Stream: Independent Submission
RFC: 9962
Category: Informational
Published: May 2026
ISSN: 2070-1721
Authors: D. Farinacci *lispers.net*
C. Cantrell *Nexus*

RFC 9962
A Decentralized Locator/ID Separation Protocol Mapping System (LISP-Decent)

Abstract

This document describes how the Locator/ID Separation Protocol (LISP) Mapping System can be distributed for scale and decentralized for management, while maintaining trust among data plane nodes. This is an Informational RFC and should be used by LISP-Decent implementations to interoperate with each other.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This is a contribution to the RFC Series, independently of any other RFC stream. The RFC Editor has chosen to publish this document at its discretion and makes no statement about its value for implementation or deployment. Documents approved for publication by the RFC Editor are not candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9962.

Copyright Notice

Table of Contents

1. Introduction

The LISP architecture [RFC9300] introduces two new numbering spaces that are intended to provide overlay network functionality: Endpoint Identifiers (EIDs) and Routing Locators (RLOCs). To map from EIDs to a set of RLOCs, a control plane Mapping System is used [RFC6836] [RFC8111]. These Mapping Systems are naturally distributed in their deployment for scalability but are centrally managed by a third-party entity, namely a Mapping System Provider (MSP). The

entities that use the Mapping System, such as data plane LISP Tunnel Routers (xTRs), depend on and trust the MSP. They do not participate in the Mapping System other than to register and retrieve information [RFC9301].

This document introduces a Decentralized Mapping System (DMS) so the xTRs can participate in the Mapping System as well as use it. They can trust each other rather than rely on third-party infrastructure. The xTRs act as Map-Servers to maintain distributed state for scale and reduce the attack surface. The trust relationships between the xTRs are established through authentication and authorization procedures in [RFC9301] and [ECDSA-AUTH].

This design was first introduced to the LISP Working Group (WG) in January of 2018. It was presented in London in March 2018 [IETF101-PRESO] and in Prague in March 2019 [IETF104-PRESO]. This Informational document is not a standard and did not reach community consensus to make it an IETF LISP WG document.

The intended audience for this specification are protocol development engineers who would implement this specification in products as well network engineers who deploy the technology in operational environments.

This design does not conflict with those designs or contradicts any other LISP design principles produced by the working group. This solution makes no fundamental LISP changes and adheres to the specifications documented in [RFC9301].

By the nature of this work, this document uses IP addresses as examples. They should not be copied or used outside of a lab environment. Deployments should use addresses appropriate for their environment.

2. Definitions of Terms

Mapping System Provider (MSP): Infrastructure service that deploys LISP Map-Resolvers and Map-Servers [RFC9301] and possibly LISP-ALT nodes [RFC6836] or LISP-DDT nodes [RFC8111]. ALT-nodes and DDT-nodes use different algorithms for connecting Map-Resolvers and Map-Servers together. The MSP can be managed by a separate organization other than the one that manages xTRs. This model provides a business separation between the organization that manages (and is responsible for) the control plane versus the organization that manages the data plane overlay service.

Decentralized Mapping System (DMS): A mapping system entity that is managed by the xTR nodes that use it and not third-party nodes/operators. The xTRs themselves are part of the Mapping System. The state of the Mapping System is fully distributed across xTRs, decentralized, and the trust relies on the xTRs that use and participate in their own mapping system.

Decent-Pull-Based Mapping System: The mapping system is pull-based, meaning that xTRs will look up and register mappings by algorithmic transformation to locate which Map-Resolvers and Map-Servers are used. It is required that the lookup and registration uses a consistent algorithmic transformation function. Map-Registers are sent to specific Map-Servers. Map-

Requests are considered external lookups when sent to Map-Resolvers on xTRs that do not participate in the mapping system and the Map-Requests are considered internal lookups when they are sent to Map-Resolvers on xTRs that do participate in the mapping system.

Modulus Value: This value is used in the Pull-based Mapping System. It defines the number of Map-Server sets used for the Mapping System. The modulus value is used to produce a Name Index used for a Domain Name System (DNS) lookup. The Modulus Value is chosen based on the horizontal scale-out width of the Map-Server cluster the network operator chooses to deploy.

Name Index: This index value <index> is used in the Pull-based Mapping System. For a Mapping System that is configured with a Map-Server set of DNS names in the form of <name>.example.com, the Name Index is prepended to <name> to form the lookup name <index>.<name>.example.com. If the Modulus Value is 8, then the Name Indexes are 0 through 7.

Hash Mask: The Hash Mask is used in the Pull-based Mapping System. It is a mask value with 1 bit left justified. The mask is used to select what high-order bits of an EID-prefix are used in the hash function.

Decent-Push-Based Mapping System: The Mapping System is push-based, meaning that xTRs will push registrations via IP multicast to a group of Map-Servers and do local lookups acting as their own Map-Resolvers.

Replication List Entry (RLE): An RLOC-record format that contains a list of RLOCs that an Ingress Tunnel Router (ITR) replicates multicast packets on a multicast overlay. The RLE format is specified in [RFC8060]. RLEs are used with the Decent-Push-based Mapping System.

Group Address EID: An EID-record format that contains IPv4 (0.0.0.0/0, G) or IPv6 (0::/0, G) state. This state is encoded as a Multicast Info Type LISP Canonical Address Format (LCAF) specified in [RFC8060]. Members of a seed-group send Map-Registers for (0.0.0.0/0, G) or (0::/0, G) with an RLOC-record that RLE encodes its RLOC address. Details are specified in [RFC8378].

Seed-Group: A set of Map-Servers joined to a multicast group for the Decent-Push-based Mapping System or are mapped by DNS names in a Pull-based Mapping System. A core seed-group is used to bootstrap a set of LISP-Decent xTRs so they can learn about each other and use each other's Mapping System service. A seed-group can be pull-based to bootstrap the Decent-Push-based Mapping System. That is, a set of DNS mapped Map-Servers can be used to join the mapping system's IP multicast group.

2.1. 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.

3. Overview

The clients of the DMS are also the providers of mapping state, see [RFC9301] for formal details. Clients are typically Egress Tunnel Routers (ETRs) that Map-Register EID-to-RLOC mapping state to the mapping database system. ITRs are clients in that they send Map-Requests to the mapping database system to obtain EID-to-RLOC mappings that are cached for data plane use. When xTRs participate in a DMS, they are also acting as Map-Resolvers and Map-Servers using the protocol machinery defined in LISP control plane specifications [RFC9301], [RFC9303], and [ECDSA-AUTH]. The xTRs are not required to run the database mapping transport system protocols specified in [RFC6836] or [RFC8111].

This document will describe two decentralized and distributed mapping system mechanisms. A Decent-Push-based Mapping System uses IP multicast so xTRs can find each other by locally joining an IP multicast group. A Decent-Pull-based Mapping System uses DNS with an algorithmic transformation function so xTRs can find each other.

The document proposes two approaches to provide state/bandwidth, ease of configurability, and robustness/complexity trade-offs. When the Decent-Push-based approach is used, there is less messaging involved. xTRs can multicast a single Map-Register message that goes to all Map-Servers joined to the multicast group. When Map-Servers are added to or removed from the Map-Server cluster group, the Mapping System updates quickly with little human intervention. In the push model, all mapping state is stored in all Map-Servers so there is robustness achieved through redundancy. However, this requires a multicast underlay in nodes between all xTRs and Map-Servers. When a multicast underlay is not available, the Decent-Pull-based approach can be used with the help of the DNS. This approach uses less state overall among the Map-Servers (they each have different Mapping System state) and the ITRs know which Map-Server to ask by using the hashing techniques described later in this document.

This document does not describe any compatibility with other mapping systems. The design is intentional so that all xTRs and Map-Servers support this specification. Moreover, all xTRs and Map-Servers **MUST** support either the pull-based or push-based algorithms. They cannot be mixed. When both are used, they are completely discrete Mapping Systems just like they would be using one of the LISP WG Mapping System designs. An implementation can decide to implement only the pull-based approach or the push-based approach and still be compliant with this specification.

4. Decent-Push-Based Mapping System

The xTRs are organized in a mapping-system group. The group is identified by an IPv4 or IPv6 multicast group address or using a Decent-Pull-based approach described in Section 5. When using multicast, the xTRs join the same multicast group and receive LISP control plane messages addressed to the group. Messages sent to the multicast group are distributed when the underlay network supports IP multicast [RFC6831] or via the overlay multicast mechanism described in [RFC8378]. When overlay multicast is used and LISP Map-Register messages are sent to the group, they are LISP data encapsulated with an instance-ID set to 0xffffff in the LISP header. The

inner header of the encapsulated packet has the destination address set to the multicast group address and the outer header that is prepended has the destination address set to the RLOC of Mapping System group member. The members of the Mapping System group are kept in the LISP data plane map-cache so packets for the group can be replicated to each member RLOC.

All xTRs in a Mapping System group will store the same registered mappings and maintain the state as Map-Servers normally do. The members are not only receivers of the multicast group, but they also send packets to the group.

4.1. Components of a Decent-Push-Based LISP-Decent xTR

When an xTR is configured to be a LISP-Decent xTR (or PxTR [RFC6832]), it runs the ITR, ETR, Map-Resolver, and Map-Server LISP network functions.

The following diagram shows 3 LISP-Decent xTRs joined to Mapping System group address 233.252.1.1. When the ETR function of xTR1 originates a Map-Register, it is sent to all xTRs (including itself) synchronizing all 3 Map-Servers in xTR1, xTR2, and xTR3. The ITR function can populate its map-cache by sending a Map-Request locally to its Map-Resolver so it can replicate packets to each RLOC for EID 233.252.1.1.

In this document, there is no special meaning for the multicast group address 233.252.1.1. It is used for example purposes only.

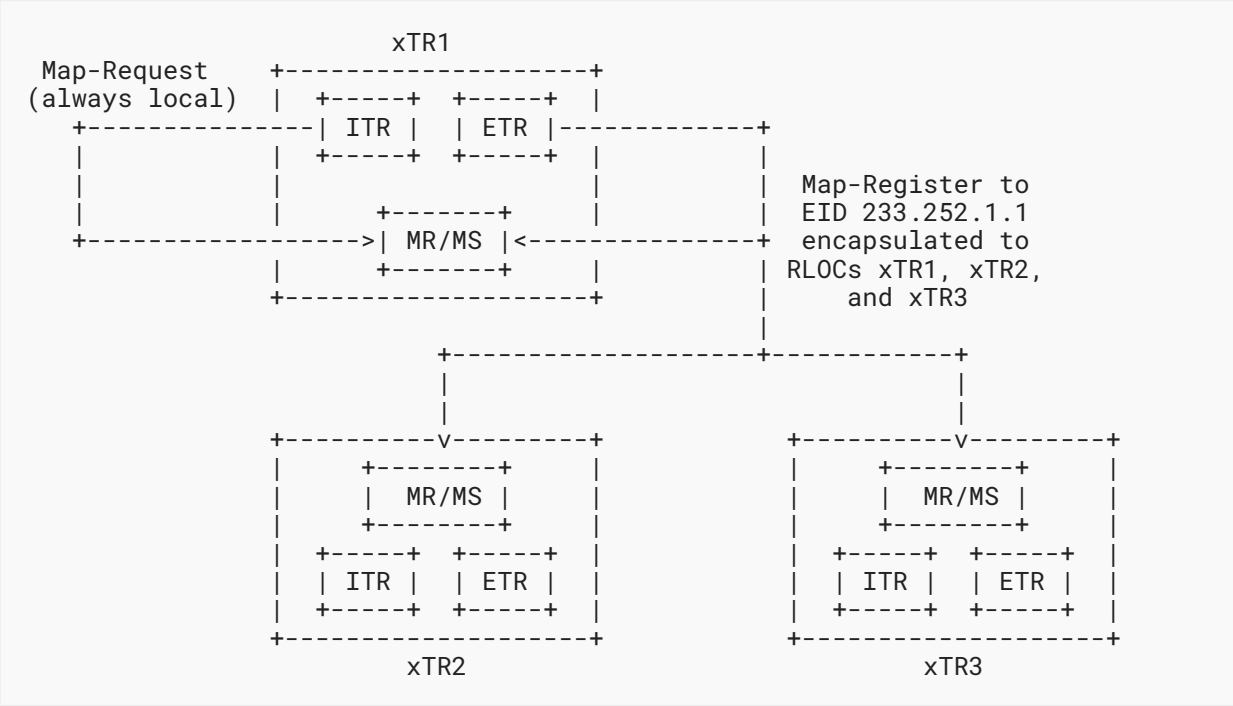

Figure 1

Note that if any external xTR would like to use a Map-Resolver from the Mapping System group, it only needs to have one of the LISP-Decent Map-Resolvers configured. By looking at the Map-Resolver for EID 233.252.1.1, the external xTR could get the complete list of members for the Mapping System group.

For future study, an external xTR could multicast the Map-Request to 233.252.1.1 and either one of the LISP-Decent Map-Resolvers would return a Map-Reply or the external xTR is prepared to receive multiple Map-Replies.

4.2. Implementation Considerations

There are no LISP changes required to support the Decent-Push-based LISP-Decent set of procedures. An implementation that sends Map-Register messages to a multicast group versus a specific Map-Server unicast address **MUST** change to call the data plane component so the ITR functionality in the node can encapsulate the Map-Register as a unicast packet to each member of the Mapping System group.

An ITR **SHOULD** look up its Mapping System group address periodically to determine if the membership has changed. The lookup interval is a configuration parameter only needed when the ITR does not use the PubSub capability documented in [RFC9437] to be notified when a new member joins or leaves the multicast group. When PubSub is not used, there needs to be coordination (via configuration management) among all xTRs so they rendezvous roughly at the same time and to the same group address.

4.3. Configuration and Authentication

When xTRs are joined to a multicast group, they **MUST** have their site registration configuration consistent. Any policy or authentication key material **MUST** be configured consistently among all members. When [ECDSA-AUTH] is used to sign Map-Register messages, public keys can be registered to the Mapping System group using the site authentication key mentioned above or using a different authentication key from the one used for registering EID records. An out-of-band registration controller, like an ETR dedicated for this function, can send Map-Register messages for public-key encoded EIDs.

4.4. Core Seed-Group

A core seed-group can be discovered using a multicast group in a Decent-Push-based system or a Map-Server set of DNS names in a Decent-Pull-based system (see Section 5 for details).

When using multicast for the Mapping System group, a core seed-group multicast group address can be preconfigured to bootstrap the decentralized Mapping System. The group address (or DNS name that maps to a group address) can be explicitly configured in a few xTRs to start building up the registrations. Then as other xTRs come online, they can add themselves to the core seed-group by joining the seed-group multicast group.

Alternatively or additionally, new xTRs can join a new Mapping System multicast group to form another layer of a decentralized Mapping System. The group address and members of this new layer seed-group would be registered to the core seed-group address and stored in the core seed-group mapping system. Note that each Mapping System layer could have a specific function or a specific circle of trust.

This multi-layer Mapping System can be illustrated:

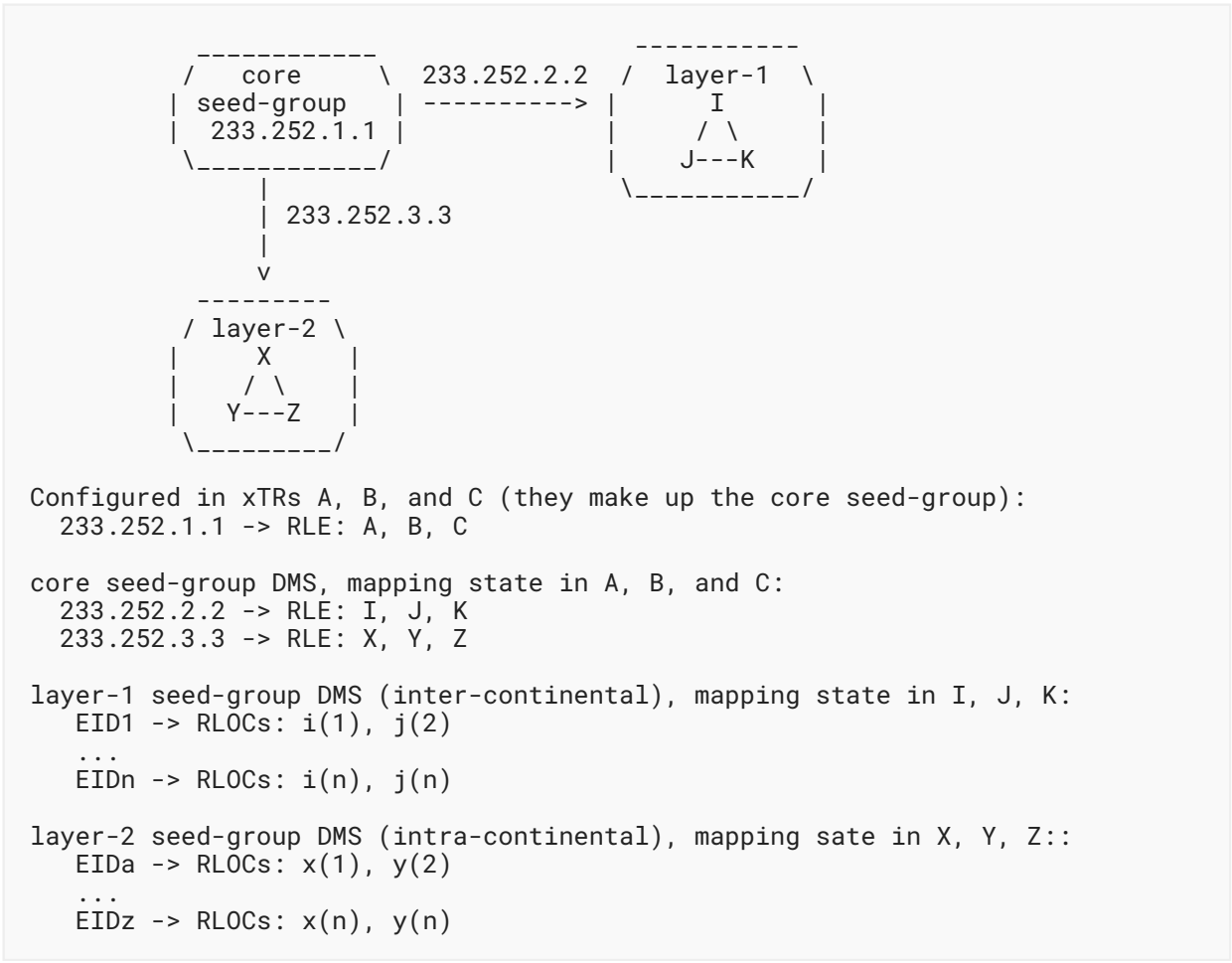

Figure 2

The core seed-group multicast address 233.252.1.1 is configured in xTRs A, B, and C so when each of them send Map-Register messages, they would all be able to maintain synchronized mapping state. Any EID can be registered to this DMS, but in this example, seed-group multicast group EIDs are being registered only to find other Mapping System groups.

For example, xTR I boots up and it wants to find its other peers in its Mapping System group address 233.252.2.2. Group address 233.252.2.2 is configured so xTR I knows what group to join for its Mapping System group. However, xTR I needs a Mapping System to register to, so the core seed-group is used and available to receive Map-Registers. The other xTRs J and K in the

Mapping System group do the same so when any of I, J, or K needs to register EIDs, they can now send their Map-Register messages to group address 233.252.2.2. Examples of EIDs being registered are EID1 through EIDn shown above.

When Map-Registers are sent to group address 233.252.2.2, they are encapsulated by the LISP data plane by looking up EID 233.252.2.2 in the core seed-group Mapping System. For the map-cache entry to be populated for 233.252.2.2, the data plane **MUST** send a Map-Request so the RLOCs I, J, and K are cached for replication. To use the core seed-group Mapping System, the data plane **MUST** know of at least one of the RLOCs A, B, and/or C.

5. Decent-Pull-Based Mapping System

5.1. Components of a Decent-Pulled Based LISP-Decent xTR

When an xTR is configured to be a LISP-Decent xTR (or PxTR [RFC6832]), it runs the ITR, ETR, Map-Resolver, and Map-Server LISP network functions.

Unlike the Decent-Push-based Mapping System, the xTRs do not need to be organized by joining a multicast group. In a Decent-Pull-based Mapping System, a hash function over an EID is used to identify which xTR is used as the Map-Resolver and Map-Server. The DNS [RFC1034] [RFC1035] is used as a resource discovery mechanism.

The RLOC addresses of the xTRs will be A and AAAA records for DNS names that map algorithmically from the hash of the EID. A SHA-256 hash function [RFC6234] over the following ASCII-formatted EID string is used:

```
[<iid>]<eid>/<ml>
[<iid>]<group>/<gml>-<source>/<sml>
```

In this section, where you see angle brackets, they are replaced with values in ASCII characters. For example, a unicast EID of 1.1.1.1 in instance-id 11 would be encoded as a string "[11]1.1.1.1/32".

<iid> is the instance-ID and <eid> is the EID of any EID-type defined in [RFC8060]. The Modulus Value <mv> is used to produce the Name Index <index> used to build the DNS lookup name:

```
eid = "[<iid>]<eid>/<ml>"
index = hash.sha_256(eid) MOD mv
```

The Hash Mask is used to select what bits are used in the SHA-256 hash function. This is required to support longest match lookups in the Mapping System. The same Map-Server set needs to be selected when looking up a more-specific EID found in the Map-Request message with one that could match a less-specific EID-prefix registered and found in the Map-Register message. For

example, if an EID-prefix [0]240.0.1.0/24 is registered to the Mapping System and EID [0]240.0.1.1/32 is looked up to match the registered prefix, a Hash Mask of 8 bytes can be used to AND both the /32 or /24 entries to produce the same hash string bits of "[0]240.0".

For (*,G) and (S,G) multicast entries in the Mapping System, the hash strings are:

```
sg-eid = "[<iid>]<group>/<gml>-<source>/<sml>"
index = hash.sha_256(sg-eid) MOD mv

starg-eid = "[<iid>]<group>/<gml>-0.0.0.0/0"
index = hash.sha_256(starg-eid) MOD mv
```

The Hash Mask **MUST** include the string "[<iid>]<group>" and not string <source>. So when looking up [0](2.2.2.2, 233.252.1.1) that will match a (*, 233.252.1.1/32), the hash string produced with a Hash Mask of 12 bytes is "[0]233.252.1.1".

When the <index> is computed from a unicast or multicast EID, the DNS lookup name becomes:

```
<index>.map-server.example.com
```

When an xTR does a DNS lookup on the lookup name, it will send Map-Register messages to all A and AAAA records for EID registrations. For Map-Request messages, xTRs **MAY** round robin EID lookup requests among the A and AAAA records.

5.2. Configurable EID Prefix Lengths for Lookups

In implementations where EID allocations follow a predictable pattern, operators **MAY** configure ITRs to use specific prefix lengths for lookups when the EID falls within well-known allocation ranges. This configuration allows ITRs to compute the correct hash index even when data packets carry more-specific EIDs than the prefix lengths used by ETRs during registration.

For example, if an operator allocates EIDs in the range [0]240.11.0.0/16 and all ETRs register these EIDs with a /24 prefix length, ITRs receiving data packets for EIDs like [0]240.11.1.1/32 will still hash on the /24 prefix to locate the correct Map-Server. Without this configuration, the ITR would hash on the /32 and potentially query the wrong Map-Server.

ITRs **SHOULD** support configuration of EID prefix ranges and their associated lookup prefix lengths. When an ITR performs a Map-Request lookup, it **SHOULD** check if the destination EID matches any configured range. If a match is found, the ITR **MUST** use the configured lookup prefix length instead of the EID's registered prefix length when computing the hash string. When multiple configured ranges match a given EID, the range with the longest (most-specific) configured prefix length **MUST** be selected.

The configuration consists of pairs of EID-prefix (the well-known EID allocation range in Classless Inter-Domain Routing (CIDR) notation, e.g., 240.11.0.0/16) and lookup-length (the prefix length to use for hash computation when EIDs fall within this range, 0-32 bytes for IPv4 and 0-128 bytes for IPv6).

Implementation note: When computing the hash string for a lookup where the EID matches a configured range, the ITR **MUST** construct the hash string using the configured lookup-length, ensuring that the bits beyond the lookup-length are zero (i.e., the EID address is masked to the lookup-length before converting to the hash string format).

Example configuration with multiple EID ranges:

```
EID Range Configuration:
  Range: [0]240.11.0.0/16  -> lookup-length: 24
  Range: [0]240.12.0.0/16  -> lookup-length: 30
  Range: [0]240.13.0.0/16  -> lookup-length: 25

Lookup Examples:
  EID [0]240.11.1.1/32     -> hash on [0]240.11.1.0/24
  EID [0]240.12.2.5/32     -> hash on [0]240.12.2.4/30
  EID [0]240.13.3.7/32     -> hash on [0]240.13.3.0/25
  EID [0]240.14.1.1/32     -> hash on [0]240.14.1.1/32
                               (no match, use full EID)
```

This approach is particularly useful in deployments where the Mapping System uses a consistent ETR registration strategy (e.g., all ETRs in a region register with /24 prefixes), but ITRs receive packets with more-specific destinations (/32 addresses). By configuring the expected registration prefix length for each allocation range, ITRs can reliably locate the correct Map-Servers without modifying LISP or introducing complex multi-level lookups.

5.3. Deployment Example

Here is an example deployment of a Decent-Pull-based model. Let's say that 4 Map-Server sets are provisioned for the Mapping System. Therefore, 4 distinct DNS names are allocated and a Modulus Value 4 is used. Each DNS name is allocated Name Index 0 through 3:

```
    0.map-server.lispers.net
    1.map-server.lispers.net
    2.map-server.lispers.net
    3.map-server.lispers.net
```

The A records for each name can be assigned as:

```
0.map-server.lispers.net:
    A <rloc1-att>
    A <rloc2-verizon>
1.map-server.lispers.net:
    A <rloc1-bt>
    A <rloc2-dt>
2.map-server.lispers.net:
    A <rloc1-cn>
    A <rloc2-kr>
3.map-server.lispers.net:
    A <rloc1-au>
    A <rloc2-nz>
```

When an xTR wants to register "[1000]fd::2222/128", it hashes the EID string to produce, for example, hash value 0x67. Using the modulus value 4 (0x67 & 0x3) produces index 0x3, so the DNS name 3.map-server.lispers.net is used and a Map-Register is sent to <rloc1-au> and <rloc2-nz>.

Note that the Decent-Pull-based method can be used for a core seed-group for bootstrapping a Decent-Push-based Mapping System where multicast groups are registered.

5.4. Management Considerations

An implementation **SHOULD** do periodic DNS lookups to determine if A records have changed for a DNS entry. This is a configuration parameter the network operator selects. This specification makes no recommendation for an interval value.

When xTRs derive Map-Resolver and Map-Server names from the DNS, they need to use the same Modulus Value. Otherwise, some xTRs will look up EIDs to the wrong place they were registered.

The Modulus Value can be configured or pushed to the LISP-Decent xTRs. A future version of this document will describe a push mechanism so all xTRs use a consistent modulus value.

6. Security Considerations

Refer to Section 9 of [RFC9301] for a complete list of security mechanisms as well as pointers to threat analysis documents.

Where DNS is deployed for the Pull-based Mapping System, it is recommended to use DNSSEC [RFC9364] to add more security to the DNS lookup system.

Replay attacks, spoofing, and trust relationships are discussed in detail in [RFC9301] and [RFC9303].

7. IANA Considerations

This document has no IANA actions.

8. References

8.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[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>.

[RFC6234] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <https://www.rfc-editor.org/info/rfc6234>.

[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>.

[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, <https://www.rfc-editor.org/info/rfc6836>.

[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>.

[RFC8111] Fuller, V., Lewis, D., Ermagan, V., Jain, A., and A. Smirnov, "Locator/ID Separation Protocol Delegated Database Tree (LISP-DDT)", RFC 8111, DOI 10.17487/RFC8111, May 2017, <https://www.rfc-editor.org/info/rfc8111>.

[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>.

[RFC8378] Moreno, V. and D. Farinacci, "Signal-Free Locator/ID Separation Protocol (LISP) Multicast", RFC 8378, DOI 10.17487/RFC8378, May 2018, <https://www.rfc-editor.org/info/rfc8378>.

[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>.

[RFC9303] Maino, F., Ermagan, V., Cabellos, A., and D. Saucez, "Locator/ID Separation Protocol Security (LISP-SEC)", RFC 9303, DOI 10.17487/RFC9303, October 2022, <https://www.rfc-editor.org/info/rfc9303>.

[RFC9364] Hoffman, P., "DNS Security Extensions (DNSSEC)", BCP 237, RFC 9364, DOI 10.17487/RFC9364, February 2023, <https://www.rfc-editor.org/info/rfc9364>.

[RFC9437] Rodriguez-Natal, A., Ermagan, V., Cabellos, A., Barkai, S., and M. Boucadair, "Publish/Subscribe Functionality for the Locator/ID Separation Protocol (LISP)", RFC 9437, DOI 10.17487/RFC9437, August 2023, <https://www.rfc-editor.org/info/rfc9437>.

8.2. Informative References

[ECDSA-AUTH] Farinacci, D. and E. Nordmark, "LISP Control-Plane ECDSA Authentication and Authorization", Work in Progress, Internet-Draft, draft-ietf-lisp-ecdsa-auth-16, 29 January 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-lisp-ecdsa-auth-16>.

[IETF101-PRESO] Farinacci, D. and C. Cantrell, "A Decentralized Mapping System: draft-farinacci-lisp-decent-00", IETF 101 Proceedings, March 2018, <https://datatracker.ietf.org/meeting/101/materials/slides-101-lisp-lisp-decentralized-mapping-system-00>.

[IETF104-PRESO] Farinacci, D. and C. Cantrell, "A Decentralized Mapping System: draft-farinacci-lisp-decent-03", IETF 104 Proceedings, March 2019, <https://datatracker.ietf.org/meeting/104/materials/slides-104-lisp-a-decent-lisp-mapping-system-00>.

Acknowledgments

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

The authors would also like to give a special thanks to Roman Shaposhnik for several discussions that occurred before the initial draft of this document.

Eliot Lear and Victor Moreno are appreciated for their efforts proofreading the draft before publication as an RFC.

Authors' Addresses

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

Colin Cantrell
Nexus
Tempe, AZ
United States of America
Email: colin@nexus.io