LISP Working Group
Internet-Draft
Intended status: Experimental
Expires: 9 April 2026

D. Farinacci
lispers.net
M. Kowal
Cisco Systems
P. Lahiri
eBay
P. Pillay-Esnault, Ed.
Independent

6 October 2025

LISP Traffic Engineering
draft-ietf-lisp-te-22

Abstract

This document describes how Locator/Identifier Separation Protocol (LISP) re-encapsulating tunnels can be used for Traffic Engineering purposes. The mechanisms described in this document require no LISP protocol changes but do introduce a new Routing Locator encoding. The Traffic Engineering features provided by these LISP mechanisms can span intra-domain, inter-domain, or a combination of both.

Status of This Memo

Copyright Notice

Table of Contents

1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Introduction

This document describes extensions to the Locator/Identifier Separation Protocol (LISP) [RFC9300] and [RFC9301] for Traffic Engineering (TE) features. For clarity, this document adopts the definition of Traffic Engineering provided in [RFC9522]. Specifically, TE in the Internet context is defined as comprising three main components: policy, path steering, and resource management. This document primarily focuses on the path steering aspect of TE, by specifying how Explicit Locator Paths (ELPs) can be used to guide traffic through specific intermediate Tunnel Routers (xTRs) in a LISP network. Elements of policy may be implicitly supported where operator intent is reflected in ELP selection, and resource management is assumed to be handled by external control systems or network management tools. A detailed discussion of those aspects is out of scope for this document.

When LISP routers encapsulate packets to other LISP routers, the path stretch is typically 1, meaning the packet travels on the shortest path from the encapsulating Ingress Tunnel Router (ITR) to the decapsulating Egress Tunnel Router (ETR) at the destination site. The direct path is determined by the underlying routing protocol and metrics it uses to find the shortest path.

This specification will examine how re-encapsulating tunnels [RFC9300] can be used so a packet can take an administratively specified path, a congestion avoidance path, a failure recovery path, or multiple load-shared paths, as it travels from ITR to ETR. By introducing an Explicit Locator Path (ELP) locator encoding see [RFC8060] section 4.6, an ITR can encapsulate a packet to a Re-Encapsulating Tunnel Router (RTR) which decapsulates the packet, then encapsulates it to the next locator in the ELP.

This document is part of a development effort to include Traffic engineering in LISP. It is not part of an experiment, as not all experimental RFCs are necessarily part of an experiment. It is rather about the maturity level of the technology. This makes it clear that the designation reflects maturity rather than a bounded experiment, and that the document does not define explicit success/ failure criteria.

3. Definition of Terms

Refer to [RFC9300] for authoritative definitions for terms Endpoint Identifier (EID), Routing Locator(RLOC), Ingress Tunnel Router (ITR), Egress Tunnel Router (ETR), Re-encapsulating Tunnel Router (RTR), Tunneling Routers (xTR), Proxy-Egress Tunnel Router (PETR), Proxy Ingress Tunnel Router (PITR), and Recursive Tunneling. The other terms defined in this section add to the canonical definition to reflect the design considerations in this specification. Note: In this document, 'xTR' is used inclusively to refer to ITRs, ETRs, and RTRs, as required by context.

Explicit Locator Path (ELP): The ELP is an explicit list of RLOCs for each RTR a packet SHOULD travel along its path toward a final destination ETR (or PETR). The list MAY be a strict ordering where each RLOC in the list is visited. However, the path from one RTR to another is determined by the underlying routing protocol and how the infrastructure assigns metrics and policies for the path. The definition of an ELP is found in section 3 of [RFC8060].

Re-Encapsulating Tunnel Router (RTR): An RTR as defined in [RFC9300] acts as an ITR (or PITR) by making a decision where to encapsulate the packet based on the next locator in the ELP towards the final destination ETR.

4. Overview

Typically, a packet's path from Source EID (seid) to Destination EID (deid) travels through the locator core via the encapsulating ITR directly to the decapsulating ETR as the following diagram illustrates:

Legend:

seid: Packet is originated by source EID 'seid'.

deid: Packet is consumed by destination EID 'deid'.

A, B, C, D : Core routers in different ASes.

---> : The physical underlay topology supported by routing protocols.

===> : A multi-hop underlay path to realize a LISP tunnel between LISP routers.

In Figure 1 below, the encapsulation tunnel path between ITR and ETR is realized by underlay routers (A, B, C, D) so packets can be delivered which are sent by EID seid to destination EID deid.

```
                              Core Network
   Source site       (----------------------------)    Destination Site
   +--------+        (                            )         +---------+
   |         \       (                            )        /          |
   | seid     ITR ---(---> A --> B --> C --> D ---)---> ETR      deid |
   |         / ||    (                            )     ^^ \          |
   +--------+  ||    (                            )     ||  +---------+
               ||    (----------------------------)     ||
               ||                                       ||
               ============================================
                                LISP Tunnel
```

Figure 1: Typical Data Path from ITR to ETR

In Figure 2, we introduce the RTRs 'X' and 'Y' which creates the opportunity for a tunnel encapsulation path between LISP routers X and Y. For packets encapsulated by ITR to ETR, it may be desirable to route around the link B-->C. One could provide an ELP of (X, Y , etr) to achieve this:

```
                              Core Network
   Source site       (----------------------------)    Destination Site
   +--------+        (                            )         +---------+
   |         \       (                            )        /          |
   | seid     ITR ---(---> A --> B --> C --> D ---)---> ETR      deid |
   |         / ||    (          /      ^          )     ^^ \          |
   |        /  ||    (         |        \         )     ||  \         |
   +-------+   ||    (         v         |        )     ||   +--------+
               ||    (         X ======> Y        )     ||
               ||    (        ^^         ||       )     ||
               ||    (--------||---------||-------)     ||
               ||             ||         ||             ||
               =================         =================
                 LISP Tunnel                 LISP Tunnel
```

Figure 2: ELP tunnel path ITR ==> X, then X ==> Y, and then Y ==> ETR

In this case, the LISP router ITR encapsulates to X, and then X re-encapsulates to Y, and then finally Y re-encapsulates to ETR.

There are various reasons why the path from 'seid' to 'deid' may want to avoid the path from B to C. To list a few:

* There may not be sufficient capacity provided by the networks that connect B and C together.

* There may be a policy reason to avoid the ASes that make up the path between B and C.

* There may be a failure on the path between B and C which makes the path unreliable.

* There may be monitoring or traffic inspection resources close to RTRs X and Y that do network accounting or measurement.

* There may be a chain of services performed at RTRs X and Y regardless if the path from ITR to ETR is through B and C.

5. Explicit Locator Paths

The notation for a general formatted ELP is (x, y, etr), see [RFC8060] (section 4.6 for packet format details, S-bit and L-bit definition), provides the list of RTRs a packet can travel through to reach the final tunnel hop to the ETR.

The procedure for using an ELP at each tunnel hop is as follows:

1. The ITR will retrieve the ELP from the mapping database. If no ELP is returned from the mapping system, follow typical procedures from [RFC9301]. When an ELP is returned, an ELP validity check MUST be performed as detailed in Section 5.4.

2. The ITR will encapsulate the packet to RLOC 'x'. If the S-bit is not set in the ELP, then the ITR MAY encapsulate to subsequent xTRs in the ELP list. Otherwise, when the S-bit is set and an xTR determines the RLOC is not reachable, it MUST NOT use any of the remaining entries in the ELP list and drop the packet. If the L-bit is set, then the ITR does a mapping system lookup on EID 'x' to obtain an RLOC, call it x'. Subsequent behaviors are based on RLOC x'.

3. The RTR with RLOC 'x' will decapsulate the packet. It will use the decapsulated packet's destination address as a lookup into the mapping database to retrieve the ELP.

4. RTR 'x' will encapsulate the packet to RTR with RLOC 'y'.

5. The RTR with RLOC 'y' will decapsulate the packet. It will use the decapsulated packet's destination address as a lookup into the mapping database to retrieve the ELP.

6. RTR 'y' will encapsulate the packet on the final tunnel hop to ETR with RLOC 'etr'.

7. The ETR will decapsulate the packet and deliver the packet to the EID inside of its site.

The specific encoding format for the ELP can be found in [RFC8060]. It is defined that an ELP will appear as a single encoded locator in a locator-set. Say for instance, we have a mapping entry for EID-prefix 192.0.2.0/24 (or if IPv6 2001:db8:200::/48) that is reachable via 4 locators. Two locators are being used as active/active and the other two are used as active/active if the first two go unreachable (as noted by the priority assignments below). This is what the mapping entry would look like:

```
EID-prefix:   192.0.2.0/24
Locator-set:  ETR-A: priority 1, weight 50
              ETR-B: priority 1, weight 50
              ETR-C: priority 2, weight 50
              ETR-D: priority 2, weight 50

EID-prefix:   2001:db8:200::/48
Locator-set:  ETR-A: priority 1, weight 50
              ETR-B: priority 1, weight 50
              ETR-C: priority 2, weight 50
              ETR-D: priority 2, weight 50
```

If an ELP is going to be used to have a policy path to ETR-A and possibly another policy path to ETR-B, the locator-set would be encoded as follows (for each example ELP entry within an RLOC-record below, S-bit=1, L-bit=0, P-bit=0):

```
EID-prefix:   192.0.2.0/24
Locator-set:  (x, y, ETR-A): priority 1, weight 50
              (q, r, ETR-B): priority 1, weight 50
              ETR-C:         priority 2, weight 50
              ETR-D:         priority 2, weight 50

EID-prefix:   2001:db8:200::/48
Locator-set:  (x, y, ETR-A): priority 1, weight 50
              (q, r, ETR-B): priority 1, weight 50
              ETR-C:         priority 2, weight 50
              ETR-D:         priority 2, weight 50
```

The mapping entry with ELP locators is registered to the mapping database system, see [RFC9301] for details, just like any other mapping entry would. The registration is typically performed by the ETR(s) that are assigned and own the EID-prefix. That is, the destination site makes the choice of the RTRs in the ELP. Alternatively, it may be common practice for a third-party system (not an ETR network entity) to register ELP mappings. This can be done via a general purpose Software Defined Network (SDN) provisioning system, for example.

Another case where a locator-set can be used for flow-based load-sharing across multiple paths to the same destination site:

```
EID-prefix:   192.0.2.0/24
Locator-set:  (x, y, ETR-A): priority 1, weight 75
              (q, r, ETR-A): priority 1, weight 25

EID-prefix:   2001:db8:200::/48
Locator-set:  (x, y, ETR-A): priority 1, weight 75
              (q, r, ETR-A): priority 1, weight 25
```

Using this mapping entry, an ITR would load split 75% of the EID flows on the (x, y, ETR-A) ELP path and 25% of the EID flows on the (q, r, ETR-A) ELP path. If any of the ELPs go down, then the other can take 100% of the load. For mapping system lookups and map-cache management, see [RFC9300] for details.

5.1. ELP Re-optimization

ELP re-optimization is a process of changing the RLOCs of an ELP due to underlying network change conditions. Just like when there is any locator change for a locator-set, the procedures from the main LISP specification [RFC9300] are followed.

When a RLOC from an ELP is changed, Map-Notify messages [RFC9301] can be used to inform the existing RTRs in the ELP so they can do a lookup to obtain the latest version of the ELP. Map-Notify messages can also be sent to new RTRs in an ELP so they can get the ELP in advance to receiving packets that will use the ELP. This can minimize packet loss during mapping database lookups in RTRs.

5.2. Using Recursion

In the previous examples, we showed how an ITR encapsulates using an ELP of (x, y, etr). When a packet is encapsulated by the ITR to RTR 'x', the RTR may want a policy path to RTR 'y' and run another level of re-encapsulating tunnels for packets destined to RTR 'y'. In this case, the L-bit is set to 1, RTR 'x' does not encapsulate packets to 'y' but rather performs a mapping database lookup on the address 'y', which returns an ELP-based locator record for a path to RTR 'y', and encapsulates packets to the first-hop of the returned ELP. If the ELP path to RTR 'y' is an internal path within a LISP site, the lookup for RTR 'y' can be done via a private mapping system. The decision to use address 'y' as an encapsulation address versus a lookup address is based on the L-bit setting for 'y' in the ELP entry. The decision and policy of ELP encodings are local to the entity which registers the EID-prefix associated with the ELP.

Another example of recursion is when the ITR uses the ELP (x, y, etr) to first prepend a header with a destination RLOC of the ETR and then prepend another header and encapsulate the packet to RTR 'x'. When

RTR ’x’ decapsulates the packet, rather than doing a mapping database lookup on RTR ’y’ as the last example showed, RTR ’x’ instead does a mapping database lookup on ETR ’etr’. In this scenario, RTR ’x’ can choose an ELP from the locator-set by considering the source RLOC address of the ITR versus considering the source EID.

This additional level of recursion also brings advantages for the provider of RTR ’x’ to store less state. Since RTR ’x’ does not need to look at the inner most header, it does not need to store EID state. It only stores an entry for RTR ’y’ which many EID flows could share for scaling benefits. The locator-set for entry ’y’ could either be a list of typical locators, a list of ELPs, or a combination of both. Another advantage is that packet load-splitting can be accomplished by examining the source of a packet. If the source is an ITR versus the source being the last-hop of an ELP the last-hop selected, different forwarding paths can be used.

5.3. ELP Selection based on Class of Service

Paths to an ETR could be selected based on different classes of service. Packets from a set of sources that have premium service can use ELP paths that are less congested whereas normal sources use ELP paths that compete for less resources or use longer paths for best effort service.

Using source/destination lookups into the mapping database can yield different ELPs. For example, a premium service flow with (source=198.51.100.1, dest=192.0.2.1) or for IPv6 (source=2001:db8:100::1, dest=2001:db8:200::1) can be described by using the following mapping entry:

```
EID-prefix:   (198.51.100.0/24, 192.0.2.0/24)
Locator-set:  (x, y, ETR-A): priority 1, weight 50
              (q, r, ETR-A): priority 1, weight 50

EID-prefix:   (2001:db8:100::/48, 2001:db8:200::/48)
Locator-set:  (x, y, ETR-A): priority 1, weight 50
              (q, r, ETR-A): priority 1, weight 50
```

And all other best-effort sources would use different mapping entry described by:

```
EID-prefix:   (0.0.0.0/0,  192.0.2.0/24)
Locator-set:  (x, x’, y, y’, ETR-A): priority 1, weight 50
              (q, q’, r, r’, ETR-A): priority 1, weight 50

EID-prefix:   (::/0, 2001:db8:200::/48)
Locator-set:  (x, x’, y, y’, ETR-A): priority 1, weight 50
              (q, q’, r, r’, ETR-A): priority 1, weight 50
```

If the source/destination lookup is coupled with recursive lookups, then an ITR can encapsulate to the ETR, prepending a header that selects source address ITR-1 based on the premium class of service source, or selects source address ITR-2 for best-effort sources with normal class of service. The ITR then does another lookup in the mapping database on the prepended header using lookup key (source=ITR-1, dest= 192.0.2.1) or for IPv6 (source=ITR-1, dest=2001:db8:200::1), that returns the following mapping entry:

```
EID-prefix:   (ITR-1,  192.0.2.0/24)
Locator-set:  (x, y, ETR-A): priority 1, weight 50
              (q, r, ETR-A): priority 1, weight 50

EID-prefix:   (ITR-1,  2001:db8:200::/48)
Locator-set:  (x, y, ETR-A): priority 1, weight 50
              (q, r, ETR-A): priority 1, weight 50
```

And all other sources would use different mapping entry with a lookup key of (source=ITR-2, dest= 192.0.2.1) or for IPv6 (source=ITR-2, dest=2001:db8:200::1):

```
EID-prefix:   (ITR-2,  192.0.2.0/24)
Locator-set:  (x, x’, y, y’, ETR-A): priority 1, weight 50
              (q, q’, r, r’, ETR-A): priority 1, weight 50

EID-prefix:   (ITR-2,  2001:db8:200::/48)
Locator-set:  (x, x’, y, y’, ETR-A): priority 1, weight 50
              (q, q’, r, r’, ETR-A): priority 1, weight 50
```

This will scale the mapping system better by having fewer source/destination combinations. Refer to the Source/Dest LCAF type described in [RFC8060] for encoding EIDs in Map-Request and Map-Register messages.

5.4. Packet Loop Avoidance

An ELP that is first used by an ITR MUST be inspected for encoding loops. If any RLOC appears twice in the ELP, it MUST NOT be used.

Since it is expected that multiple mapping systems will be used, there can be a loop across ELPs when registered in different mapping systems. The TTL copying procedures for re-encapsulating tunnels and recursive tunnels in [RFC9300] MUST be followed.

TTL-based loop mitigation is used as a pragmatic safeguard, not a formal loop prevention mechanism. A possible proper encoding loop checks (e.g., ELP inspection for possible loops) will be implemented in future standards track specifications.

6. RLOC Probing by RTRs

Since an RTR knows the next tunnel hop to encapsulate to, it can monitor the reachability of the next-hop RTR. As long as the next-hop RTR sets the P-bit in the ELP list entry, the RTR can use RLOC-probing according to the procedures in [RFC9301]. When the RLOC is determined unreachable by the RLOC-probing mechanisms, the RTR can use another locator in the locator-set. That could be the final ETR, a RLOC of another RTR, or an ELP where it MUST search for itself and use the next RLOC in the ELP list to encapsulate to.

RLOC-probing can also be used to measure delay on the path between RTRs and when it is desirable switch to another lower delay ELP.

7. ELP Probing

Since an ELP-node knows the reachability of the next ELP-node in a ELP by using RLOC probing, the sum of reachability can determine the reachability of the entire path. A head-end ITR/RTR/PITR can determine the quality of a path and decide to select one path from another based on the telemetry data gathered by RLOC-probing for each encapsulation hop.

ELP-Probing uses the RLOC-Probing mechanism defined in [RFC9301], but is executed between each pair of adjacent RLOCs along the Explicit Locator Path (ELP), rather than solely from the ITR to the final hop. However, for telemetry and network management reasons, the ITR could also RLOC-probe the ETR directly to see how a non TE path (the underlay path) compares.

8. Service Chaining

An ELP can be used to deploy services at each re-encapsulation point in the network. One example is to implement a packet scrubber service when a destination EID is being DoS attacked. That is, when a DoS attack is recognized when the encapsulation path is between ITR and ETR, an ELP can be registered for a destination EID in the mapping database system. The ELP can include an RTR so the ITR can encapsulate packets to the RTR, the latter will decapsulate and deliver packets to a scrubber service device. The scrubber could decide if the offending packets are dropped or allowed to be sent to the destination EID. In which case, the scrubber delivers packets back to the RTR, which encapsulates them to the ETR.

9. Interworking Considerations

[RFC6832] defines procedures for how non-LISP sites talk to LISP sites. The network elements defined in the Interworking specification, the Proxy-ITR (PITR) and Proxy-ETR (PETR) (as well as their multicast counterparts defined in [RFC6831]) can participate in LISP-TE. That is, a PITR and a PETR can appear in an ELP list and act as an RTR.

Note when an RLOC appears in an ELP, it can be of any address family. There can be a mix of IPv4 and IPv6 locators present in the same ELP. This can provide benefits where islands of one address-family or the other are supported and connectivity across them is necessary. For instance, an ELP can look like:

(x4, a46, b64, y4, etr)

Where an IPv4 ITR will encapsulate using an IPv4 RLOC 'x4' and 'x4' could reach an IPv4 RLOC 'a46', but RTR 'a46' encapsulates to an IPv6 RLOC 'b64' when the network between them is IPv6-only. Then RTR 'b64' encapsulates to IPv4 RLOC 'y4' if the network between them is dual-stack.

Note that RTRs can be used for NAT-traversal scenarios [I-D.ermagan-lisp-nat-traversal] as well to reduce the state in both an xTR that resides behind a NAT and the state the NAT needs to maintain. In this case, the xTR only needs a default map-cache entry pointing to the RTR for outbound traffic and all remote ITRs can reach EIDs through the xTR behind a NAT via a single RTR (or a small set RTRs for redundancy).

RTRs have some scaling features to reduce the number of locator-set changes, the amount of state, and control packet overhead:

* When ITRs and PITRs are using a small set of RTRs for encapsulating to "orders of magnitude" more EID-Prefixes, the probability of locator-set changes is limited to the RTR RLOC changes versus the RLOC changes for the ETRs associated with the EID-prefixes if the ITRs and PITRs were directly encapsulating to the ETRs. This comes at an expense in packet expansion, but depending on RTR placement, this expense can be mitigated.

* When RTRs are on-path between many pairwise EID flows, ITRs and PITRs can store a small number of coarse EID-prefixes.

* RTRs can be used to help scale RLOC-Probing. Instead of ITRs RLOC-Probing all ETRs for each destination site it has cached, the ITRs can probe a smaller set of RTRs which in turn, probe the destination sites.

10. Multicast Considerations

ELPs have application in multicast environments. Just like RTRs can be used to provide connectivity across different address family islands, RTRs can help concatenate a multicast region of the network to one that does not support native multicast.

Note that there are various combinations of connectivity that can be accomplished with the deployment of RTRs and ELPs:

* Providing multicast forwarding between IPv4-only-unicast regions and IPv4-multicast regions.

* Providing multicast forwarding between IPv6-only-unicast regions and IPv6-multicast regions.

* Providing multicast forwarding between IPv4-only-unicast regions and IPv6-multicast regions.

* Providing multicast forwarding between IPv6-only-unicast regions and IPv4-multicast regions.

* Providing multicast forwarding between IPv4-multicast regions and IPv6-multicast regions.

An ITR or PITR can do a (S-EID, G) lookup into the mapping database. What can be returned is a typical locator-set that could be made up of the various RLOC addresses:

```
Multicast EID key:  (S-EID, G)
Locator-set:        ETR-A: priority 1, weight 25
                    ETR-B: priority 1, weight 25
                    g1:    priority 1, weight 25
                    g2:    priority 1, weight 25
```

Figure 3: An entry for host 'S-EID' sending to application group 'G'

The locator-set above can be used as a replication list. That is, some RLOCs listed can be unicast RLOCs and some can be delivery group RLOCs. A unicast RLOC in this case is used to encapsulate a multicast packet originated by a multicast source EID into a unicast packet for unicast delivery on the underlying network. ETR-A could be an IPv4 unicast RLOC address and ETR-B could be a IPv6 unicast RLOC address.

A delivery group address is used when a multicast packet originated by a multicast source EID is encapsulated in a multicast packet for multicast delivery on the underlying network. Group address 'g1' could be an IPv4 delivery group RLOC and group address 'g2' could be an IPv6 delivery group RLOC.

Flexibility for these various types of connectivity combinations can be achieved and provided by the mapping database system. And the RTR placement allows the connectivity to occur where the differences in network functionality is located.

Extending this concept by allowing ELPs in locator-sets, one could have this locator-set registered in the mapping database for (S-EID, G). For example:

```
Multicast EID key:  (S-EID, G)
Locator-set:        (x, y, ETR-A):    priority 1, weight 50
                    (a, g, b, ETR-B): priority 1, weight 50
```

Figure 4: Using ELPs for multicast flows

In the above situation, an ITR would encapsulate a multicast packet originated by a multicast source EID to the RTR with unicast RLOC 'x'. Then RTR 'x' would decapsulate and unicast encapsulate to RTR 'y' ('x' or 'y' could be either IPv4 or IPv6 unicast RLOCs), which would decapsulate and unicast encapsulate to the final RLOC 'ETR-A'. The ETR 'ETR-A' would decapsulate and deliver the multicast packet natively to all the receivers joined to application group 'G' inside the LISP site.

Let's look at the ITR using the ELP (a, g, b, ETR-B). Here the encapsulation path would be the ITR unicast encapsulates to unicast RLOC 'a'. RTR 'a' multicast encapsulates to delivery group 'g'. The packet gets to all ETRs that have joined delivery group 'g' so they can deliver the multicast packet to joined receivers of application group 'G' in their sites. RTR 'b' is also joined to delivery group 'g'. Since it is in the ELP, it will be the only RTR that unicast encapsulates the multicast packet to ETR 'ETR-B'. Lastly, 'ETR-B' decapsulates and delivers the multicast packet to joined receivers to application group 'G' in its LISP site.

As one can see there are all sorts of opportunities to provide multicast connectivity across a network with non-congruent support for multicast and different address-families. One can also see how using the mapping database can allow flexible forms of delivery policy, rerouting, and congestion control management in multicast environments.

11. Security Considerations

When an RTR receives a LISP encapsulated packet, it can look at the outer source address to verify that RLOC is the one listed as the previous hop in the ELP list. If the outer source RLOC address appears before the RLOC which matches the outer destination RLOC address, the decapsulating RTR (or ETR if last hop), MUST choose to drop the packet as it indicates there is a loop. Loop detection is considered a configuration issue, not a security vulnerability in the context of this draft.

12. IANA Considerations

This document does not make any request to IANA.

13. References

13.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC9300] Farinacci, D., Fuller, V., Meyer, D., Lewis, D., and A. Cabellos, Ed., "The Locator/ID Separation Protocol (LISP)", RFC 9300, DOI 10.17487/RFC9300, October 2022, <https://www.rfc-editor.org/info/rfc9300>.

[RFC9301] Farinacci, D., Maino, F., Fuller, V., and A. Cabellos, Ed., "Locator/ID Separation Protocol (LISP) Control Plane", RFC 9301, DOI 10.17487/RFC9301, October 2022, <https://www.rfc-editor.org/info/rfc9301>.

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

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

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

[RFC9522] Farrel, A., Ed., "Overview and Principles of Internet Traffic Engineering", RFC 9522, DOI 10.17487/RFC9522, January 2024, <https://www.rfc-editor.org/info/rfc9522>.

13.2. Informative References

[I-D.ermagan-lisp-nat-traversal]
Saucez, D., Iannone, L., Ermagan, V., Farinacci, D., Lewis, D., Maino, F., Portoles-Comeras, M., Skriver, J., White, C., and A. L. Bresc坦, "NAT traversal for LISP", Work in Progress, Internet-Draft, draft-ermagan-lisp-nat-traversal-20, 12 March 2025, <https://datatracker.ietf.org/doc/html/draft-ermagan-lisp-nat-traversal-20>.

Appendix A. Acknowledgments

The authors would like to thank the following people for their ideas and comments. They are Albert Cabellos, Khalid Raza, and Vina Ermagan, Gregg Schudel, Yan Filyurin, Robert Raszuk, and Truman Boyes.

Appendix B. Document Change Log

B.1. Changes to draft-ietf-lisp-te-22

* Posted Oct 2025.

* Addressed IANA review comments from Last Call by David Dong

* Addressed Gen-ART review comments from Last Call by Peter Yee

* Addressed Introduction clarity for Traffic engineering comments by Adrian Farrel

* Addressed comments by Eric Vyncke and added a reference to the base spec for ELP probing.

* Addressed comments by Gorry Fairhurst

B.2. Changes to draft-ietf-lisp-te-21

* Posted May 2025.

* Padma as Editor. Fixed IDnits findings and addressed AD comments

B.3. Changes to draft-ietf-lisp-te-20

* Posted November 2024.

* Fix IDnits findings.

B.4. Changes to draft-ietf-lisp-te-19

* Posted June 2024.

* When describing S-bit processing change "must not" to "MUST NOT".

* Change other occurences of "must" to "MUST".

B.5. Changes to draft-ietf-lisp-te-18

* Posted June 2024.

* Add Padma clarification that an ELP should not be used if an S=1 entry is determined unreachable.

B.6. Changes to draft-ietf-lisp-te-17

* Posted June 2024.

* Made changes to reflect Padma’s comments.

B.7. Changes to draft-ietf-lisp-te-16

* Posted May 2024.

* Made some document clarifications based on Luigi’s comments.

B.8. Changes to draft-ietf-lisp-te-15

* Posted April 2024.

* Made changes to reflect comments from Luigi as we ready document for standards track.

B.9. Changes to draft-ietf-lisp-te-14

* Posted February 2024.

* Update references and document timer.

B.10. Changes to draft-ietf-lisp-te-13

* Posted August 2023.

* Update references (to proposed standard documents) and document timer.

B.11. Changes to draft-ietf-lisp-te-12

* Posted March 2023.

* Update references (to propsed standard documents) and document timer.

B.12. Changes to draft-ietf-lisp-te-11

* Posted September 2022.

* Update document timer and references.

B.13. Changes to draft-ietf-lisp-te-10

* Posted March 2022.

* Update document timer and references.

B.14. Changes to draft-ietf-lisp-te-09

* Posted September 2021.

* Update document timer and references.

B.15. Changes to draft-ietf-lisp-te-08

* Posted March 2021.

* Update document timer and references.

B.16. Changes to draft-ietf-lisp-te-07

* Posted October 2020.

* Update document timer and references.

B.17. Changes to draft-ietf-lisp-te-06

* Posted April 2020.

* Update document timer and references.

B.18. Changes to draft-ietf-lisp-te-05

* Posted October 2019.

* Update document timer and references.

B.19. Changes to draft-ietf-lisp-te-04

* Posted April 2019.

* Update document timer and references.

B.20. Changes to draft-ietf-lisp-te-03

* Posted October 2018.

* Update document timer and references.

B.21. Changes to draft-ietf-lisp-te-02

* Posted April 2018.

* Update document timer and references.

B.22. Changes to draft-ietf-lisp-te-01

* Posted October 2017.

* Added section on ELP-probing that tells an ITR/RTR/PITR the feasibility and reachability of an Explicit Locator Path.

B.23. Changes to draft-ietf-lisp-te-00

* Posted April 2017.

* Changed draft-farinacci-lisp-te-12 to working group document.

B.24. Changes to draft-farinacci-lisp-te-02 through -12

* Many postings from January 2013 through February 2017.

* Update references and document timer.

B.25. Changes to draft-farinacci-lisp-te-01.txt

* Posted July 2012.

* Add the Lookup bit to allow an ELP to be a list of encapsulation and/or mapping database lookup addresses.

* Indicate that ELPs can be used for service chaining.

* Add text to indicate that Map-Notify messages can be sent to new RTRs in a ELP so their map-caches can be pre-populated to avoid mapping database lookup packet loss.

* Fixes to editorial comments from Gregg.

B.26. Changes to draft-farinacci-lisp-te-00.txt

* Initial draft posted March 2012.

Authors' Addresses

Dino Farinacci
lispers.net
San Jose, California
United States of America
Email: farinacci@gmail.com

Michael Kowal
Cisco Systems
111 Wood Avenue South
ISELIN, NJ
United States of America
Email: mikowal@cisco.com

Parantap Lahiri
eBay
United States of America
Email: parantap.lahiri@gmail.com

Padma Pillay-Esnault (editor)
Independent
San Jose, California
United States of America
Email: padma.ietf@gmail.com