Network Working Group D. Farinacci
Internet-Draft lispers.net
Intended status: Experimental P. Pillay-Esnault
Expires: November 10, 2017 Huawei Technologies
May 9, 2017

LISP Predictive RLOCs
draft-farinacci-lisp-predictive-rlocs-02

Abstract

This specification will describe a method to achieve near-zero packet loss when an EID is roaming quickly across RLOCs.

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

Copyright Notice

Table of Contents

1. Introduction 2
2. Definition of Terms 3
3. Overview 3
4. Design Details 5
4.1. RLE Encoding 5
4.2. Packet Delivery Optimizations 6
4.3. Trading Off Replication Cost 7
5. Directional Paths with Intersections 8
6. Multicast Considerations 9
7. Multiple Address-Family Considerations 10
8. Scaling Considerations 10
9. Security Considerations 11
10. IANA Considerations 11
11. References 11
11.1. Normative References 11
11.2. Informative References 12
Appendix A. Acknowledgments 12
Appendix B. Document Change Log 13
B.1. Changes to draft-farinacci-lisp-predictive-rlocs-02.txt . 13
B.2. Changes to draft-farinacci-lisp-predictive-rlocs-01.txt . 13
B.3. Changes to draft-farinacci-lisp-predictive-rlocs-00.txt . 13
Authors' Addresses 13

1. Introduction

The LISP architecture [RFC6830] specifies two namespaces, End-Point IDs (EIDs) and Routing Locators (RLOCs). An EID identifies a node in the network and the RLOC indicates the EID's topological location. When an node roams in the network, its EID remains fixed and unchanged but the RLOCs associated with it change to reflect its new topological attachment point. This specification will focus EIDs and RLOCs residing in separate nodes. An EID is assigned to a host node that roams while the RLOCs are assigned to network nodes that stay stationary and are part of the network topology. For example, a set of devices on an aircraft are assigned EIDs, and base stations on the ground attached to the Internet infrastructure are configured as LISP xTRs where their RLOCs are used for the bindings of the EIDs on the aircraft up in the air.

The scope of this specification will not emphasize general physical roaming as an aircraft would do in the sky but in a direction that is

more predictable such as a train traveling on a track or vehicle that travels along a road.

2. Definition of Terms

Roaming-EID - is a network node that moves from one topological location in the network to another. The network node uses the same EID when it is roaming. That is, the EID address does not change for reasons of mobility. A roaming-EID can also be a roaming EID-prefix where a set of EIDs covered by the prefix are all roaming and fate-sharing the same set of RLOCs at the same time.

Predictive RLOCs - is a set of ordered RLOCs in a list each assigned to LISP xTRs where the next RLOC in the list has high probability it will be the next LISP xTR in a physical path going in a single predictable direction.

Road-Side-Units (RSUs) - is a network node that acts as a router, more specifically as a LISP xTR. The xTR automatically discovers roaming-EIDs that come into network connectivity range and relays packets to and from the roaming-EID. RSUs are typically deployed along a directional path like a train track or road and are in connectivity range of devices that travel along the directional path.

3. Overview

The goal of this specification is to describe a make-before-break EID-mobility mechanism that offers near-zero packet loss. Offering minimal packet loss, not only allows transport layers to operate more efficiently, but because an EID does not change while moving, transport layer session continuity is maintained. To achieve these requirements, a mechanism that reacts to the mobility event is necessary but not sufficient. So the question is not that there isn't a reaction but when it happens. By using some predictive algorithms, we can guess with high probability where the EID will roam to next. We can achieve this to a point where packet data will be at the new location when the EID arrives.

First we should examine both the send and receive directions with respect to the roaming-EID. Refer to Figure 1 for discussion. We show a network node with a fixed EID address assigned to a roaming-EID moving along a train track. And there are LISP xTRs deployed as Road-Side-Units to support the connectivity between the roaming-EID and the infrastructure or to another roaming-EID.


```
          Roaming-EID ---->

   ====//====//====//====//====//====//====//====//===//====//====//====
      //    //    //    //    //    //    //    //   //    //    //
  ====//====//====//====//====//====//====//====//===//====//====//====

       xTR         xTR         xTR         xTR        xTR         xTR
        A           B           C           D          E           F
```

Figure 1: Directional Mobility

For the send direction from roaming-EID to any destination can be accomplish as a local decision. As long as the roaming-EID is in signal range to any xTR along the path, it can use it to forward packets. The LISP xTR, acting as an ITR, can forward packets to destinations in non-LISP sites as well as to stationary and roaming EIDs in LISP sites. This is accomplished by using the LISP overlay via dynamic packet encapsulation. When the roaming-EID sends packets, the LISP xTR must discover the EID and MAY register the EID with a set of RLOCs to the mapping system [I-D.portoles-lisp-eid-mobility]. The discovery process is important because the LISP xTR, acting as an ETR for decapsulating packets that arrive, needs to know what local ports or radios to send packets to the roaming-EID.

Much of the focus of this design is on the packet direction to the roaming-EID. And how remote LISP ITRs find the current location (RLOCs) quickly when the roaming-EID is moving at high speed. This specification solves the fast roaming with the introduction of the Predictive-RLOCs algorithm.

Since a safe assumption is that the roaming-EID is going in one direction and cannot deviate from it allows us to know a priori the next set of RLOCs the roaming-EID will pass by. Referring to Figure 1, if the roaming-EID is in range near xTR-A, then as it moves, it will at some point pass by xTR-B and xTR-C, and so on. As the roaming-EID moves, one could time when the EID is mapped to RLOC A, and when it should change to RLOC B and so on. However, the speed of movement of the roaming-EID won't be constant and the variables involved in consistent timing cannot be relied on. Furthermore, timing the move is not a make-before-break algorithm, meaning the reaction of the binding happens at the time the roaming-EID is discovered by an xTR. One cannot achieve fast hand-offs when message signaling will be required to inform remote ITRs of the new binding.

The Predictive RLOCs algorithm allows a set of RLOCs, in an ordered list, to be provided to remote ITRs so they have the information

available and local for when they need to use it. Therefore, no control-plane message signaling occurs when the roaming-EID is discovered by LISP xTRs.

4. Design Details

Predictive RLOCs accommodates for encapsulated packets to be delivered to Road-Side-Unit LISP xTRs regardless where the roaming-EID is currently positioned.

Referring to Figure 1, the following sequence is performed:

1. The Predictive RLOCs are registered to the mapping system as a LCAF encoded Replication List Entry (RLE) Type [I-D.ietf-lisp-lcaf]. The registration can happen by one or more RSUs or by a third-party. When registered by an RSU, and when no coordination is desired, they each register their own RLOC with merge-semantics so the list can be created and maintained in the LISP Map-Server. When registered by a third-party, the complete list of RLOCs can be included in the RLE.

2. There can be multiple RLEs present each as different RLOC-

records so a remote ITR can select one RLOC-record versus the other based in priority and weight policy [RFC6830].

3. When a remote ITR receives a packet destined for a roaming-EID, it encapsulates and replicates to each RLOC in the RLE thereby delivering the packet to the locations the roaming-EID is about to appear. There are some cases where packets will go to locations where the roaming-EID has already been, but see Section 4.2 for packet delivery optimizations.

4. When the ETR resident RSU receives an encapsulated packet, it decapsulates the packet and then determines if the roaming-EID had been previously discovered. If the EID has not been discovered, the ETR drops the packet. Otherwise, the ETR delivers the decapsulated packet on the port interface the roaming-EID was discovered on.

4.1. RLE Encoding

The LCAF [I-D.ietf-lisp-lcaf] Replication List Entry (RLE) will be used to encode the Predictive RLOCs in an RLOC-record for Map-Registers, Map-Reply, and Map-Notify messages [RFC6830].


```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           AFI = 16387         |     Rsvd1     |     Flags     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type = 13   |     Rsvd2     |             4 + n             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Rsvd3            |     Rsvd4     |  Level Value  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              AFI = x          |           RTR/ETR #1 ...      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Rsvd3            |     Rsvd4     |  Level Value  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              AFI = x          |           RTR/ETR  #n ...     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

When the RLOC-record contains an RLE with RLOC entries all with the same level value, it means the physical order listed is the directional path of the RSUs. This will typically be the result of a third-party doing the registration where it knows ahead of time the RSU deployment.

When each RSU is registering with merge-semantics on their own, the level number is used to place them in an ordered list. Since the registrations come at different times and therefore arrive in different order than the physical RSU path, the level number creates the necessary sequencing. Each RSU needs to know its position in the path relative to other RSUs. For example, in xTR-B, it would register with level 1 since it is after xTR-A (and before xTR-C). So if the registration order was xTR-B with level 1, xTR-C with level 2, and xTR-A with level 0, the RLE list stored in the mapping system would be (xTR-A, xTR-B, xTR-C). It is recommended that level numbers be assigned in increments of 10 so latter insertion is possible.

The use of Geo-Prefixes and Geo-Points can be used to compare the physical presence of each RSU with respect to each other, so they can choose level numbers to sequence themselves. Also if the xTRs register with a Geo-Point in an RLOC-record, then perhaps the Map-Server could sequence the RLE list.

4.2. Packet Delivery Optimizations

Since the remote ITR will replicate to all RLOCs in the RLE, a situation is created where packets go to RLOCs that don't need to. For instance, if the roaming-EID is along side of xTR-B and the RLE is (xTR-A, xTR-B, xTR-C), there is no reason to replicate to xTR-A since the roaming-EID has passed it and the the signal range is weak or lost. However, replicating to xTR-B and xTR-C is important to

deliver packets to where the roaming-EID resides and where it is about to go to.

A simple data-plane option, which converges fairly quickly is to have the remote xTR, acting as an ETR, when packets are sent from the roaming-EID, examine the source RLOC in the outer header of the encapsulated packet. If the source RLOC is xTR-B, the remote xTR can determine that the roaming-EID has moved past xTR-A and no longer needs to encapsulate packets to xTR-A's RLOC.

In addition, the remote ITR can use RLOC-probing to determine if each RLOC in the RLE is reachable. And if not reachable, exclude from the list of RLOCs to replicate to.

This solution also handles the case where xTR-A and xTR-B may overlap in radio signal range, but the signal is weak from the roaming-EID to xTR-A but stronger to xTR-B. In this case, the roaming-EID selects xTR-B to send packets that inform the remote xTR that return packets should not be encapsulated to xTR-A.

There are also situations where the RSUs are in signal range of each other in which case they could report reachability status of each other. The use of the Locator-Status-Bits of the LISP encapsulation header could be used to convey this information to the remote xTR. This would only occur when the roaming-EID was discovered by both xTR-A and xTR-B so it was possible for either xTR to reach the roaming-EID. Either an IGP like routing protocol would be required to allow each xTR to know the other could reach the roaming-EID or a path trace tool (i.e. traceroute) could be originated by one xTR targeted for the roaming-EID but MAC-forwarded through the other xTR. These and other roaming-EID reachability mechanisms are work in progress and for further study.

4.3. Trading Off Replication Cost

If RLE lists are large, packet replication can occur to locations well before the roaming-EID arrives. Making RLE lists small is useful without sacrificing hand-off issues or incurring packet loss to the application. By having overlapping RLEs in separate RLOC-records we a simple mechanism to solve this problem. Here is an example mapping entry to illustrate the point:

```
EID = <roaming-EID>, RLOC-records:
  RLOC = (RLE: xTR-A, xTR-B)
  RLOC = (RLE: xTR-B, xTR-C, xTR-D, xTR-E)
  RLOC = (RLE: xTR-E, xTR-F)
```


When the remote ITR is encapsulating to xTR-B as a decision to use the first RLOC-record, it can decide to move to use the second RLOC-record because xTR-B is the last entry in the first RLOC-record and the first entry in the second RLOC-record. When there are overlapping RLEs, the remote ITR can decide when it is more efficient to switch over. For example, when the roaming-EID is in range of xTR-A, the remote ITR uses the first RLOC-record so the wasted replication cost is to xTR-B only versus a worse cost when using the second RLOC-record. But when the roaming-EID is in range of xTR-B, then replicating to the other xTRs in the second RLOC-record may be crucial if the roaming-EID has increased speed. And when the roaming-EID may be at rest in a parked mode, then the remote ITR encapsulates to only xTR-F using the third RLOC-record since the roaming-EID has moved past xTR-E.

In addition, to eliminate unnecessary replication to xTRs further down a directional path, GEO-prefixes [I-D.farinacci-lisp-geo] can be used so only nearby xTRs that the roaming-EID is about to come in contact with are the only ones to receive encapsulated packets.

Even when replication lists are not large, we can reduce the cost of replication that the entire network bears by moving the replicator away from the the source (i.e. the ITR) and closer to the RSUs (i.e. the ETRs). See the use of RTRs for Replication Engineering techniques in [I-D.ietf-lisp-signal-free-multicast].

5. Directional Paths with Intersections

A roaming-EID could be registered to the mapping system with the following nested RLE mapping:

```
EID = <roaming-EID>, RLOC-records:
  RLOC = (RLE: xTR-A, xTR-B, xTR-C, (RLE: xTR-X, xTR-Y, xTR-Z),
               (RLE: xTR-I, xTR-J, xTR-K), xTR-D, xTR-E)
```

The mapping entry above describes 3 directional paths where the ordered list has encoded one-level of two nested RLEs to denote intersections in a horizontal path. Which is drawn as:

```
                                       | |  xTR-K
                                       | |
                                       | |
                                       | |  xTR-J
                                       | |
                                       | |
                          Roaming      | |  xTR-I
                            EID ---->  | |
---------------------------------------   ------------------------------
---------------------------------------   ------------------------------
    xTR-A        xTR-B        xTR-C    | |          xTR-D      xTR-E
                                       | |
                                       | |  xTR-X
                                       | |
                                       | |
                                       | |  xTR-Y
                                       | |
                                       | |
                                       | |  xTR-Z
```

When the roaming-EID is on the horizontal path, the remote-ITRs typically replicate to the rest the of the xTRs in the ordered list. When a list has nested RLEs, the replication should occur to at least the first RLOC in a nested RLE list. So if the remote-ITR is replicating to xTR-C, xTR-D, and xTR-E, it should also replicate to xTR-X and xTR-I anticipating a possible turn at the intersection. But when the roaming-EID is known to be at xTR-D (a left or right hand turn was not taken), replication should only occur to xTR-D and xTR-E. Once either xTR-I or xTR-X is determined to be where the roaming-EID resides, then the replication occurs on the respective directional path only.

When nested RLEs are used it may be difficult to get merge-semantics to work when each xTR registers itself. So it is suggested a third-party registers nested RLEs. It is left to further study to understand better how to automate this.

6. Multicast Considerations

In this design, the remote ITR is receiving a unicast packet from an EID and replicating and encapsulating to each RLOC in an RLE list. This form of replication is no different than a traditional multicast replication function. So replicating multicast packets in the same fashion is a fallout from this design.

If there are multiple roaming-EIDs joined to the same multicast group but reside at different RSUs, a merge has to be done of any pruned RLEs used for forwarding. So if roaming-EID-1 resides at xTR-A and

roaming-EID-2 resides at xTR-B and the RLE list is (xTR-A, xTR-B, xTR-C), and they are joined to the same multicast group, then replication occurs to all of xTR-A, xTR-B, and xTR-C. Even since roaming-EID-2 is past xTR-A, packets need to be delivered to xTR-A for roaming-EID-1. In addition, packets need to be delivered to xTR-C because roaming-EID-1 and roaming-EID-2 will get to xTR-C (and roaming-EID-1 may get there sooner if it is traveling faster than roaming-EID-2).

When a roaming-EID is a multicast source, procedures from [I-D.ietf-lisp-signal-free-multicast] are used to deliver packets to multicast group members anywhere in the network. The solution requires no signaling to the RSUs. When RSUs receive multicast packets from a roaming-EID, they do a (roaming-EID,G) mapping database lookup to find the replication list of ETRs to encapsulate to.

7. Multiple Address-Family Considerations

Note that roaming-EIDs can be assigned IPv6 EID addresses while the RSU xTRs could be using IPv4 RLOC addresses. Any combination of address-families can be supported as well as for multicast packet forwarding, where (S,G) are IPv6 addresses entries and replication is done with IPv4 RLOCs in the outer header.

8. Scaling Considerations

One can imagine there will be a large number of roaming-EIDs. So there is a strong desire to efficiently store state in the mapping database and the in remote ITRs map-caches. It is likely, that roaming-EIDs may share the same path and move at the same speed (EID devices on a train) and therefore share the same Predictive RLOCs. And since EIDs are not reassigned for mobility purposes or may be temporal , they will not be topologically aggregatable, so they cannot compress into a single EID-prefix mapping entry that share the same RLOC-set.

By using a level of indirection with the mapping system this problem can be solved. The following mapping entries could exist in the mapping database:


```
EID = <eid1>, RLOC-records:
  RLOC = (afi=<dist-name>: "am-train-to-paris")
EID = <eid2>, RLOC-records:
  RLOC = (afi=<dist-name>: "am-train-to-paris")
EID = <eid3>, RLOC-records:
  RLOC = (afi=<dist-name>: "am-train-to-paris")

EID = "am-train-to-paris", RLOC-records:
  RLOC = (afi=lcaf/RLE-type: xTR-A, xTR-B, xTR-C)

EID = "am-train-to-paris-passengers", RLOC-records:
  RLOC = (afi=lcaf/afi-list-type: <eid1>, <eid2>, <eid3>)
```

Each passenger that boards a train has their EID registered to point to the name of the train "am-train-to-paris". And then the train with EID "am-train-to-paris" stores the Predictive RLOC-set. When a remote-ITR wants to encapsulate packets for an EID, it looks up the EID in the mapping database gets the name "am-train-to-paris" returned. Then the remote-ITR does another lookup for the name "am-train-to-paris" to get the RLE list returned.

When new EIDs board the train, the RLE mapping entry does not need to be modified. Only an EID-to-name mapping is registered for the specific new EID. Optionally, another name "am-train-to-paris-passengers" can be registered as an EID to allow mapping to all specific EIDs which are on the train. This can be used for inventory, billing, or security purposes.

This optimization comes at a cost of a 2-stage lookup. However, if both sets of mapping entries are registered to the same Map-Server, a combined RLOC-set could be returned. This idea is for further study.

9. Security Considerations

LISP has procedures for supporting both control-plane security [I-D.ietf-lisp-sec] and data-plane security [I-D.ietf-lisp-crypto].

10. IANA Considerations

At this time there are no requests for IANA.

11. References

11.1. 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, <http://www.rfc-editor.org/info/rfc2119>.

[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>.

11.2. Informative References

[I-D.farinacci-lisp-geo] Farinacci, D., "LISP Geo-Coordinate Use-Cases", draft-farinacci-lisp-geo-03 (work in progress), April 2017.

[I-D.ietf-lisp-crypto] Farinacci, D. and B. Weis, "LISP Data-Plane Confidentiality", draft-ietf-lisp-crypto-10 (work in progress), October 2016.

[I-D.ietf-lisp-lcaf] Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format (LCAF)", draft-ietf-lisp-lcaf-22 (work in progress), November 2016.

[I-D.ietf-lisp-sec]
Maino, F., Ermagan, V., Cabellos-Aparicio, A., and D. Saucez, "LISP-Security (LISP-SEC)", draft-ietf-lisp-sec-12 (work in progress), November 2016.

[I-D.ietf-lisp-signal-free-multicast]
Moreno, V. and D. Farinacci, "Signal-Free LISP Multicast", draft-ietf-lisp-signal-free-multicast-04 (work in progress), May 2017.

[I-D.portoles-lisp-eid-mobility]
Portoles-Comeras, M., Ashtaputre, V., Moreno, V., Maino, F., and D. Farinacci, "LISP L2/L3 EID Mobility Using a Unified Control Plane", draft-portoles-lisp-eid-mobility-02 (work in progress), April 2017.

Appendix A. Acknowledgments

The author would like to thank the LISP WG for their review and acceptance of this draft.

Appendix B. Document Change Log

[RFC Editor: Please delete this section on publication as RFC.]

B.1. Changes to draft-farinacci-lisp-predictive-rlocs-02.txt

- o Posted May 2017 to update document timer.

B.2. Changes to draft-farinacci-lisp-predictive-rlocs-01.txt

- o Posted November 2016 to update document timer.

B.3. Changes to draft-farinacci-lisp-predictive-rlocs-00.txt

- o Initial post April 2016.

Authors' Addresses

Dino Farinacci
lispers.net
San Jose, CA
USA

Email: farinacci@gmail.com

Padma Pillay-Esnault
Huawei Technologies
San Clara, CA
USA

Email: padma@huawei.com