusive and hence attackers can combine them to mount attacks.

For example, an attacker can launch an attack using the control-plane directly from within a LISP site to which it is able to get temporary access (i.e., internal + control-plane attacker) to create a vulnerability on its target and later on (i.e., time-shifted + external attacker) mount an attack on the data plane (i.e., data-plane attacker) that leverages the vulnerability.

2.2. Threat categories

Attacks can be classified according to the nine following categories. These categories are not mutually exclusive and can be used by attackers in any combination.

2.2.1. Replay attack

A replay attack happens when an attacker retransmits at a later time, and without modifying it, a packet (or a sequence of packets) that

has already been transmitted.

2.2.2. Packet manipulation

A packet manipulation attack happens when an attacker receives a packet, modifies the packet (i.e., changes some information contained in the packet) and finally transmits the packet to its final destination that can be the initial destination of the packet or a different one.

2.2.3. Packet interception and suppression

In a packet interception and suppression attack, the attacker captures the packet and drops it before it can reach its final destination.

2.2.4. Spoofing

With a spoofing attack, the attacker injects packets in the network pretending to be another node. Spoofing attacks are made by forging source addresses in packets.

It should be noted that with LISP, packet spoofing is similar to spoofing with any other existing tunneling technology currently deployed in the Internet. Generally the term "spoofed packet" indicates a packet containing a source IP address that is not the actual originator of the packet. Hence, since LISP uses encapsulation, the spoofed address could be in the outer header as well as in the inner header, this translates to two types of spoofing.

Inner address spoofing: the attacker uses encapsulation and uses a spoofed source address in the inner packet. In case of data-plane LISP encapsulation, that corresponds to spoofing the source EID (End-point IDentifier) address of the encapsulated packet.

Outer address spoofing: the attacker does not use encapsulation and spoofs the source address of the packet. In case of data-plane LISP encapsulation, that corresponds to spoofing the source RLOC (Routing LOCator) address of the encapsulated packet.

Note that the two types of spoofing are not mutually exclusive, rather all combinations are possible and could be used to perform different kinds of attacks. For example, an attacker outside a LISP site can generate a packet with a forged source IP address (i.e., outer address spoofing) and forward it to a LISP destination. The packet is then eventually encapsulated by a PITR (Proxy Ingress

Tunnel Router) so that once encapsulated the attack corresponds to a inner address spoofing. One can also imagine an attacker forging a packet with encapsulation where both inner and outer source addresses are spoofed.

It is important to note that the combination of inner and outer spoofing makes the identification of the attacker complex as the packet may not contain information that allows to detect the origin of the attack.

2.2.5. Rogue attack

In a rogue attack the attacker manages to appear as a legitimate source of information, without faking its identity (as opposed to a spoofing attacker).

2.2.6. Denial of Service (DoS) attack

A Denial of Service (DoS) attack aims at disrupting a specific targeted service to make it unable to operate properly.

2.2.7. Performance attack

A performance attacks aims at exploiting computational resources (e.g., memory, processor) of a targeted node so as to make it unable to operate properly.

2.2.8. Intrusion attack

In an intrusion attack, the attacker gains remote access to a resource (e.g., a host, a router, or a network) or information that it legitimately should not have access. Intrusion attacks can lead to privacy leakages.

2.2.9. Amplification attack

In an amplification attack, the traffic generated by the target of the attack in response to the attack is larger than the traffic that the attacker must generate.

In some cases, the data-plane can be several orders of magnitude faster than the control-plane at processing packets. This difference can be exploited to overload the control-plane via the data-plane without overloading the data-plane.

2.2.10. Passive Monitoring Attacks

An attacker can use pervasive monitoring, which is a technical attack [RFC7258], targeting information about LISP traffic that may or not be used to mount other type of attacks.

2.2.11. Multi-category attacks

Attacks categories are not mutually exclusive and any combination can be used to perform specific attacks.

For example, one can mount a rogue attack to perform a performance attack starving the memory of an ITR (Ingress Tunnel Router) resulting in a DoS (Denial-of-Service) on the ITR.

3. Attack vectors

This section presents attack techniques that may be used by attackers when leveraging the LISP protocol and/or architecture.

3.1. Gleaning

To reduce the time required to obtain a mapping, the optional gleaning mechanism defined for LISP allows an xTR (Ingress and/or Egress Tunnel Router) to directly learn a mapping from the LISP data encapsulated packets and the Map-Request packets that it receives. LISP encapsulated data packets contain a source RLOC, destination RLOC, source EID and destination EID. When an xTR receives an encapsulated data packet coming from a source EID for which it does not already know a mapping, it may insert the mapping between the source RLOC and the source EID in its EID-to-RLOC Cache. The same technique can be used when an xTR receives a Map-Request as the Map-Request also contains a source EID address and a source RLOC. Once a gleaned entry has been added to the EID-to-RLOC cache, the xTR sends a Map-Request to retrieve the actual mapping for the gleaned EID from the mapping system.

If a packet injected by an off-path attacker and with a spoofed inner address is gleaned by an xTR then the attacker may divert the traffic meant to be delivered to the spoofed EID as long as the gleaned entry is used by the xTR. This attack can be used as part of replay, packet manipulation, packet interception and suppression, or DoS attacks as the packets are sent to the attacker.

If the packet sent by the attacker contains a spoofed outer address instead of a spoofed inner address then it can achieve a DoS or a performance attack as the traffic normally destined to the attacker

will be redirected to the spoofed source RLOC. Such traffic may overload the owner of the spoofed source RLOC, preventing it from operating properly.

If the packet injected uses both inner and outer spoofing, the attacker can achieve a spoofing, a performance, or an amplification attack as traffic normally destined to the spoofed EID address will be sent to the spoofed RLOC address. If the attacked LISP site also generates traffic to the spoofed EID address, such traffic may have a positive amplification factor.

A gleaning attack does not only impact the data-plane but can also have repercussions on the control-plane as a Map-Request is sent after the creation of a gleaned entry. The attacker can then achieve DoS and performance attacks on the control-plane. For example, if an attacker sends a packet for each address of a prefix not yet cached in the EID-to-RLOC cache of an xTR, the xTR will potentially send a Map-Request for each such packet until the mapping is installed which leads to an over-utilisation of the control-plane as each packet generates a control-plane event. In order for this attack to succeed, the attacker may not need to use spoofing. This issue can occur even if gleaning is turned off since whether or not gleaning is used as the ITR may need to send a Map-Request in response to incoming packets whose EID is not currently in the cache.

Gleaning attacks are fundamentally involving a time-shifted mode of operation as the attack may last as long as the gleaned entry is kept by the targeted xTR. RFC 6830 [RFC6830] recommends to store the gleaned entries for only a few seconds which limits the duration of the attack.

Gleaning attacks always involve external data-plane attackers but results in attacks on either the control-plane or data-plane.

Note, the outer spoofed address does not need to be the RLOC of a LISP site, it may be any address.

3.2. Locator Status Bits

When the L bit in the LISP header is set to 1, it indicates that the second 32-bits longword of the LISP header contains the Locator Status Bits. In this field, each bit position reflects the status of one of the RLOCs mapped to the source EID found in the encapsulated packet. The reaction of a LISP xTR that receives such a packet is left as operational choice in [RFC6830].

When an attacker sends a LISP encapsulated packet with an illegitimately crafted LSB to an xTR, it can influence the xTR's

choice of the locators for the prefix associated to the source EID. In case of an off-path attacker, the attacker must inject a forged packet in the network with a spoofed inner address. An on-path attacker can manipulate the LSB of legitimate packets passing through it and hence does not need to use spoofing. Instead of manipulating the LSB field, an on-path attacker can also obtain the same result of injecting packets with invalid LSB values by replaying packets.

The LSB field can be leveraged to mount a DoS attack by either declaring all RLOCs as unreachable (all LSB set to 0), or by concentrating all the traffic to one RLOC (e.g., all but one LSB set to 0) and hence overloading the RLOC concentrating all the traffic from the xTR, or by forcing packets to be sent to RLOCs that are actually not reachable (e.g., invert LSB values).

The LSB field can also be used to mount a replay, a packet manipulation, or a packet interception and suppression attack. Indeed, if the attacker manages to be on the path between the xTR and one of the RLOCs specified in the mapping, forcing packets to go via that RLOC implies that the attacker will gain access to the packets.

Attacks using the LSB are fundamentally involving a time-shifted mode of operation as the attack may last as long as the reachability information gathered from the LSB is used by the xTR to decide the RLOCs to be used.

3.3. Map-Version

When the Map-Version bit of the LISP header is set to 1, it indicates that the low-order 24 bits of the first 32 bits longword of the LISP header contain a Source and Destination Map-Version. When a LISP xTR receives a LISP encapsulated packet with the Map-Version bit set to 1, the following actions are taken:

- o It compares the Destination Map-Version found in the header with the current version of its own configured EID-to-RLOC mapping, for the destination EID found in the encapsulated packet. If the received Destination Map-Version is smaller (i.e., older) than the current version, the ETR should apply the SMR (Solicit-Map-Request) procedure described in [RFC6830] and send a Map-Request with the SMR bit set.
- o If a mapping exists in the EID-to-RLOC Cache for the source EID, then it compares the Map-Version of that entry with the Source Map-Version found in the header of the packet. If the stored mapping is older (i.e., the Map-Version is smaller) than the source version of the LISP encapsulated packet, the xTR should send a Map-Request for the source EID.

A cross-mode attacker can use the Map-Version bit to mount a DoS attack, an amplification attack, or a spoofing attack. For instance if the mapping cached at the xTR is outdated, the xTR will send a Map-Request to retrieve the new mapping which can yield to a DoS attack (by excess of signalling traffic) or an amplification attack if the data-plane packet sent by the attacker is smaller, or otherwise uses fewer resources, than the control-plane packets sent in response to the attacker's packet. With a spoofing attack, and if the xTR considers that the spoofed ITR has an outdated mapping, it will send an SMR to the spoofed ITR which can result in performance, amplification, or DoS attack as well.

Map-Version attackers are inherently cross mode as the Map-Version is a method to put control information in the data-plane. Moreover, this vector involves live attackers. Nevertheless, on-path attackers do not have specific advantage over off-path attackers.

3.4. Routing Locator Reachability

The Nonce-Present and Echo-Nonce bits in the LISP header are used to verify the reachability of an xTR. A testing xTR sets the Echo-Nonce and the Nonce-Present bits in LISP data encapsulated packets and include a random nonce in the LISP header of packets. Upon reception of these packets, the tested xTR stores the nonce and echoes it whenever it returns a LISP encapsulated data packets to the testing xTR. The reception of the echoed nonce confirms that the tested xTR is reachable.

An attacker can interfere with the reachability test by sending two different types of packets:

1. LISP data encapsulated packets with the Nonce-Present bit set and a random nonce. Such packets are normally used in response to a reachability test.
2. LISP data encapsulated packets with the Nonce-Present and the Echo-Nonce bits both set. These packets will force the receiving ETR to store the received nonce and echo it in the LISP encapsulated packets that it sends. These packets are normally used as a trigger for a reachability test.

The first type of packets are used to make xTRs think that an other xTR is reachable while it is not. It is hence a way to mount a DoS attack (i.e., the ITR will send its packet to a non-reachable ETR when it should use another one).

The second type of packets could be exploited to attack the nonce-based reachability test. If the attacker sends a continuous flow of

packets that each have a different random nonce, the ETR that receives such packets will continuously change the nonce that it returns to the remote ITR, which can yield to a performance attack. If the remote ITR tries a nonce-reachability test, this test may fail because the ETR may echo an invalid nonce. This hence yields to a DoS attack.

In the case of an on-path attacker, a packet manipulation attack is necessary to mount the attack. To mount such an attack, an off-path attacker must mount an outer address spoofing attack.

If an xTR chooses to periodically check with active probes the liveness of entries in its EID-to-RLOC cache (as described in section 6.3 of [RFC6830]), then this may amplify the attack that caused the insertion of entries being checked.

3.5. Instance ID

LISP allows to carry in its header a 24-bits value called Instance ID and used on the ITR to indicate which local Instance ID has been used for encapsulation, while on the ETR the instance ID decides the forwarding table to use to forward the decapsulated packet in the LISP site.

An attacker (either a control-plane or data-plane attacker) can use the instance ID functionality to mount an intrusion attack.

3.6. Interworking

[RFC6832] defines Proxy-ITR and Proxy-ETR network elements to allow LISP and non-LISP sites to communicate. The Proxy-ITR has functionality similar to the ITR, however, its main purpose is to encapsulate packets arriving from the DFZ (Default-Free Zone) in order to reach LISP sites. A PETR (Proxy Egress Tunnel Router) has functionality similar to the ETR, however, its main purpose is to inject de-encapsulated packets in the DFZ in order to reach non-LISP sites from LISP sites. As a PITR (or PETR) is a particular case of ITR (or ETR), it is subject to similar attacks as ITRs (or ETRs).

As any other system relying on proxies, LISP interworking can be used by attackers to hide their exact origin in the network.

3.7. Map-Request messages

A control-plane off-path attacker can exploit Map-Request messages to mount DoS, performance, or amplification attacks. By sending Map-Request messages at high rate, the attacker can overload nodes involved in the mapping system. For instance sending Map-Requests at

high rate can considerably increase the state maintained in a Map-Resolver or consume CPU cycles on ETRs that have to process the Map-Request packets they receive in their slow path (i.e., performance or DoS attack). When the Map-Reply packet is larger than the Map-Request sent by the attacker, that yields to an amplification attack. The attacker can combine the attack with a spoofing attack to overload the node to which the spoofed address is actually attached.

Note, if the attacker sets the P bit (Probe Bit) in the Map-Request, it will cause legitimately sending the Map-Request directly to the ETR instead of passing through the mapping system.

The SMR bit can be used to mount a variant of these attacks.

For efficiency reasons, Map-Records can be appended to Map-Request messages. When an xTR receives a Map-Request with appended Map-Records, it does the same operations as for the other Map-Request messages and so is subject to the same attacks. However, it also installs in its EID-to-RLOC cache the Map-Records contained in the Map-Request. An attacker can then use this vector to force the installation of mappings in its target xTR. Consequently, the EID-to-RLOC cache of the xTR is polluted by potentially forged mappings allowing the attacker to mount any of the attacks categorized in Section 2.2 (see Section 3.8 for more details). Note, the attacker does not need to forge the mappings present in the Map-Request to achieve a performance or DoS attack. Indeed, if the attacker owns a large enough EID prefix it can de-aggregate it in many small prefixes, each corresponding to another mapping and it installs them in the xTR cache by mean of the Map-Request.

Moreover, attackers can use Map Resolver and/or Map Server network elements to relay its attacks and hide the origin of the attack. Indeed, on the one hand, a Map Resolver is used to dispatch Map-Request to the mapping system and, on the other hand, a Map Server is used to dispatch Map-Requests coming from the mapping system to ETRs that are authoritative for the EID in the Map-Request.

3.8. Map-Reply messages

Most of the security risks associated with Map-Reply messages will depend on the 64 bits nonce that is included in a Map-Request and returned in the Map-Reply. Given the size of the nonce (64 bits), if best current practice is used [RFC4086] and if an ETR does not accept Map-Reply messages with an invalid nonce, the risk of an off-path attack is limited. Nevertheless, the nonce only confirms that the Map-Reply received was sent in response to a Map-Request sent, it does not validate the contents of that Map-Reply.

If an attacker manages to send a valid (i.e., in response to a Map-Request and with the correct nonce) Map-Reply to an ITR, then it can perform any of the attacks categorised in Section 2.2 as it can inject forged mappings directly in the ITR EID-to-RLOC cache. For instance, if the mapping injected to the ITR points to the address of a node controlled by the attacker, it can mount replay, packet manipulation, packet interception and suppression, or DoS attacks, as it will receive every packet destined to a destination lying in the EID prefix of the injected mapping. In addition, the attacker can inject a plethora of mappings in the ITR to mount a performance attack by filling up the EID-to-RLOC cache of the ITR. The attacker can also mount an amplification attack if the ITR at that time is sending a large number of packets to the EIDs matching the injected mapping. In this case, the RLOC address associated to the mapping is the address of the real target of the attacker and so all the traffic of the ITR will be sent to the target which means that with one single packet the attacker may generate very high traffic towards its final target.

If the attacker is a valid ETR in the system, it can mount a rogue attack if it uses prefixes over-claiming. In such a scenario, the attacker ETR replies to a legitimate Map-Request message which it received with a Map-Reply message that contains an EID-Prefix that is larger than the prefix owned by the attacker. For example if the owned prefix is 192.0.2.0/25 but the Map-Reply contains a mapping for 192.0.2.0/24, then the mapping will influence packets destined to other EIDs than the one attacker has authority on. With such technique, the attacker can mount the attacks presented above as it can (partially) control the mappings installed on its target ITR. To force its target ITR to send a Map-Request, nothing prevents the attacker to initiate some communication with the ITR. This method can be used by internal attackers that want to control the mappings installed in their site. To that aim, they simply have to collude with an external attacker ready to over-claim prefixes on behalf of the internal attacker.

Note, when the Map-Reply is in response to a Map-Request sent via the mapping system (i.e., not send directly from the ITR to an ETR), the attacker does not need to use a spoofing attack to achieve its attack as by design the source IP address of a Map-Reply is not known in advance by the ITR.

Map-Request and Map-Reply messages are exposed to any type of attackers, on-path or off-path but also external or internal attackers. Also, even though they are control message, they can be leveraged by data-plane attackers. As the decision of removing mappings is based on the TTL indicated in the mapping, time-shifted attackers can take advantage of injecting forged mappings as well.

3.9. Map-Register messages

Map-Register messages are sent by ETRs to Map Servers to indicate to the mapping system the EID prefixes associated to them. The Map-Register message provides an EID prefix and the list of ETRs that are able to provide Map-Replies for the EID covered by the EID prefix.

As Map-Register messages are protected by an authentication mechanism, only a compromised ETR can register itself to its allocated Map Server.

A compromised ETR can over-claim the prefix it owns in order to influence the route followed by Map-Requests for EIDs outside the scope of its legitimate EID prefix (see Section 3.8 for the list of over-claiming attacks).

A compromised ETR can also de-aggregate its EID prefix in order to register more EID prefixes than necessary to its Map Servers (see Section 3.7 for the impact of de-aggregation of prefixes by an attacker).

Similarly, a compromised Map Server can accept an invalid registration or advertise an invalid EID prefix to the mapping system.

3.10. Map-Notify messages

Map-Notify messages are sent by a Map Server to an ETR to acknowledge the reception and processing of a Map-Register message.

Similarly to the pair Map-Request/Map-Reply, the pair Map-Register/Map-Notify is protected by a nonce making it difficult for an attacker to inject a falsified notification to an ETR to make this ETR believe that the registration succeeded when it has not.

4. Note on Privacy

As reviewed in [RFC6973], universal privacy considerations are difficult to establish as the privacy definitions may vary for different scenarios. As a consequence, this document does not aim at identifying privacy issues related to the LISP protocol but the security threats identified in this document could play a role in privacy threats as defined in section 5 of [RFC6973].

Similar to public deployments of any other control plane protocols, in an Internet deployment, LISP mappings are public and hence provide information about the infrastructure and reachability of LISP sites

(i.e., the addresses of the edge routers). Depending upon deployment details, LISP map replies might or might not provide finer grained and more detailed information than is available with currently deployed routing and control protocols.

5. Threats Mitigation

Most of the above threats can be mitigated with careful deployment and configuration (e.g., filter) and also by applying the general rules of security, e.g. only activating features that are necessary for the deployment and verifying the validity of the information obtained from third parties.

The control-plane is the most critical part of LISP from a security viewpoint and it is worth to notice that the LISP specifications already offer an authentication mechanism for mappings registration ([RFC6833]). This mechanism, combined with LISP-SEC [I-D.ietf-lisp-sec], strongly mitigates threats in non-trustable environments such as the Internet. Moreover, an authentication data field for Map-Request messages and Encapsulated Control messages was allocated [RFC6830]. This field provides a general authentication mechanism technique for the LISP control-plane which future specifications may use while staying backward compatible. The exact technique still has to be designed and defined. To maximally mitigate the threats on the mapping system, authentication must be used, whenever possible, for both Map-Request and Map-Reply messages and for messages exchanged internally among elements of the mapping system, such as specified in [I-D.ietf-lisp-sec] and [I-D.ietf-lisp-ddt].

Systematically applying filters and rate-limitation, as proposed in [RFC6830], will mitigate most of the threats presented in this document. In order to minimise the risk of overloading the control-plane with actions triggered from data-plane events, such actions should be rate limited.

Moreover, all information opportunistically learned (e.g., with LSB or gleaning) should be used with care until they are verified. For example, a reachability change learned with LSB should not be used directly to decide the destination RLOC, but instead should trigger a rate-limited reachability test. Similarly, a gleaned entry should be used only for the flow that triggered the gleaning procedure until the gleaned entry has been verified [Trilogy].

6. Security Considerations

This document provides a threat analysis and proposes mitigation techniques for the Locator/Identifier Separation Protocol.

7. IANA Considerations

This document makes no request to IANA.

8. Acknowledgments

This document builds upon the document of Marcelo Bagnulo ([I-D.bagnulo-lisp-threat]), where the flooding attack and the reference environment was first described.

The authors would like to thank Deborah Brungard, Ronald Bonica, Albert Cabellos, Ross Callon, Noel Chiappa, Florin Coras, Vina Ermagan, Dino Farinacci, Stephen Farrell, Joel Halpern, Emily Hiltzik, Darrel Lewis, Edward Lopez, Fabio Maino, Terry Manderson, and Jeff Wheeler for their comments.

This work has been partially supported by the INFISO-ICT-216372 TRILOGY Project (www.trilogy-project.org).

The work of Luigi Iannone has been partially supported by the ANR-13-INFR-0009 LISP-Lab Project (www.lisp-lab.org) and the EIT KIC ICT-Labs SOFNETS Project.

9. References

9.1. Normative References

- [RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <<http://www.rfc-editor.org/info/rfc6830>>.
- [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, <<http://www.rfc-editor.org/info/rfc6832>>.
- [RFC6833] Fuller, V. and D. Farinacci, "Locator/ID Separation Protocol (LISP) Map-Server Interface", RFC 6833,

DOI 10.17487/RFC6833, January 2013,
<<http://www.rfc-editor.org/info/rfc6833>>.

[RFC6834] Iannone, L., Saucez, D., and O. Bonaventure, "Locator/ID Separation Protocol (LISP) Map-Versioning", RFC 6834, DOI 10.17487/RFC6834, January 2013, <<http://www.rfc-editor.org/info/rfc6834>>.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <<http://www.rfc-editor.org/info/rfc6973>>.

9.2. Informative References

- [I-D.bagnulo-lisp-threat]
Bagnulo, M., "Preliminary LISP Threat Analysis", draft-bagnulo-lisp-threat-01 (work in progress), July 2007.
- [I-D.ietf-lisp-ddt]
Fuller, V., Lewis, D., Ermagan, V., and A. Jain, "LISP Delegated Database Tree", draft-ietf-lisp-ddt-03 (work in progress), April 2015.
- [I-D.ietf-lisp-sec]
Maino, F., Ermagan, V., Cabellos-Aparicio, A., and D. Saucez, "LISP-Security (LISP-SEC)", draft-ietf-lisp-sec-09 (work in progress), October 2015.
- [RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <<http://www.rfc-editor.org/info/rfc4086>>.
- [RFC7215] Jakab, L., Cabellos-Aparicio, A., Coras, F., Domingo-Pascual, J., and D. Lewis, "Locator/Identifier Separation Protocol (LISP) Network Element Deployment Considerations", RFC 7215, DOI 10.17487/RFC7215, April 2014, <<http://www.rfc-editor.org/info/rfc7215>>.
- [RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <<http://www.rfc-editor.org/info/rfc7258>>.
- [Trilogy] Saucez, D. and L. Iannone, "How to mitigate the effect of scans on mapping systems", Trilogy Future Internet Summer

School., 2009.

Appendix A. Document Change Log (to be removed on publication)

- o Version 15 Posted January 2016.
 - * Few changes to address Stephen Farrel comments as part of the IESG Review.
- o Version 14 Posted December 2015.
 - * Editorial changes according to Deborah Brungard's (Routing AD) review.
- o Version 13 Posted August 2015.
 - * Keepalive version.
- o Version 12 Posted March 2015.
 - * Addressed comments by Ross Callon on the mailing list (<http://www.ietf.org/mail-archive/web/lisp/current/msg05829.html>).
 - * Addition of a section discussing mitigation techniques for deployments in non-trustable environments.
- o Version 11 Posted December 2014.
 - * Editorial polishing. Clarifications added in few points.
- o Version 10 Posted July 2014.
 - * Document completely remodelled according to the discussions on the mailing list in the thread <http://www.ietf.org/mail-archive/web/lisp/current/msg05206.html> and to address comments from Ronald Bonica and Ross Callon.
- o Version 09 Posted March 2014.
 - * Updated document according to the review of A. Cabellos.
- o Version 08 Posted October 2013.
 - * Addition of a privacy consideration note.
 - * Editorial changes

- o Version 07 Posted October 2013.
 - * This version is updated according to the thorough review made during October 2013 LISP WG interim meeting.
 - * Brief recommendations put in the security consideration section.
 - * Editorial changes
- o Version 06 Posted October 2013.
 - * Complete restructuration, temporary version to be used at October 2013 interim meeting.
- o Version 05 Posted August 2013.
 - * Removal of severity levels to become a short recommendation to reduce the risk of the discussed threat.
- o Version 04 Posted February 2013.
 - * Clear statement that the document compares threats of public LISP deployments with threats in the current Internet architecture.
 - * Addition of a severity level discussion at the end of each section.
 - * Addressed comments from V. Ermagan and D. Lewis' reviews.
 - * Updated References.
 - * Further editorial polishing.
- o Version 03 Posted October 2012.
 - * Dropped Reference to RFC 2119 notation because it is not actually used in the document.
 - * Deleted future plans section.
 - * Updated References
 - * Deleted/Modified sentences referring to the early status of the LISP WG and documents at the time of writing early versions of the document.

- * Further editorial polishing.
- * Fixed all ID nits.
- o Version 02 Posted September 2012.
 - * Added a new attack that combines over-claiming and de-aggregation (see Section 3.8).
 - * Editorial polishing.
- o Version 01 Posted February 2012.
 - * Added discussion on LISP-DDT.
- o Version 00 Posted July 2011.
 - * Added discussion on LISP-MS>.
 - * Added discussion on Instance ID.
 - * Editorial polishing of the whole document.
 - * Added "Change Log" appendix to keep track of main changes.
 - * Renamed "draft-saucez-lisp-security-03.txt".

Authors' Addresses

Damien Saucez
INRIA
2004 route des Lucioles BP 93
06902 Sophia Antipolis Cedex
France

Email: damien.saucez@inria.fr

Luigi Iannone
Telecom ParisTech
23, Avenue d'Italie, CS 51327
75214 PARIS Cedex 13
France

Email: ggx@gigix.net

Olivier Bonaventure
Universite catholique de Louvain
Place St. Barbe 2
Louvain la Neuve
Belgium

Email: olivier.bonaventure@uclouvain.be

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: August 18, 2014

C. Cassar
I. Kouvelas
D. Lewis
Cisco Systems
February 14, 2014

LISP Reliable Transport
draft-kouvelas-lisp-reliable-transport-00.txt

Abstract

The communication between LISP ETRs and Map-Servers is based on unreliable UDP message exchange coupled with periodic message transmission in order to maintain soft state. The drawback of periodic messaging is the constant load imposed on both the ETR and the Map-Server. New use cases for LISP have increased the amount of state that needs to be communicated with requirements that are not satisfied by the current mechanism. This document introduces the use of a reliable transport for ETR to Map-Server communication in order to eliminate the periodic messaging overhead, while providing reliability, flow-control and endpoint liveness detection.

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 <http://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 August 18, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Requirements Notation	3
3. Message Format	3
4. Session Establishment	4
5. Error Notifications	5
6. EID Prefix Registration	7
6.1. Reliable Mapping Registration Messages	7
6.1.1. Registration Message	7
6.1.2. Registration Acknowledgement Message	8
6.1.3. Registration Rejected Message	9
6.1.4. Registration Refresh Message	9
6.1.5. Mapping Notification Message	10
6.2. ETR Behavior	11
6.3. Map-Server Behavior	15
7. Security Considerations	16
8. IANA Considerations	16
9. Acknowledgments	16
10. Normative References	16
Authors' Addresses	16

1. Introduction

The communication channel between LISP ETRs and Map-Servers is based on unreliable UDP message exchange [RFC6833]. Where required, reliability is pursued through periodic retransmissions that maintain soft state on the peer. Map-Register messages are retransmitted every minute by an ETR and the Map-Server times out its state if the state is not refreshed for three successive periods. When registering multiple EID-Prefixes, the ETR includes multiple mapping records in the Map-Register message. Packet size limitations provide an upper bound to the number of mapping records that can be placed in each Map-Register message. When the ETR has more EID-Prefixes to register than can be packed in a single Map-Register message, the mapping records for the EID-Prefixes are split across multiple Map-Register messages.

The drawback of the periodic registration is the constant load that it introduces on both the ETR and the Map-Server. The ETR uses resources to periodically build and transmit the Map-Register

messages, and to process the resulting Map-Notify messages issued by the Map-Server. The Map-Server uses resources to process the received Map-Register messages, update the corresponding registration state, and build and transmit the matching Map-Notify messages. When the number of EID-Prefixes to be registered by an ETR is small, the resulting load imposed by periodic registrations may not be significant. The ETR will only transmit a single Map-Register message each period that contains a small number of mapping records.

In some LISP deployments, a large set of EID-Prefixes must be registered by each ETR (e.g. mobility, database redistribution). Use cases with a large set of EID-Prefixes behind an ETR will result in a much higher load. An example is LISP mobility deployments where EID-Prefixes are limited to host entries. ETRs may have thousands of hosts to register resulting in hundreds of Map-Register and Map-Notify messages per registration period.

A transport is required for the ETR to Map-Server communication that provides reliability, flow-control and endpoint liveness notifications. This document describes the use of TCP or SCTP as a LISP reliable transport. The initial application for the LISP reliable transport session is the support of scalable EID prefix registration. The reliable session mechanism is defined to be extensible so that it can support additional LISP communication requirements as they arise using a single reliable transport session between an ETR and a Map-Server. The use of the reliable transport session for EID prefix registration is an alternative and does not replace the existing UDP based mechanism.

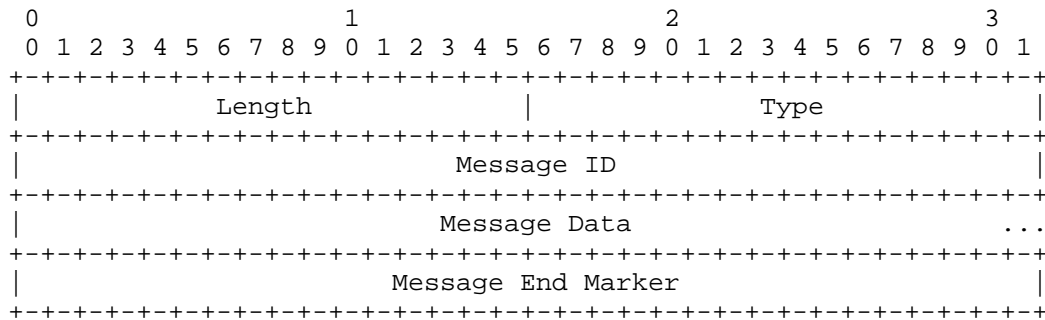
2. Requirements Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Message Format

A single LISP reliable transport session may carry information for multiple LISP applications. One such application is the registration of EID to RLOC mappings that operates over a session between an ETR and a Map-Server. Communication over a session is based on the exchange of messages. This document defines a base set of messages to support session establishment and management. It also defines the messages for the EID to RLOC mapping registration application.

To support protocol extensibility when new applications, or extensions to existing applications are introduced, the messages are based on a TLV format.



Reliable transport message format

- o Type: 16 bit type field identifying the message type.
- o Length: 16 bit field that provides the total size of the message in octets including the length, type and end marker fields. The length allows the receiver to locate the next message in the TCP stream. The minimum value of the length field is 8.
- o ID: A 32-bit value that identifies the message. May be used by the receiver to identify the message in replies or notification messages.
- o Data: Type specific message contents.
- o End Marker: A 32-bit message end marker that must be set to 0x9FACADE9. The End Marker is used by the receiver to validate that it has correctly parsed or skipped a message and provides a method to detect formatting errors. Note that message data may also contain this marker, and that the marker itself is not sufficient for parsing the message.

The base message format does not indicate how the peer should deal with the message in cases where the message type is not supported/understood. This is best dealt with by the application. For example, in case an error notification is returned, or an expected acknowledgement message is not received, the application might choose various courses of action; from simply logging that the feature is not supported, all the way to tearing the relationship with the peer down for the feature, or for all LISP features.

4. Session Establishment

The LISP router that performs the active open initiates the connection from a locally generated source transport port number to the well-known destination transport port assigned to LISP. The LISP

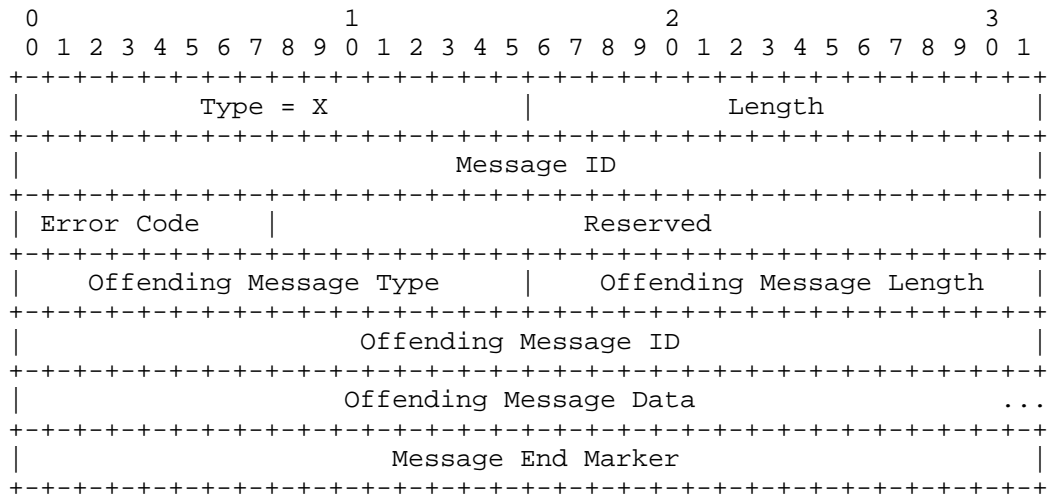
router that performs the passive open listens on the well-known local transport port and does not qualify the remote transport port number. In the ETR to Map-Server reliable transport session, the ETR assumes the active role and the Map-Server passively accepts connections.

A single reliable transport session can be established between a pair of LISP peers to cover all communication needs. For example, an ETR that has EID prefix registrations for multiple EID instances and EID address families might only establish a single session with the Map-Server.

When using TCP and symmetric connection establishment LISP must perform collision detection and duplicate session elimination. To accomplish that, LISP peer ID messages will be exchanged between the peers once a session is established. If duplicate sessions are detected then the one that was initiated by the router with the higher ID is kept and the other session is torn down. TBD

5. Error Notifications

The error notification message is used to communicate base reliable transport session communication errors. LISP applications making use of the reliable transport session and having to communicate application specific errors must define their own messages to do so. An error notification is issued when the receiver of a message does not recognize the message type or cannot parse the message contents. The notification includes the offending message type and ID and as much of the offending message data as the notification sender wishes to.



Error notification message format

- o Error Code: An 8 bit field identifying the type of error that occurred. Defined errors are:
 - * Unrecognized message type.
 - * Message format error.
- o Reserved: Set to zero by the sender and ignored by the receiver.
- o Offending Message Type: 16 bit type field identifying the message type of the offending message that triggered this error notification. This is copied from the Type field of the offending message.
- o Offending Message Length: 16 bit field that provides the total size of the offending message in octets. This is copied from the Length field of the offending message.
- o Offending Message ID: A 32-bit field that is set to the Message ID field of the offending message.
- o Offending Message Data: The Data from the offending message that triggered this error notification. The sender of the notification may include as much of the original data as is deemed necessary. The length of the Offending Message Data field is not provided by the Offending Message Length field and is determined by subtracting the size of the other fields in the message from the

Length field. It is valid to not include any of the offending message data when sending an error notification.

- o End Marker: A 32-bit message end marker that must be set to 0x9FACADE9. The End Marker is used by the receiver to validate that it has correctly parsed or skipped a message and provides a method to detect formatting errors. Note that message data may also contain this marker, and that the marker itself is not sufficient for parsing the message.

An error notification cannot be the offending message in another error notification and MUST NOT trigger such a message.

6. EID Prefix Registration

EID prefix registration uses the reliable transport session between an ETR and a Map-Server to communicate the ETR local EID database EID to RLOC mappings to the Map-Server. In contrast to the UDP based periodic registration, mapping information over the reliable transport session is only sent when there is new information available for the Map-Server. The Map-Server does not maintain a timer to expire registrations communicated over the reliable transport session. Instead an explicit de-registration (a registration carrying a zero TTL) is needed to delete the state maintained by the Map-Server.

The key used to identify registration mapping records in the ETR to Map-Server communication is the EID prefix. The prefix may be specified using an LCAF encoding that includes an EID instance ID.

When the reliable transport session goes down, registration mappings learned by the Map-Server are treated as periodic UDP registrations and a timer is used to expire them after 3 minutes. During this period UDP based registrations or the re-establishment of the reliable transport session and subsequent communication of a new mapping can update the EID prefix mapping state.

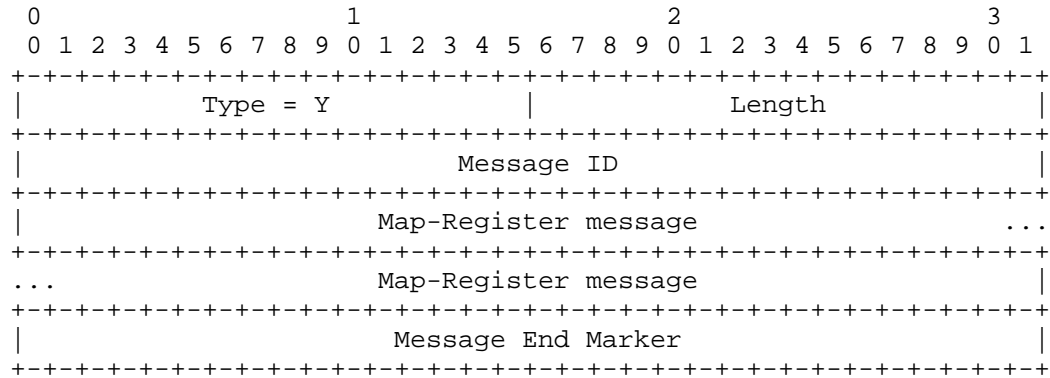
6.1. Reliable Mapping Registration Messages

This section defines the LISP reliable transport session messages used to communicate local EID database registrations between the ETR and the Map-Server.

6.1.1. Registration Message

The reliable transport Registration message is used to communicate EID to RLOC mapping registrations from the ETR to the Map-Server. The Registration message uses exactly the same format as the UDP Map-

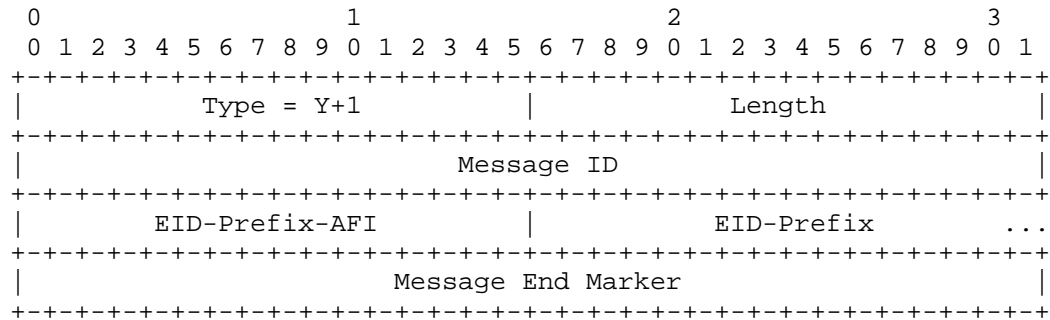
Register message but instead of the IP/UDP header, the Map-Register is placed within the value section of the reliable transport TLV. A common message format is proposed to leverage the authentication features built into the UDP Map-Register message and increase code reuse.



Registration message format

6.1.2. Registration Acknowledgement Message

The Acknowledgement message is sent from the Map-Server to the ETR to confirm successful registration of an EID prefix previously communicated by a reliable transport session Registration message. The Registration Acknowledgement message does not carry a mapping record (the map servers view of the mapping). This is accomplished by the LISP reliable transport Map Notification message.



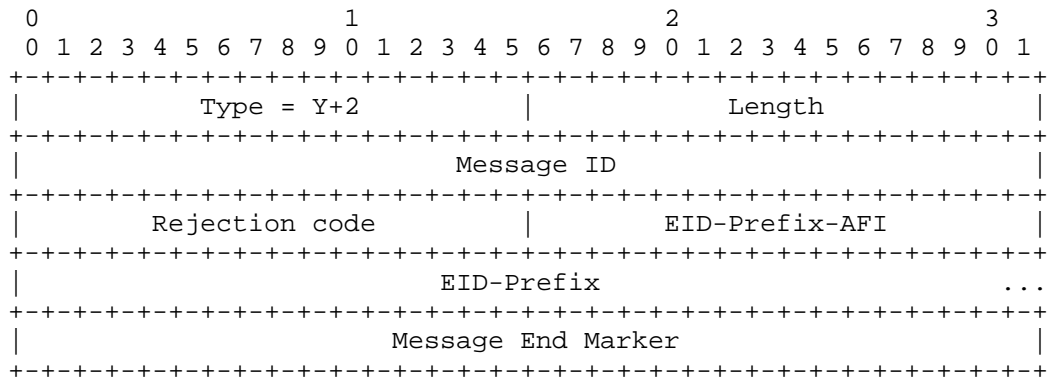
Registration Acknowledgement message format

- o EID-Prefix AFI: Address family identifier for the EID prefix in the following field.

- o EID-Prefix: The EID prefix from the received Registration.

6.1.3. Registration Rejected Message

Negative acknowledgement sent from the Map-Server to the ETR to indicate that the registration of a specific EID prefix was rejected. The ETR must keep track of the fact that the registration of the EID prefix was rejected by the Map-Server and be prepared to re-register the mapping when requested through a failed Registration Refresh request.

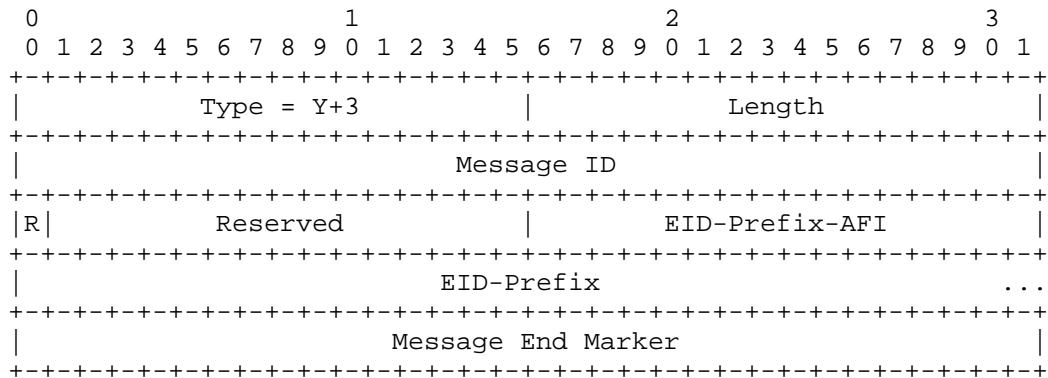


Registration Rejected message format

- o Rejection code: Code identifying the reason for which the Map-Server rejected the registration. Codes:
 - * 1 - Not a valid site EID prefix.
 - * 2 - Authentication failure.
 - * 3 - Locator set not allowed.
- o EID-Prefix AFI: Address family identifier for the EID prefix in the following field.
- o EID-Prefix: The EID prefix from the received Registration.

6.1.4. Registration Refresh Message

Sent by the Map-Server to the ETR to request the re-transmission of EID prefix database mapping Registration messages.

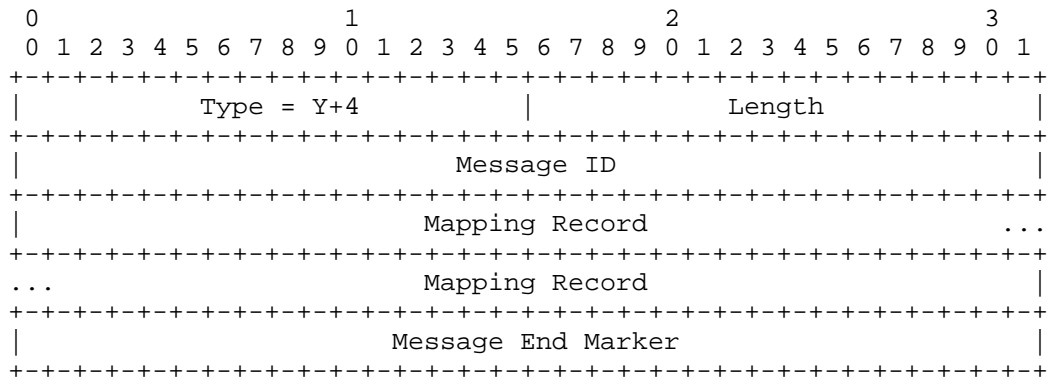


Registration Refresh message format

- o R: Request from the ETR to only refresh registrations that have been previously rejected by the Map-Server.
- o EID prefix, and its more specifics, to refresh. The prefix can be in LCAF format allowing specification of a complete refresh (unspecified prefix), refresh of all the prefixes under an EID instance or even of more specific registrations under a specific EID prefix.

6.1.5. Mapping Notification Message

Mapping Notification messages communicate the Map-Server view of the mapping for an EID prefix and no longer serve as a registration acknowledgement. Mapping Notifications do not need message level authentication as they are received over a reliable transport session to a known Map-Server. Note that reliable transport Mapping Notification messages do not reuse the UDP Map-Notify message format.



Registration message format

6.2. ETR Behavior

The ETR operates the following per EID prefix, per MS state machine that defines the reliable transport EID prefix registration behavior.

There are five states:

- o No state: The local EID database prefix does not exist.
- o Periodic: The local EID database prefix is being periodically registered through UDP Map-Register messages as specified in [1].
- o Stable: From the ETR's perspective, no registrations are due to be sent to the peer. The session to the peer is up, and the peer has either acknowledged the registration, or is expected to request a refresh in the future.
- o AckWait: A Registration message for the prefix has been transmitted to the Map-Server and the ETR is waiting for either a Registration Acknowledge or Registration Rejected reply from the Map-Server.
- o Reject: The reliable transport registration for the local EID database prefix was rejected by the Map-Server. From the ETR's perspective, no registration is due to the peer AND the peer is known to have rejected the registration.

The following events drive the state transitions:

- o DB creation: The local EID database entry for the EID prefix is created.

- o DB deletion: The local EID database entry for the EID prefix is deleted.
- o DB change: The mapping contents or authentication information for the local EID database entry changes.
- o Session up: The reliable transport session to the Map-Server is established.
- o Session down: The reliable transport session the Map-Server goes down.
- o Recv Refresh: A Registration refresh message is received from the Map-Server.
- o Recv ACK: A Registration Acknowledge message is received from the Map-Server.
- o Recv Rejected: A Registration Rejected message is received from the Map-Server.
- o Periodic timer: The timer that drives generation of periodic UDP Map-Register messages fires.

The state machine is:

Event	Prev State	
	No state	Periodic
DB creation [session down]	-> Periodic A1	N/A
DB creation [session up]	-> AckWait A2	N/A
DB deletion	N/A	-> No state A3
DB change	N/A	- A1
Session up	-	-> Stable A4
Session down	-	N/A
Recv Refresh	-	N/A
Recv Refresh [rejected]	-	N/A
Recv ACK	-	N/A
Recv Rejection	-	N/A
Timer	N/A	- A5

xTR per EID prefix per MS state machine

Event	Prev State		
	Stable	AckWait	Rejected
DB creation	N/A	N/A	N/A
DB deletion	-> No state A6	-> No state A6	-> No state
DB change	-> AckWait A2	- A2	-> AckWait A2
Session up	N/A	N/A	N/A
Session down	-> Periodic A7	-> Periodic A7	-> Periodic A7
Recv Refresh	-> AckWait A2	- A2	-> AckWait A2
Recv Refresh [rejected]	-	- A2	-> AckWait A2
Recv ACK	-	-> Stable	-> AckWait A2
Recv Rejection	-> Rejected	-> Rejected	-
Timer	N/A	N/A	N/A

xTR per EID prefix per MS state machine

Action descriptions:

- o A1: Start periodic registration timer with zero delay.
- o A2: Send Registration over reliable transport session.
- o A3: Send UDP registration with zero TTL.
- o A4: Stop periodic registration timer.

- o A7: Send UDP registration and start periodic registration timer with registration period.
- o A6: Send Registration with TTL zero over reliable transport session.
- o A7: Start periodic registration timer with registration period.

All timer start actions must be jittered.

When the reliable transport session is established the state machine moves into the Stable state without first registering the EID prefix over the reliable transport session. The subsequent refresh issued by the Map-Server will trigger the registration message to be sent. This model will allow future optimisations where the Map-Server may retain registration state from a previous instantiation of the reliable transport session with the ETR and only request the refresh of EID prefix state beyond some negotiated session progress marker.

Aa Map-Server authentication key change is treated as a DB change event and will result in triggering a new Registration message to be transmitted.

6.3. Map-Server Behavior

Received registrations create/update or delete mapping state.

A refresh for an unspecified prefix is sent when a session is first established to obtain the complete database contents from the ETR.

Refresh for rejected registrations sent (R bit set) when a new EID prefix is configured on the Map-Server.

Rejection sent to the ETR when an EID prefix that is registered is deconfigured.

Rejected Refresh (R bit set) sent when authentication for an EID prefix changes followed by a Rejection for existing registrations which fail authentication following change.

Mapping Notification message sent whenever the mapping for a registered or more specific prefix for which notifications are requested changes. ETR acknowledgement or rejection messaging for Mapping Notification is not required because the ETR decides how to process the message based on the registered mapping information. If the mapping information changes the resulting registration will trigger a new Mapping Notification message from the Map-Server.

7. Security Considerations

The LISP reliable transport session SHOULD be authenticated. On controlled RLOC networks that can guarantee that the source RLOC address of data packets cannot be spoofed, the authentication check can be a source address validation on the reliable transport packets. When the RLOC network does not provide such guarantees, reliable transport authentication SHOULD be used. Implementations SHOULD support the TCP Authentication Option (TCP-AO) [RFC5925] and SCTP Authenticated Chunks [RFC4895].

8. IANA Considerations

TCP port 4342 already reserved for LISP CONS and not used. Need to reserve a SCTP port. LISP reliable transport message types to be allocated by IANA.

9. Acknowledgments

The authors would like to thank Noel Chiappa, Dino Farinacci, Jesper Skriver, Johnson Leong, Andre Pelletier and Les Ginsberg for their contributions to this document.

10. Normative References

- [I-D.ietf-lisp-lcaf]
Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format (LCAF)", draft-ietf-lisp-lcaf-04 (work in progress), January 2014.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, January 2013.
- [RFC6833] Fuller, V. and D. Farinacci, "Locator/ID Separation Protocol (LISP) Map-Server Interface", RFC 6833, January 2013.

Authors' Addresses

Chris Cassar
Cisco Systems
10 New Square Park
Bedfont Lakes, Feltham TW14 8HA
United Kingdom

Email: ccassar@cisco.com

Isidor Kouvelas
Cisco Systems
Monumental Plaza, Building C
44 Kifissias Ave.
Maroussi, Athens 15125
Greece

Email: kouvelas@cisco.com

Darrel Lewis
Cisco Systems
Tasman Drive
San Jose, CA 95134
USA

Email: darlewis@cisco.com

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: August 18, 2014

V. Moreno
F. Maino
D. Lewis
M. Smith
S. Sinha
Cisco Systems
February 14, 2014

LISP Deployment Considerations in Data Center Networks
draft-moreno-lisp-datacenter-deployment-00

Abstract

This document discusses scenarios and implications of LISP based overlay deployment in the Data Center. The alternatives for topological location of the different LISP functions are analyzed in the context of the most prevalent Data Center topologies. The role and deployment of LISP in the Wide Area Network and Data Center Interconnection are also discussed.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

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 <http://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 August 18, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. LISP in the Data Center	3
2. Definition of Terms	4
3. Data Center Network Reference Topologies	4
4. LISP Deployment in Data Center Fabric Topologies	6
4.1. xTR Topological Placement	7
4.1.1. vLeaf nodes as xTRs	7
4.1.2. Disjoint xTR function: vLeaf to Leaf-proxy function separation	7
4.1.3. Full LISP xTR function at Leaf node	8
4.2. Mapping System topological placement	9
4.2.1. Map-Server/Resolver out of band	9
4.2.2. Map-Server/Resolver in-line	10
5. LISP deployment in the DC WAN and Data Center Interconnect	12
5.1. Fabric-WAN handoff: topology, peering and administrative delineation	13
5.1.1. Two-tier Fabric-WAN normalized handoff	13
5.1.2. Single-tier Fabric-WAN handoff	14
5.2. Fabric to WAN interoperability scenarios	14
5.2.1. LISP Fabric to LISP WAN interoperability with common RLOC space	14
5.2.2. LISP Fabric to LISP WAN interoperability with disjoint RLOC spaces	15
5.2.3. Non-LISP Fabric to LISP WAN interoperability	15
5.2.4. LISP Fabric to Non-LISP WAN interoperability	16
6. Security Considerations	17
7. IANA Considerations	17
8. Acknowledgements	17
9. References	17
9.1. Normative References	17
9.2. Informative References	17
Authors' Addresses	19

1. LISP in the Data Center

Data Center Networks require support for segmentation, mobility and scale as described in [I-D.ietf-nvo3-overlay-problem-statement]. These requirements can be addressed with the use of overlay networks.

The LISP [RFC6830] control plane can be used for the creation of network overlays that support mobility and segmentation at large scale in Layer 2, Layer 3 and combined Layer2/Layer3 overlays as described in [I-D.maino-nvo3-lisp-cp] and [I-D.hertoghs-nvo3-lisp-controlplane-unified].

The needs for overlay provided services are typically different within and across Data Centers versus the needs for Data Center Wide Area Network connectivity. LISP is relevant in both of these areas.

The use of LISP as a control protocol for the creation of overlays can be optimized for the specific topologies that are relevant in the context of Data Center Networks within and across Data Centers. This document discusses different deployment options for LISP in the context of different data center topologies, the implications of the use of the different deployment options are part of the discussion.

Data Center Networks include an intra-Data-Center fabric topology as well as inter-Data-Center and Wide Area Network components. The topologies found inside the Data Center Network fabric are designed in a deterministic manner with a very high degree of symmetry. This high degree of symmetry assures that a multitude of paths exist between any two end-points in the data center and that these paths are of equal cost and deterministic latency. As a result of such designs, traffic patterns within or across the data center are guaranteed to traverse specific tiers of the network. A reference Data Center Network inclusive of an intra-Data-Center fabric topology as well as inter-Data-Center and Wide Area Network components is described in Section 3.

The predictability of the different possible data paths in the Data Center Network opens opportunities for optimizations in the deployment of demand based control plane models such as LISP. In the specific case of LISP, some of the interesting deployment options are centered around the topological location of the mapping-system and xTRs, equally interesting is the option of separating the LISP control-plane from the encapsulation/decapsulation functions in an xTR. Different deployment scenarios are discussed in Section 4 and Section 5.

2. Definition of Terms

Proxy Reply Mode: Map Servers may reply to a map-request directly, rather than forwarding the map-request to the registering ETR

Fabric Manager: Orchestration tool in charge of provisioning and monitoring Fabric Network nodes.

WAN Manager: Orchestration tool in charge of provisioning and monitoring WAN/DCI Network nodes.

For definition of NVO3 related terms, notably Virtual Network (VN), Virtual Network Identifier (VNI), Network Virtualization Edge (NVE), Data Center (DC), please consult [I-D.ietf-nvo3-framework].

For definitions of LISP related terms, notably Map-Request, Map-Reply, Ingress Tunnel Router (ITR), Egress Tunnel Router (ETR), Map-Server (MS) and Map-Resolver (MR) please consult the LISP specification [RFC6830].

3. Data Center Network Reference Topologies

The reference topology that will be used for our discussion is a folded clos topology. The folded clos is considered to be the prevalent topology in large-scale data centers requiring the use of overlays. Although other topologies may be utilized within the data center, most of such topologies may be modeled as a folded clos or collection of clos topologies.

The reference clos topology is illustrated in Figure 1. The connectivity depicted in the diagram is not exhaustive; every Leaf node has at least one connection to every Spine node, effectively providing at least N equal cost paths between any two Leaf nodes, where N is the number of spine nodes.

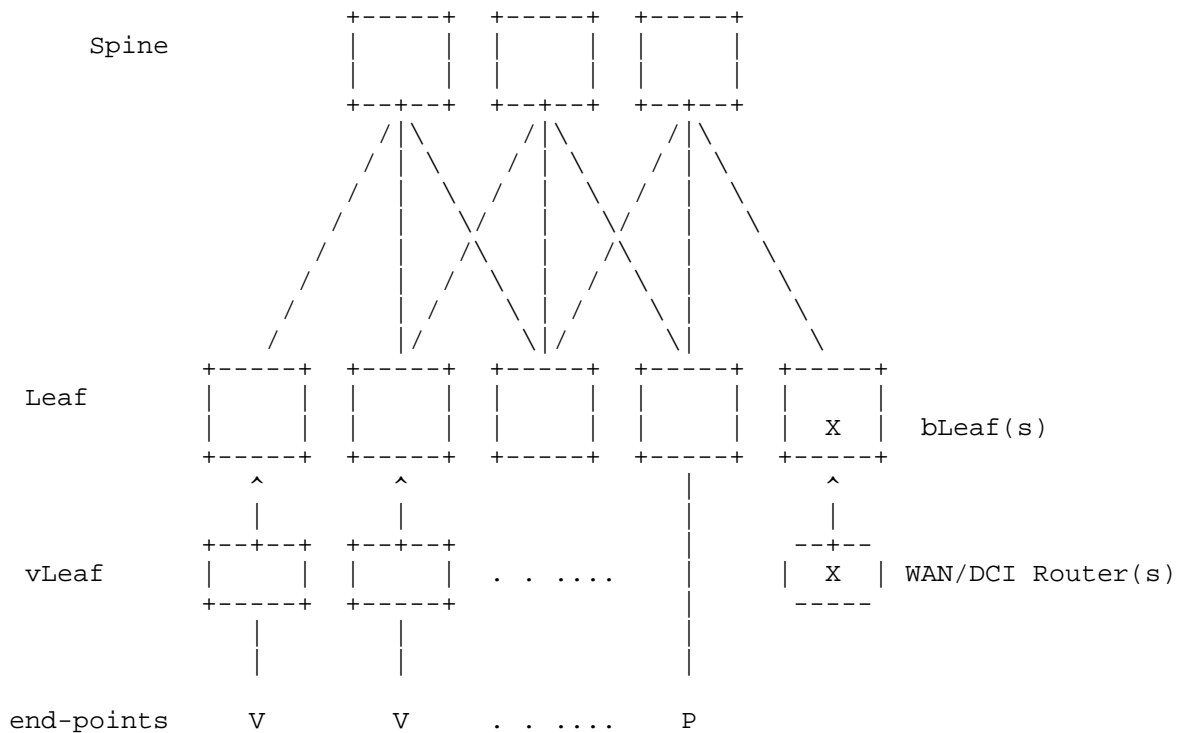


Figure 1: Folded Clos Data Center Topology

The clos topology is a multi-stage topology. In the folded clos instantiation, there is an ingress stage, a middle stage and an egress stage. The elements of the topology are described as:

- o Spine Nodes: Nodes that make the middle stage of the clos. In the absence of failures, each Spine Node is connected to every Leaf Node.
- o Leaf Nodes: Nodes that make the ingress and egress stage of the clos. Each Leaf Node is connected to every Spine Node. In the case of a link or Spine Node failure, a Leaf Node will remain connected to the surviving Spine Nodes over the remaining healthy links. The fabric will continue to operate under such condition of partial connectivity.
- o vLeaf Nodes: Virtual Leaf Nodes are those network nodes that connect to the Leaf Nodes. vLeaf Nodes are usually in the form of

virtual switches/routers running on the physical compute end-points hosts. Many vLeaf nodes may connect to a single Leaf node. Likewise, a vLeaf node may connect to multiple Leaf nodes for redundancy purposes.

- o WAN Routers: Routers attached to leaf nodes in the clos topology to provide connectivity outside the clos over a WAN or Data Center Interconnect (DCI). WAN Routers are usually deployed in pairs for resiliency and they are also usually attached to at least two Leaf nodes in a redundant topology. WAN routers participate in a Wide Area Network (WAN) or Data Center Interconnect (DCI).
- o bLeaf: Border Leaf nodes are Leaf nodes that attach to the WAN routers.
- o WAN: The Wide Area Network is the network over which client sites as well as Disaster Recovery sites connect to the data center.
- o DCI: Data Center Interconnect is a Metropolitan Area Network over which multiple Data Centers connect to each other. There is an assumption of bound latency over the DCI.
- o end-points: Hosts that connect to the Data Center Fabric. End-points can be physical (P) or virtual (V)
- o V: Virtual end-points such as Virtual Machines.
- o P: Physical end-points such as non-virtualized servers.

4. LISP Deployment in Data Center Fabric Topologies

The main LISP functions to be deployed in the Data Center fabric topology are:

- o xTRs
- o Map-Servers
- o Map-Resolvers

The Map-Server and Map-Resolver functions will generally be co-located on the same network nodes, we will refer to the co-located function as Map-Server/Resolver.

4.1. xTR Topological Placement

LISP xTRs perform both Data Plane as well as Control Plane tasks. From the Data Plane perspective the xTRs are tasked with encapsulating (ITRs) or decapsulating (ETRs) traffic. From the Control Plane perspective the tasks of registering mappings and requesting/caching mappings are handled by the ETRs and ITRs respectively.

It is possible to split the roles of ETR and ITR to different network nodes. The most prevalent deployment model combines the roles of ETR and ITR onto a single device. We will focus our discussion on the combined ETR/ITR model. A device that serves as both ETR and ITR is generally referred to as an xTR.

It is also possible to split the Data Plane and Control Plane responsibilities and distribute them across different network nodes. Some of the deployment options enabled by this separation are discussed in the following sections:

4.1.1. vLeaf nodes as xTRs

The full xTR function including both Control and Data Plane can be deployed at the vLeaf nodes. The entire clos including Leaf nodes and Spine nodes simply serves as an underlay transport to the LISP overlay.

The advantage of distributing the xTR role to the vLeaf is mainly that the map-cache state at each xTR is minimized by virtue of it being distributed to a larger number of nodes.

Amongst the challenges are:

- o Operational implications of managing a larger number of overlay end-points.
- o Large number of xTRs sending requests and registrations to the mapping system has implications on the scale requirements and limits of the mapping system (servers and resolvers)
- o vLeaves do not service non-virtualized Physical end-points

4.1.2. Disjoint xTR function: vLeaf to Leaf-proxy function separation

The control and data plane tasks can be separated and distributed amongst Leaf and vLeaf. In this approach the vLeaf can retain the Data Plane encaps/decap function while the LISP signaling is done by

the Leaf. Thus, the Leaf will proxy the signaling for the vLeaf nodes connected to it as follows

- o Map-register messages will be issued by the Leaf with the corresponding RLOC addresses of the vLeaf nodes.
- o Map-request messages will be sent by the ITR vLeaf to its upstream Leaf and then relayed by the Leaf to the Map-resolver on behalf of the vLeaf.
- o The mapping system, when not in proxy reply mode, will forward the map-request directly to the RLOC of the vLeaf node with which the EID being requested was registered.
- o Map-reply messages will be sent to the requesting Leaf and then relayed to the corresponding vLeaf where the mapping is cached.

This scheme allows the caching of state to remain fully distributed and also enables more granular distribution of the mappings amongst a larger number of ETRs. The scheme also allows the consolidation of the signaling at the Leaf nodes where map-registers and map-requests can be batched. The scheme does however present the challenge of maintaining mappings in the mapping system for an increased number of xTRs since it will track the larger number of xTRs at the vLeaf level rather than the smaller number of xTRs at the Leaf level, this can be alleviated by fully proxying the vLeaf LISP functions at the Leaf as discussed in Section 4.1.3.

4.1.3. Full LISP xTR function at Leaf node

Both the data plane and control plane xTR function may reside on the Leaf nodes. This deployment model leverages the LISP control plane without any changes and enables interworking of different hypervisor overlays across a normalized LISP fabric.

vLeaf nodes may encapsulate traffic into the Fabric in different ways (802.1Q, VXLAN, NVGRE, etc) depending on the hypervisor they may use. It is the role of the Leaf node, in this model, to terminate any encapsulation being used by the vLeaf nodes, extract any metadata information, such as the VNID [I-D.ietf-nvo3-framework], from the headers imposed at the vLeaf nodes and normalize the different encapsulations received from different vLeaf nodes to a common LISP encapsulation amongst Leaf nodes. The LISP encapsulation at the Leaf nodes will encode the VNIDs extracted from the encapsulated vLeaf traffic as instance-ids in the LISP overlay.

The mapping of the VNIDs imposed by the vLeaf nodes to LISP instance-ids is provisioned at each Leaf/xTR node either manually or by an

automated provisioning system linked to an orchestration system with visibility into the different hypervisor overlays and the LIISP overlay.

The ability to normalize different hypervisor overlays is predicated on the use and exchange of Universally Unique Identifiers (UUIDs) as the constant parameter identifying a common segment across different overlay administrative domains that may use a variety of VNIDs to encapsulate traffic for the same segment. The assignment and administration of tenant UUIDs is an important orchestration task and outside the scope of this document.

4.2. Mapping System topological placement

Within the data center, we will assume that the functions of map-server and map-resolver are co-located on the same nodes. We will refer to such nodes as Map-Server/Resolver.

Resiliency is important in the deployment of Map-Server/Resolvers for the Data Center. Multiple Map-Server/Resolvers must be deployed for redundancy. Depending on the topological location of the Map-Server/Resolver nodes, there may be different deployment requirements to achieve appropriate reachability to the multiple Map-Server/Resolvers involved.

4.2.1. Map-Server/Resolver out of band

The Map-Server/Resolver function can be deployed out of band on one or more network nodes that are either reachable in the WAN or attached to a leaf as end-points but are not part of the clos topology. Out of band Map-Server/Resolvers will only be exposed to control plane traffic.

Clusters of Map-Server/Resolver nodes can be deployed by having ETRs register to multiple Map-Server addresses and ITRs issue map-requests to an anycast address that is shared by the map-resolver function of the Map-Server/Resolver nodes in the cluster. Thus, the map-resolver function can use a common anycast IP address across all Map-Server/Resolver nodes, separate unicast IP addresses will be assigned the map-server component in each node. ETRs registering to the map-servers should be configured to send map-registers to all map-server IP addresses. ITRs should issue map-requests to the shared anycast IP address of the map-resolver component of the cluster.

Alternatively, clusters of Map-Server/Resolver nodes can be deployed by having ETRs register to one Map-Server and use an alternate mechanism to allow the synchronization of this registration across all Map-Servers in the cluster. In this case ETRs will be configured

to register to a single IP address, this would likely be a shared anycast IP address amongst the Map-Servers in order to ensure resiliency. ITRs will continue to issue requests to a Map-Resolver shared anycast IP.

The creation of Map-Server/Resolver resiliency clusters based on IP addressing is a topologically agnostic deployment model which fits the out of band model well and gives a high degree of flexibility for the deployment of the necessary mapping system nodes.

4.2.2. Map-Server/Resolver in-line

The Map-Server/Resolver function can be deployed in-line on one or more network nodes that participate in the clos topology either as Spine nodes or Leaf nodes. In-line Map-Server/Resolvers will receive both Control Plane and Data Plane traffic.

4.2.2.1. Map-Server/Resolver at the Spine

The Map-Server/Resolver function can be deployed on Spine nodes. In order to guarantee resiliency, Map-Server/Resolvers are deployed on at least two Spine nodes, but preferably on all Spine nodes.

Any map-register messages must be sent to all Map-Server/Resolver enabled Spine nodes in order to ensure synchronization of the mapping system. Therefore in a system where all Spine nodes host the Map-Server/Resolver functionality, all Spine nodes will have complete mapping state for all EIDs in the fabric. Alternatively, map-register messages can be sent to a single Map-Server and the state may be relayed by other methods to other Map-Servers in the Spine.

When all Spine nodes host the Map-Server/Resolver functionality, all data plane traffic that cuts across different leaf nodes will traverse a Spine node that has the full LISP mapping state for the fabric. In this deterministic topology it is possible to implement avoid transient drops that may occur when looking up destinations that have not been previously cached at the ITRs.

In order to avoid transient drops during Mapping System lookups, ITRs (Leaf or vLeaf nodes) without a pre-existing cache entry for a particular destination must encapsulate and forward the traffic with the unknown destination to the Spine. Since the Spine has full location information, it knows which ETR to send the traffic to, so it encapsulates and forwards the traffic accordingly. Simultaneously, the ITR may send a map-request to the anycast address for the map-resolvers that are present across the spine. The ITR will continue to send the traffic for the unknown destination towards the Spine until a mapping for the destination is received in a map-

reply and cached at the ITR, at which point the ITR will start encapsulating traffic directly to the appropriate ETR.

In order to forward traffic for unknown destinations to the Spine, the ITR must LISP encapsulate the unknown destination traffic towards the anycast IP address configured for the map-resolver function on the Spine nodes or it may hash the traffic to select a discrete RLOC for a specific Spine node. It is important that traffic be LISP encapsulated in order to preserve the semantics of instance-id and other parameters that may be encoded in the LISP header.

An alternative implementation could leverage the data plane traffic in order to reduce the amount of control plane messages that are exchanged. For instance, when traffic with unknown destinations is sent to the spine, rather than waiting to receive a map-request for the unknown destination, the spine could react to the reception of the unknown destination data plane traffic by sending a map-reply to the source ITR with the mappings for the corresponding unknown destination. This effectively replaces the map-request messaging with data plane events. Either way this is implemented, the main benefit of the deployment model is the ability to avoid packet drops while mapping system lookups are completed.

4.2.2.2. Map-Server/Resolver at Leaf nodes

The Map-Server/Resolver function could be deployed at Leaf nodes. In this scenario it may be beneficial to make the different Leaf hosted Map-Server/Resolvers authoritative for specific prefixes or instance-ids (VNIs) and allow the distribution of the mapping state across the different Leaf nodes.

One mechanism to achieve the distribution of the mapping state across the different leaf nodes is to have different Leaf nodes be the authority for a particular prefix or instance-id in a Delegated Database Tree (DDT)[I-D.ietf-lisp-ddt]. The Leaf nodes can be arranged in pairs for redundancy and the association of a particular prefix or instance-id to specific Leaf nodes is achieved by configuration of the Leaf nodes and the DDT.

This type of in-line deployment of the Map-Server/Resolver could, in theory, also provide a no-drop LISP lookup service at the expense of maintaining all LISP mapping state at all Leaf nodes. In the case of a no-drop LISP lookup service, the Map-Server/Resolver function would be deployed in the same way as explained for deployment in the Spine and the system would therefore not benefit from the distribution of state at the Leaf nodes.

5. LISP deployment in the DC WAN and Data Center Interconnect

The placement of IP workloads within and across Data Centers has a high degree of entropy, which renders existing topology congruent routing summarization methods ineffective in containing the explosion of routing state in the Data Center WAN and Data Center Interconnect. The problem of entropy in the placement of workloads across Data Centers is analogous to the challenge of prefix disaggregation found on the Internet. The seminal motivation behind the development of LISP was the mitigation of the scale issues caused by the disaggregation of user prefixes in the Internet, the multi-Data Center problem is of similar nature, hence LISP is a very good fit to address such scale problem.

In the WAN, LISP will be in charge of handling reachability information between Branch sites and IP workloads that are distributed across different Data Centers. The use of LISP allows the scalable handling of very granular location information as IP workloads are deployed in diverse Data Centers. The management domain is usually a single one for the LISP WAN environment and the diverse Data Centers involved may be managed independently. The diversity of management domains is a key deployment consideration which is found in the WAN and DCI connectivity scenarios and requires a certain level of federation between the different domains as will be discussed in the next few sections.

In the Data Center Interconnect (DCI), LISP provides reachability and policies (such as inbound load-balancing) between Data Centers for those cases in which IP workloads must communicate with each other across Data Centers. The communication may happen between Data Centers in independent administrative domains, with a single administrative domain being a more specific instantiation of the general case.

In the WAN and DCI, as inside the Data Center, the main LISP functions to deploy are:

- o xTRs
- o Map-Servers
- o Map-Resolvers
- o PxTRs

Map-Servers/Map-Resolvers will be distributed across Data Centers and over the WAN following the DDT model. DDT prefix delegation provides

a way to partition different administrative domains, providing a very basic form of federation.

xTRs will normally be deployed at the edge of the Data Center on the WAN/DCI routers.

PxTR location may vary, but in general will be placed as close as possible to Internet gateways, or other non LISP-speaking sources. See [RFC6832] for considerations on Interworking with non-LISP sites.

Particularly interesting in the deployment of LISP is the type of handoff between the Data Center Fabric and the devices providing the WAN/DCI services. Different handoff options are discussed in Section 5.1.

Of further relevance is the careful evaluation of the different interoperability scenarios between the Data Center Fabric and the WAN /DCI that are discussed in Section 5.2.

5.1. Fabric-WAN handoff: topology, peering and administrative delineation

There are a couple of approaches to realizing the handoff between the Fabric and the WAN:

- o Two-tier: Border-Leaf to WAN-router peering
- o Single-tier: Consolidated border-Leaf/WAN-router function

5.1.1. Two-tier Fabric-WAN normalized handoff

In the two-tier approach the border-Leaf and WAN-router peer over a normalized handoff. There is clear administrative delineation at the handoff between Fabric and WAN where the Fabric Manager is authoritative solely over the border-Leaf and the WAN Manager is authoritative solely over the WAN-router. The border-Leaf and WAN router may enter a peering relationship and exchange routes and other attributes of their respective overlay services. Segments (VNs) in the Fabric overlay will map at the border-Leaf to a normalized segment over the handoff that can be realized by use of VRFs or VLANs interconnected over an 802.1Q trunked set of links or over an overlay encapsulation such as VXLAN, GRE, NVGRE, MPLS, LISP or other. These segments will in turn be mapped at the WAN-router to their corresponding segments in the WAN service (for example, a VRF in an IP MPLS VPN or a VRF/instance-id segment in LISP). Reachability and other information can be exchanged between the border-Leaf nodes and the WAN routers over the normalized handoff using an extensible routing protocol such as MP-BGP or IS-IS.

5.1.2. Single-tier Fabric-WAN handoff

In the single tier approach, the border-Leaf and the WAN-router are the same device. Mapping of segments (VNs) in one domain to segments in the other domain is achieved by sharing certain components between the two domains. One good example is a VRF that routes traffic between the Fabric overlay domain and the WAN overlay domain at the consolidated border-Leaf/WAN-router device.

The device that consolidates the border-Leaf and WAN-router roles is managed by two entities: the Fabric manager and the WAN manager. Delineation of management authority must be established carefully and precisely to separate WAN relevant elements within the device from Fabric relevant elements and their corresponding administration. There will be a series of common components where the mapping between the domains happens (e.g. VRFs). Policy to resolve any conflicts related to the administration of common components in the handoff must be in place.

5.2. Fabric to WAN interoperability scenarios

5.2.1. LISP Fabric to LISP WAN interoperability with common RLOC space

To integrate a LISP based Fabric with the LISP WAN it is possible to merge the Fabric LISP domain with the WAN LISP domain. To achieve this, the Map-Servers/Resolvers for the Fabric can join the DDT structure in the WAN.

The merging of the domains assumes that the Fabric ITR/Leaf nodes are reachable in the underlay (RLOC space) from the WAN. In this model the xTRs are the Fabric Leaf nodes as well as the Branch Routers at the different WAN sites and an end-to-end overlay is realized between xTRs cutting across WAN and Fabric.

Although the domains are merged into one, the hierarchical structure of the DDT provides some isolation of the control plane and failure isolation. The granular state for the Fabric is maintained solely on the Fabric Map-Server/Resolvers and LISP signaling for the Fabric xTRs will be scoped and serviced by the Fabric Map-Server/Resolvers.

The domain isolation achieved with this model may not be sufficient for specific deployments, it is possible to follow the model described for a disjoint RLOC space in Section 5.2.2 even if the RLOC space is contiguous.

5.2.2. LISP Fabric to LISP WAN interoperability with disjoint RLOC spaces

There are scenarios in which the RLOC space in the Fabric is not part of the WAN routing space. In these scenarios the LISP overlay realized inside the Fabric must terminate at the border of the Fabric and should be mapped to a WAN overlay that terminates at the border of the WAN.

The handoff between the two domains may be a single-tier handoff. In this case, the WAN/DCI device and bLeaf are the same device. The WAN/DCI device will be an xTR in two domains: Fabric and WAN/DCI. In one deployment model the xTR may receive Map-Notifications from the WAN Mapping System and these may trigger registrations into the Fabric Mapping System and vice versa. This model basically uses the xTR as an exchange point between the two LISP control plane instances. From the data plane perspective the xTR acts as an RTR (Re-encapsulating Tunnel Router). A different deployment model may rely heavily on Mapping System intelligence and may assume that the mapping systems are federated and that the location of the requesting ITR gives the mapping system enough context to provide map-replies that will steer traffic to the appropriate RTRs as packets progress through the different Data Center edges.

In the disjoint RLOC scenario, the handoff between the two domains may alternatively be a two-tier handoff with separate border-Leaf and WAN-router devices. The exchange of routing information between the bLeaf and WAN-router may be achieved by dynamic routing protocols such as MP-BGP or IS-IS. The routes received by either device must be registered into the respective mapping system. Similarly, any mappings in one domain must be advertised as routes on the handoff to the other domain.

When the LISP implementation does not support redistribution of routes into LISP and the converse redistribution of Mappings into routing protocols, the use of static routes and static mappings can help the interoperability of the two domains.

5.2.3. Non-LISP Fabric to LISP WAN interoperability

In order to support mobility, Data Center Fabrics are either built as pure Layer 2 Fabrics or, when built as Layer 3 networks, they rely on host routing to achieve mobility in the routed Fabric. The implication of the use of Layer 3 in the Fabric is that every Leaf will advertise host routes for the devices that are directly connected to the Leaf and all Leaf nodes will hold complete state for all end-points that attach to the fabric.

When multiple Fabrics are interconnected with a DCI/WAN service, end-points that are reachable outside the fabric can be represented inside the fabric by a default route advertised by the bLeaves into the Fabric. The routing is thus made of specific host routes for end-points attached to the local Fabric and any end-points attached to other Fabrics will not be known specifically, but would be assumed to be reachable via the default route that points to the bLeaf. Another way of describing this model is to simply state that all local end-points are explicitly known and that traffic destined to any unknown destinations will be default forwarded to the bLeaf. The bLeaf can be a LISP xTR (in the single-tier model) and it would receive default routed traffic for which it can issue map-requests and encapsulate to the appropriate destination as the traffic is received. Alternatively, the bLeaf may handoff the traffic to a WAN/DCI router that provides the required xTR functionality.

The advantages of this model are several:

- o Remote state is kept out of the local fabric by using default routing
- o xTRs in the LISP WAN/DCI only resolve and cache mappings for active flows as expected of an on-demand control plane
- o Host routes terminate at the bLeaf and are kept outside of the routing tables in the WAN/DCI as they are handled by LISP instead

5.2.4. LISP Fabric to Non-LISP WAN interoperability

In this scenario, the prefixes handled by LISP inside the Fabric must be advertised to the Non-LISP WAN/DCI. In order to do so, the bLeaf must be able to pull all EIDs that are registered with the Fabric's mapping system and advertise these EIDs in a routing protocol. If the EIDs are easily summarizable, it may be sufficient to simply add a static route at the bLeaf. More generally, the EIDs may not be easily summarizable and may even change dynamically. When the EIDs are not summarizable or static, a mechanism for the bLeaf to subscribe to the mapping system and download all EIDs dynamically is required. Since LISP does not provide such subscription service, one option is to run a push protocol such as BGP between the mapping system and the bLeaf, since all registered EIDs are known at the Map-Server/Resolver, the Map-Server/Resolver may redistribute these EIDs into BGP and the required information may be advertised over BGP to the xTRs on the bLeaves. Another option is for the Map-Server-Resolver to be located at the bLeaves and have the EIDs be redistributed from LISP into a routing protocol on the same device.

6. Security Considerations

[I-D.ietf-lisp-sec] defines a set of security mechanisms that provide origin authentication, integrity and anti-replay protection to LISP's EID-to-RLOC mapping data conveyed via mapping lookup process. LISP-SEC also enables verification of authorization on EID-prefix claims in Map-Reply messages.

Additional security mechanisms to protect the LISP Map-Register messages are defined in [RFC6833].

The security of the Mapping System Infrastructure depends on the particular mapping database used. The [I-D.ietf-lisp-ddt] specification, as an example, defines a public-key based mechanism that provides origin authentication and integrity protection to the LISP DDT protocol.

7. IANA Considerations

This document has no IANA implications

8. Acknowledgements

The authors want to thank Yves Hertoghs for the early review, insightful comments and suggestions.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

9.2. Informative References

[I-D.farinacci-lisp-mr-signaling]
Farinacci, D. and M. Napierala, "LISP Control-Plane Multicast Signaling", draft-farinacci-lisp-mr-signaling-03 (work in progress), September 2013.

[I-D.hertoghs-nvo3-lisp-controlplane-unified]
Hertoghs, Y., Maino, F., Moreno, V., Smith, M., Farinacci, D., and L. Iannone, "A Unified LISP Mapping Database for L2 and L3 Network Virtualization Overlays", draft-hertoghs-nvo3-lisp-controlplane-unified-01 (work in progress), February 2014.

- [I-D.ietf-lisp-ddt]
Fuller, V., Lewis, D., Ermagan, V., and A. Jain, "LISP Delegated Database Tree", draft-ietf-lisp-ddt-01 (work in progress), March 2013.
- [I-D.ietf-lisp-lcaf]
Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format (LCAF)", draft-ietf-lisp-lcaf-04 (work in progress), January 2014.
- [I-D.ietf-lisp-sec]
Maino, F., Ermagan, V., Cabellos-Aparicio, A., Saucez, D., and O. Bonaventure, "LISP-Security (LISP-SEC)", draft-ietf-lisp-sec-05 (work in progress), October 2013.
- [I-D.ietf-nvo3-dataplane-requirements]
Bitar, N., Lasserre, M., Balus, F., Morin, T., Jin, L., and B. Khasnabish, "NVO3 Data Plane Requirements", draft-ietf-nvo3-dataplane-requirements-02 (work in progress), November 2013.
- [I-D.ietf-nvo3-framework]
Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for DC Network Virtualization", draft-ietf-nvo3-framework-05 (work in progress), January 2014.
- [I-D.ietf-nvo3-nve-nva-cp-req]
Kreeger, L., Dutt, D., Narten, T., and D. Black, "Network Virtualization NVE to NVA Control Protocol Requirements", draft-ietf-nvo3-nve-nva-cp-req-01 (work in progress), October 2013.
- [I-D.ietf-nvo3-overlay-problem-statement]
Narten, T., Gray, E., Black, D., Fang, L., Kreeger, L., and M. Napierala, "Problem Statement: Overlays for Network Virtualization", draft-ietf-nvo3-overlay-problem-statement-04 (work in progress), July 2013.
- [I-D.maino-nvo3-lisp-cp]
Maino, F., Ermagan, V., Hertoghs, Y., Farinacci, D., and M. Smith, "LISP Control Plane for Network Virtualization Overlays", draft-maino-nvo3-lisp-cp-03 (work in progress), October 2013.
- [I-D.smith-lisp-layer2]
Smith, M., Dutt, D., Farinacci, D., and F. Maino, "Layer 2 (L2) LISP Encapsulation Format", draft-smith-lisp-layer2-03 (work in progress), September 2013.

- [RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, January 2013.
- [RFC6831] Farinacci, D., Meyer, D., Zwiebel, J., and S. Venaas, "The Locator/ID Separation Protocol (LISP) for Multicast Environments", RFC 6831, January 2013.
- [RFC6832] Lewis, D., Meyer, D., Farinacci, D., and V. Fuller, "Interworking between Locator/ID Separation Protocol (LISP) and Non-LISP Sites", RFC 6832, January 2013.
- [RFC6833] Fuller, V. and D. Farinacci, "Locator/ID Separation Protocol (LISP) Map-Server Interface", RFC 6833, January 2013.
- [RFC6836] Fuller, V., Farinacci, D., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol Alternative Logical Topology (LISP+ALT)", RFC 6836, January 2013.

Authors' Addresses

Victor Moreno
Cisco Systems
170 Tasman Drive
San Jose, California 95134
USA

Email: vimoreno@cisco.com

Fabio Maino
Cisco Systems
170 Tasman Drive
San Jose, California 95134
USA

Email: fmaino@cisco.com

Darrel Lewis
Cisco Systems
170 West Tasman Dr.
San Jose, California 95134
USA

Email: darlewis@cisco.com

Michael Smith
Cisco Systems
170 West Tasman Dr.
San Jose, California 95134
USA

Email: michsmit@cisco.com

Satyam Sinha
Cisco Systems
170 West Tasman Dr.
San Jose, California 95134
USA

Email: satysinh@cisco.com

LISP Working Group
Internet-Draft
Intended status: Experimental
Expires: August 11, 2014

A. Rodriguez-Natal
A. Cabellos-Aparicio
Technical University of Catalonia
S. Barkai
ConteXtream, Inc.
V. Ermagan
D. Lewis
F. Maino
Cisco Systems
D. Farinacci
lispers.net
February 7, 2014

Software Defined Networking extensions for the Locator/ID Separation
Protocol
draft-rodrigueznatal-lisp-sdn-00

Abstract

This document describes extensions for the Locator/ID Separation Protocol (LISP) to make it more suitable to be used on Software Defined Networking (SDN) scenarios.

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 <http://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 August 11, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	3
2. Definition of terms	3
3. Overview	4
4. Protocol operation	4
4.1. LISP tunnel routers	4
4.2. Mapping System	5
5. Extended-EID types	5
5.1. 5-tuple	5
6. Extended-EID lookups	5
6.1. 5-tuple lookup	5
7. Mapping updates	6
7.1. Proactive update pushing	6
7.2. Mapping subscription	6
8. Provisioning and Discovery	6
9. Acknowledgements	6
10. IANA Considerations	6
11. Security Considerations	6
12. Normative References	6
Authors' Addresses	7

1. Introduction

The Locator/ID Separation Protocol LISP [RFC6830] splits current IP addresses in two different namespaces, Endpoint Identifiers (EIDs) and Routing Locators (RLOCs). LISP uses a map-and-encap approach that relies in two entities, the Mapping System and the Ingress/Egress Tunnel Routers (xTRs). The Mapping System is a distributed database that stores and disseminates EID-RLOC bindings. The xTRs are deployed at LISP sites edge points and perform encapsulation and decapsulation of LISP data packets.

With this architecture in place, LISP is inherently decoupling control-plane from data-plane. LISP moves all control onto the Mapping System, while keeps data at the xTR level. This decoupling entitles network operators to build a Software Defined Network on top of LISP.

However, vanilla LISP offers a limited feature set on terms of SDN requirements. To position LISP as the foundations for a SDN solution, advanced interaction between LISP elements and some extensions to the stock protocol can be defined. This document describes SDN extensions for LISP.

On the present iteration of this draft, the LISP protocol operating in a SDN deployment manages network traffic in terms of flows identified by a 5-tuple identifier. 5-tuples are encoded in a specific type of LISP Canonical Address Format (LCAF). Flows are routed over the network using Explicit Locator Paths (ELPs). The Mapping-System stores 5-tuple - ELP bindings. Network functions (i.e. firewalls, etc) can be deployed at Re-encapsulating Tunnel Routers (RTRs) spread over the network. These RTRs can be included as part of a ELP.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Definition of terms

- o n-tuple: The term n-tuple is used in this document to describe the set of n elements present in a data packet (e.g. IP address, port, protocol) that can be used to identify unequivocally a packet or a set of packets.
- o 5-tuple: The term 5-tuple is used in this document to describe the set comprised by 5 elements, being these the source IP address, the destination IP address, the level 4 protocol number, the level 4 protocol source port and the level 4 protocol destination port of a data packet.
- o Extended-EID: This document uses the term Extended-EID to refer to any n-tuple (including a 5-tuple) used in a EID role.
- o Flow: The term flow is used in this document to refer to the sequence of packets identified by the same n-tuple.

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

3. Overview

This document describes extensions to LISP protocol definition and operation to optimize it to work on SDN scenarios.

Protocol operation follows the specification defined on [LISP] except for the following. Besides of IP to IP mappings, Mapping System stores also Extended-EID to ELP mappings. Being Extended-EID a n-tuple identifying a flow. LISP routers perform look-ups based on these Extended-EIDs, instead of on destination IPs. Apart from using n- tuples instead of IPs, retrieving information from the Mapping System follows LISP standard mechanisms (i.e. Map-Request, Map-Reply).

Traditionally ETRs register EID-prefixes that include their own RLOC addresses as well as other RLOCs for ETRs at the same site. Here a third-party will also register Extended-EID-to-ELP bindings.

4. Protocol operation

4.1. LISP tunnel routers

LISP routers (xTRs, RTRs) behave as specified on [RFC6830] and [RFC6833], except for the following. LISP routers perform mapping lookups based on Extended-EID (n-tuple) not on IP address EID and they obtain an ELP instead of an IP address RLOC. Which specific n-tuple lookup to use and how to configure the router to use it, is to be covered on future iterations of this document.

Any LISP router must keep an internal map-cache indexed by Extended-EIDs. When a LISP router receives a packet to encapsulate, it extracts the fields required by the n-tuple lookup in use and stores them in an Extended-EID structure. In the case of a 5-tuple lookup, it will extract the source address, destination address, level 4 protocol, source port (if any) and destination port (if any) from the packet. The LISP router uses the Extended-EID to perform a look-up into the map-cache. The map- cache can contain entries with an Extended-EID more coarse in some fields. The lookup process must follow the procedure described in section Section 6. If there is an entry on the map-cache that matches the Extended-EID, the LISP router retrieves the mapping information (i.e. the ELP) and uses the first hop (if it is an ITR) or the next hop (if it is a RTR) of the ELP to encapsulate and forward the packet.

If the map-cache of the xTR contains no entry for the Extended-EID, the xTR sends a Map-Request to the Mapping System. This Map-Request carries the Extended-EID (encoded in the specific LCAF for that Extended-EID type) in the EID-prefix field of the Map-Request. The

Mapping System must reply with a Map-Reply carrying on the locator field an ELP. This Map-Reply can carry on the EID-prefix field an Extended-EID more coarse in some fields, but covering the original Extended-EID. The LISP router must store this Extended-EID entry (even if more coarse) in its map-cache.

4.2. Mapping System

Mapping System (comprising Map Servers and Map Resolvers) behaves as specified on [RFC6830] and [RFC6833], except for the following. It also stores mappings indexed by Extended-EID. These mappings contain n-tuple to ELP mappings.

Map Resolvers must be capable of processing Map-Requests with an Extended-EID on the EID-prefix field. The Extended-EID carried on the Map-Request contains fully qualified most specific values on all its fields. Map Servers can store more coarse Extended-EID entries. Map Resolvers must be capable of finding the Map-Server containing the longest match Extended-EID entry, according to the lookup rules described in section Section 6. Once found, the Map Resolver forwards the Map-Request to the Map Server. The Map Server replies itself to Map-Requests. It must not forward Map-Requests comprising Extended-EIDs to any ITRs.

LISP elements must perform the mapping update mechanisms defined in [RFC6830] (e.g, SMR) using as EID the Extended-EID.

5. Extended-EID types

Possible Extended-EID types and the LCAFs to support them.

5.1. 5-tuple

The 5-tuple LCAF is the combination of LCAF types 4 and 12.

6. Extended-EID lookups

This section describes the lookup process to be followed when using Extended-EID instead of vanilla IP address EID. At this point, this document only covers 5-tuple kind of Extended-EID lookups. It is expected to include lookup mechanism for n-tuple lookups with more complex protocol combinations.

6.1. 5-tuple lookup

TBD more specific / exact match lookup process. TBD address, port, protocol preference.

7. Mapping updates

Advanced mapping update mechanisms to support SDN scenarios.

7.1. Proactive update pushing

MS can send proactive SMRs carrying Map-Reply information to some LISP devices whenever there is a mapping update.

7.2. Mapping subscription

LISP devices can be subscribed or subscribe themselves to specific mappings to get updates whenever these change.

8. Provisioning and Discovery

9. Acknowledgements

10. IANA Considerations

This memo includes no request to IANA.

11. Security Considerations

Security Considerations TBD

12. Normative References

[I-D.farinacci-lisp-te]

Farinacci, D., Lahiri, P., and M. Kowal, "LISP Traffic Engineering Use-Cases", draft-farinacci-lisp-te-04 (work in progress), January 2014.

[I-D.ietf-lisp-lcaf]

Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format (LCAF)", draft-ietf-lisp-lcaf-04 (work in progress), January 2014.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, January 2013.

[RFC6833] Fuller, V. and D. Farinacci, "Locator/ID Separation Protocol (LISP) Map-Server Interface", RFC 6833, January 2013.

Authors' Addresses

Alberto Rodriguez-Natal
Technical University of Catalonia
Barcelona
Spain

Email: arnatal@ac.upc.edu

Albert Cabellos-Aparicio
Technical University of Catalonia
Barcelona
Spain

Email: acabello@ac.upc.edu

Sharon Barkai
ConteXtream, Inc.
Mountain View, CA
USA

Email: sbarkai@gmail.com

Vina Ermagan
Cisco Systems
170 Tasman Drive
San Jose, CA
USA

Email: vermagan@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

Dino Farinacci
lispers.net
San Jose, CA
USA

Email: farinacci@gmail.com