Network Working Group
Internet-Draft
Intended status: Experimental
Expires: May 31, 2012

V. Fuller
D. Lewis
D. Farinacci
cisco Systems

November 28, 2011

LISP Delegated Database Tree
draft-fuller-lisp-ddt-00.txt

Abstract

This draft describes the LISP Delegated Database Tree (LISP-DDT), a hierarchical, distributed database which embodies the delegation of authority to provide mappings from LISP Endpoint Identifiers (EIDs) to Routing Locators (RLOCs). It is a statically-defined distribution of the EID namespace among a set of LISP-speaking servers, called DDT nodes. Each DDT node is configured with an EID-prefix that it "owns" plus information, including the RLOCs for Map Servers or other DDT nodes for each defined more-specific EID-prefix of the "owned" prefix.

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 May 31, 2012.

Copyright Notice

Copyright (c) 2011 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

[LISP] specifies an architecture and mechanism for replacing the addresses currently used by IP with two separate name spaces: relatively static Endpoint Identifiers (EIDs), used end-to-end for terminating transport-layer associations, and Routing Locators (RLOCs), which are more dynamic, are bound to topological location, and are used for routing and forwarding through the Internet infrastructure.

LISP offers a general-purpose mechanism for mapping between EIDs and RLOCs. In organizing a database of EID to RLOC mappings, this specification extends the definition of the EID numbering space by logically prepending and appending several fields for purposes of defining the database index key: Key-ID (16 bits), Instance Identifier (IID, 32-bits), Address Family Identifier (16 bits), and EID-prefix (variable, according to AFI value). The resulting concatenation of these fields is termed an "Extended EID prefix" or XEID-prefix.

The term "Extended EID" (XEID) is also used to for an individual LISP EID that is further qualified through the use of an Instance ID. See [LCAF] for further discussion of the use of Instance IDs.

The Key-ID is provided for possible use in case a need evolves for another, higher level in the hierarchy, to allow the creation of multiple, separate database trees.

LISP-DDT is a hierarchical distributed database which embodies the delegation of authority to provide mappings, i.e. its internal structure mirrors the hierarchical delegation of address space. It also provides delegation information to Map-Resolvers, which use the information to locate EID-to-RLOC mappings. A Map-Resolver which needs to locate a given mapping will follow a path through the tree-structured database, contacting, one after another, the DDT nodes along that path until it reaches the leaf DDT node(s) authoritative for the mapping it is seeking.

LISP-DDT defines a new device type, the DDT node, that is configured with one or more XEID-prefix which it "owns" (for which it is termed to be "authoritative") and the set of more-specific sub-prefixes of the prefix(es) that are further delegated to other DDT nodes. To delegate a sub-prefix, the "parent" DDT node is configured with the RLOCs of each "child" DDT node that is authoritative for the sub-prefix. Each RLOC is either for a Map Server (sometimes termed a "terminal DDT node") that is responsible for contacting the Egress Tunnel Routers (ETRs) for that sub-prefix or is for another DDT node in the database tree that provides further sub-prefix delegation.

See [LISP-MS] for a description of the functionality of the Map Server and Map Resolver. Note that the target of a delegation must always be an RLOC (not an EID) to avoid any circular dependency.

To provide a mechanism for traversing the database tree, LISP-DDT defines a new LISP message type, the Map-Referral, which is returned to the sender of a Map-Request when the receiving DDT node can refer the sender to another DDT node that has more detailed information.

A DDT client uses LISP-DDT to find an EID-to-RLOC mapping by first sending a Map-Request to the RLOC of a DDT node. The initial choice of DDT node is configured on the client. If the receiving DDT node is also a Map Server that is responsible for the XEID queried, the Map-Request is handled as described in [LISP-MS], with the DDT Map Server also returning a Map-Referral message with the "done" flag set to the Map-Request sender. Otherwise, the DDT node answers the Map-Request with a Map-Referral; the DDT client then re-sends its DDT Map-Request to one of the RLOCs listed in the Map-Referral. This iterative process of sending requests and following referrals continues until the client receives a Map-Referral with the "done" flag set. This is an indication that the terminal DDT Map-Server has either answered the Map-Request (if offering proxy service) or has forwarded it to the correct ETR which will answer it. Conceptually, this is similar to the way that a client of the Domain Name System (DNS) follows referrals (DNS responses that contain only NS records) from a series of DNS servers until it finds an answer.

2. Definition of Terms

Extended EID (XEID): a LISP EID, optionally extended with a non-zero Instance ID (IID) if the EID is intended for use in a context where it may not be a unique value, such as on a Virtual Private Network where "private" address space is used. See "Using Virtualization and Segmentation with LISP" in [LISP] for more discussion of Instance IDs.

XEID-prefix: a LISP EID-prefix with 16-bit LISP-DDT Key-ID (provided to allow the definition of multiple databases; currently always zero in this version of DDT, with other values reserved for future use), 32-bit IID and 16-bit AFI prepended. An XEID-prefix is used as a key index into the database.

DDT node: a network infrastructure component responsible for specific XEID-prefix and for delegation of more-specific sub-prefixes to other DDT nodes.

DDT client: a network infrastructure component that sends Map-Request messages and implements the iterative following of Map-Referral results. Typically, a DDT client will be a Map Resolver but it is also possible for an ITR to implement DDT client functionality.

DDT Map Server: a DDT node that also implements Map Server functionality (forwarding Map-Requests and/or returning Map-Replies if offering proxy-mode service) for a subset of its delegated prefixes.

DDT Map Resolver: a network infrastructure element that accepts a Map-Request, adds the XEID to its lookup queue, then queries one or more DDT nodes for the requested EID, following returned referrals until it receives one with the "done" flag. This indicates that the Map-Request has been sent to a Map-Server that will forward it to an ETR that, in turn, will provide a Map-Reply to the original sender. A DDT Map Resolver maintains both a cache of Map-Referral message results containing RLOCs for DDT nodes responsible for XEID-prefixes of interest (termed the "referral cache") plus a lookup queue of XEIDs that are being resolved through iterative querying of DDT nodes.

Encapsulated Map-Request: a LISP Map-Request carried within an Encapsulated Control Message, which has an additional LISP header prepended. Sent to UDP destination port 4342. The "outer" addresses are globally-routable IP addresses, also known as RLOCs. Used by an ITR when sending to a Map-Resolver and by a Map-Server when forwarding a Map-Request to an ETR as documented in

[LISP-MS].

DDT Map-Request: an Encapsulated Map-Request sent by a DDT client to a DDT node. The "DDT-originated" flag is set in the encapsulation header indicating that the DDT node should return Map-Referral messages if the Map-Request EID matches a delegated XEID-prefix known to the DDT node. Section 7.1 describes how DDT Map-Requests are sent.

Authoritative XEID-prefix: an XEID-prefix delegated to a DDT node and for which the DDT node may provide further delegations of more-specific sub-prefixes.

Map-Referral: a LISP message sent by a DDT node when it receives a DDT Map-Request for an XEID that matches a configured XEID-prefix delegation. The Map-Referral message includes a "referral", a set of RLOCs for DDT nodes that have more information about the sub-prefix; a DDT client "follows the referral" by sending another DDT Map-Request to one of those RLOCs to obtain either an answer or another referral to DDT nodes responsible for a more-specific XEID-prefix. See Section 5 and Section 7.2 for details on the sending and processing of Map-Referral messages.

negative Map-Referral: a LISP message sent by a DDT node when it receives a DDT Map-Request for an EID that matches a configured authoritative XEID-prefix but for which no delegation (or registration if the DDT node is also a Map Server) is configured.

For definitions of other terms, notably Map-Request, Map-Reply, Ingress Tunnel Router (ITR), Egress Tunnel Router (ETR), Map Server, and Map Resolver, please consult the LISP specification [LISP] and the LISP Mapping Service specification [LISP-MS].

3. EID-prefix tree structure and instance IDs

LISP-DDT defines a tree structure that is indexed by a binary encoding of five fields, in order of significance: Key-ID (16 bits), Instance Identifier (IID, 32 bits), Address Family Identifier (AFI, 16 bits), and EID-prefix (variable, according to AFI value). The fields are concatenated, with the most significant fields as listed above. The index into this structure is also referred to as an Extended EID-prefix (XEID-prefix).

It is important to note that LISP-DDT does not store actual EID-to-RLOC mappings; it is, rather, a distributed index that can be used to find the devices (Map Servers and their registered EIDs) that can be queried with LISP to obtain those mappings. Changes to EID-to-RLOC mappings are made on the ETRs which define them, not to any DDT node configuration. DDT node configuration changes are only required when branches of the database hierarchy are added, removed, or modified.

4. Configuring XEID-prefix delegation

Every DDT node is configured with one or more authoritative XEID-prefixes that it "owns" along with a list of delegations of XEID-prefixes that are known to other DDT nodes. A DDT node is required to maintain a list of delegations for all sub- prefixes of its authoritative XEID-prefixes but also may list "hints", which are prefixes that it knows about that belong to its parents, to the root, or to any other point in the XEID-prefix hierarchy. A delegation (or hint) consists of an XEID-prefix and a set of RLOCs for DDT nodes that have more detailed knowledge of the XEID-prefix. Those RLOCs are returned in Map-Referral messages when the DDT node receives a DDT Map-Request with an xEID that matches a delegation. A DDT Map Server will also have a set of sub-prefixes for which it accepts ETR mapping registrations and for which it will forward (or answer, if it implements proxy mode) Map-Requests.

4.1. Example DDT node configuration

The following is an example of parent and child DDT nodes, where the parent has all of 10.0.0.0/8 and delegates two sub-prefixes, 10.0.0.0/12 and 10.0.16.0/12 to two child DDT nodes. All of these prefixes are within the DDT sub-tree Key-ID=0, IID=223, and AFI=1 (IPv4).

```
lisp ddt authoritative-prefix instance-id 223 10.0.0.0/8
lisp ddt child 192.168.1.100 instance-id 223 eid-prefix 10.0.0.0/12
lisp ddt child 192.168.1.200 instance-id 223 eid-prefix 10.16.0.0/12
```

One of the child nodes is a DDT Map Server, configured to allow ETRs to register the sub-prefixes 10.18.0.0/16 and 10.17.0.0/16:

```
lisp ddt authoritative-prefix instance-id 223 eid-prefix 10.16.0.0/12
lisp site site-1
  eid-prefix 10.18.0.0/16 instance-id 223
lisp site site-2
  eid-prefix 10.17.0.0/16 instance-id 223
```

4.2. The root DDT node

The root DDT node is the logical "top" of the database hierarchy: Key-ID=0, EID=0, AFI=0, EID-prefix=0/0. A DDT Map-Request that matches no configured XEID-prefix will be referred to the root node. The root node in a particular instantiation of LISP-DDT must therefore be configured with delegations for at least all defined IIDs and AFIs.

To aid in defining a "sub-root" DDT node that is responsible for all EID-prefixes within multiple IIDs (say, for using LISP to create virtual networks that use overlapping address space), it may be useful to implement configuration language that allows for a range of IIDs to be delegated together. Additional configuration shorthand for delegating of a range of IIDs (and all of the EIDs under them) may also be helpful.

5. DDT node operation - sending referrals

When a DDT node receives a DDT Map-Request, it compares the requested XEID against its list of XEID-prefix delegations and its list of authoritative XEID-prefixes and acts as follows:

5.1. Match of a delegated prefix (or sub-prefix)

If the requested XEID matches one of the DDT node's delegated prefixes, then a Map-Referral message is returned with the matching more-specific XEID-prefix and the set of RLOCs for the referral target DDT nodes.

Note that a matched delegation does not have to be for a sub-prefix of an authoritative prefix; in addition to being configured to delegate sub-prefixes of an authoritative prefix, a DDT node may also be configured with other XEID-prefixes for which it can provide referrals to DDT nodes anywhere in the database hierarchy. This capability to define "shortcut hints" is never required to be configured but may be a useful heuristic for reducing the number of iterations needed to find an EID, particular for private network deployments.

5.2. Missing delegation from an authoritative prefix

If the requested XEID did not match a configured delegation but does match an authoritative XEID-prefix, then the DDT node returns a negative Map-Referral that includes the least-specific XEID-prefix that does not match any of the DDT node's authoritative XEID-prefixes. This indicates that the XEID is not a LISP destination.

If the requested XEID did not match either a configured delegation or an authoritative XEID-prefix, then the request is dropped. This should only happen if either a DDT Map Resolver or DDT Map Server is misconfigured. Logging an error message may be a good idea to assist in detecting and resolving such configuration problems.

6. DDT Map Server operation

When a DDT Map Server receives a DDT Map-Request, its operation is similar to that of a DDT node with one exception: if the requested XEID matches a registered XEID-prefix, then the Map-Request is forwarded to one of the destination ETR RLOCs (or the Map-Server sends a Map-Reply, if it is providing proxy service) and a Map-Referral with the "done" flag is returned to the sender of the DDT Map-Request.

7. DDT Map Resolver operation

Just as any other Map Resolver, a DDT Map Resolver accepts Map-Requests from its clients (typically, ITRs) and ensures that those Map-Requests are forwarded to the correct ETR, which generates Map-Replies. Unlike a Map Resolver that uses the ALT mapping system [LISP-ALT], however, a DDT Map Resolver needs to maintain more state as it uses an iterative process of following referrals to find the correct ETR to answer a Map-Request.

7.1. Queuing, Sending, and Retransmitting DDT Map-Requests

When a DDT Map Resolver receives an encapsulated Map-Request, it first does a longest-match search for the XEID in its referral cache. If nothing is found or if a negative cache entry is found, then the destination is not in the database; a negative Map-Reply is returned and no further processing is done by the DDT Map Resolver.

Next, the DDT Map Resolver creates a lookup queue entry for the XEID and saves the original Map-Request along with other information, such as the longest XEID-prefix matched so far, needed for tracking progress through the iterative referral process. The Map Resolver then creates a DDT Map-Request (an encapsulated Map-Request with the "DDT-originated" flag set in the message header) for the XEID but without any authentication data that may have been included in the original Map-Request. It sends the DDT Map-Request to one of the RLOCs in the chosen referral cache entry. The referral cache is initially populated with one or more statically-configured entries; additional entries are added when referrals are followed, as described below. A DDT Map Resolver is not absolutely required to cache referrals but it not doing so will significantly increase latency and cause lookup delays.

Note that in normal use on the public Internet, the statically-configured initial referral cache for a DDT Map Resolver should include a "default" entry with RLOCs for one or more DDT nodes that can reach the DDT root node. If a Map Resolver does not have such configuration, it will return a Negative Map-Reply if it receives a query for an EID outside the subset of the mapping database known to it. While this may be desirable on private network deployments or during early transition to LISP when few sites are using it, this behavior is not appropriate when LISP is in general use on the Internet.

7.2. Receiving and following referrals

After sending a DDT Map-Request, a DDT Map Resolver can expect one of the following to occur:

- No response. The DDT Map Resolver retransmits the request, choosing a different RLOC from the referral cache entry if one is available. If the maximum number of retransmissions has occurred, then the lookup queue entry is dequeued and a negative Map-Reply is returned to the original Map-Request sender.

- A negative Map-Referral from the DDT node. This indicates that the destination XEID is not in the mapping database. The lookup queue entry is dequeued and a negative Map-Reply is returned to the original Map-Request sender. A negative referral cache entry is also created for the XEID-prefix and TTL value in the negative Map-Referral message.

- A Map-Referral with the "done" indication from the DDT node (see Section 6). This indicates that the Map-Request has been sent to a Map Server that has ETR RLOCs for the destination XEID. If the original Map-Request included a LISP-SEC ECM Authentication Data field (saved in Section 7.1, Paragraph 2) then the request is re-sent, with the Authentication Data included, to the Map Server. The Map Server will forward the request to an ETR that can provide a Map-Reply to the original Map-Request sender. This is a successful completion of the DDT iteration process, so the lookup queue entry is is dequeued. A referral cache entry is also created (or updated) for the XEID-prefix, RLOC set, and TTL value in the Map-Referral message.

- A Map-Referral from the DDT node. The DDT Map-Request is updated with the RLOCs contained in the Map-Referral, the referred prefix is updated in the lookup queue entry, and the DDT Map-Request is sent to one of the new destination DDT node RLOCs. A referral cache entry is also created (or updated) for the XEID-prefix, RLOC set, and TTL value in the Map-Referral message.

7.2.1. Handling referral errors

Other states are possible, such as a misconfigured DDT node (acting as a proxy Map Server, for example) returning a Map-Reply to the DDT Map Resolver; they should be considered errors and logged as such. It is not clear exactly what else the DDT Map-Resolver should do in such cases; one possibility is to dequeue the lookup queue entry and send a negative Map-Reply to the original Map-Request sender. Alternatively, if a DDT Map Resolver detects unexpected behavior by a DDT node, it could mark that node as unusable in its referral cache and update the lookup queue entry to try a different DDT node if more than one is listed in the referral cache.

7.2.2. Referral loop detection

With any iterative process, there is always the danger of an iteration loop. To prevent this, a DDT Map Resolver must check that it does not receive and follow a referral that is for a less-specific XEID-prefix than it has received in a previous referral. For this reason, it stores the most recent XEID-prefix received by referral in each lookup queue entry; if it receives a referral that is a less-specific match for the XEID than the last referral received, then a loop has occurred and the Map-Resolver handles the request as described in Section 7.2.1. As an extra measure to prevent referral loops, it is probably also wise to limit the total number of referrals for any request to some reasonable number; the exact value of that number will be determined during experimental deployment of LISP-DDT.

Note that when a Map-Request is originally received and an entry has been added to the lookup queue, the new request has no previous referral XEID-prefix; this means that the first DDT node contacted by a DDT Map Resolver may provide a referral to anywhere in the DDT hierarchy. This, in turn, allows a DDT Map Resolver to use essentially any DDT node RLOCs for its initial cache entries and depend on the initial referral to provide a good starting point for Map-Requests; there is no need to configure the same set of root DDT nodes in all DDT Map Resolvers.

8. Example message flow

The following describes the message flows among an ITR, a DDT Map Resolver, a number of DDT nodes, a DDT Map Server, and an ETR. It assumes no security associations between the DDT nodes but does show how [LISP-SEC] can be used between the ITR, Map Resolver, Map Server, and ETR.

8.1. ITR sends a Map-Request to a DDT Map Resolver

The first step in using LISP-DDT is the same as for any other Map-Request using the Map Server interface: an ITR sends an encapsulated Map-Request to one of its configured Map Resolvers, in this case a DDT Map Resolver. The outer header source IP address is the ITR and the outer header destination IP address is the DDT Map Resolver. If [LISP-SEC] is in use, then LISP-SEC ECM Authentication Data field is included.

8.2. DDT Map Resolver receives and processes Map-Request

The DDT Map Resolver receives and processes the encapsulated Map-Request by stripping the encapsulation header and creating a lookup queue entry for the XEID, saving the resulting, non-encapsulated Map-Request for later retransmission and re-use during the referral process. The lookup queue entry will be dequeued when the DDT Map Resolver is finished with the request (see Section 8.8).

Note that if a lookup queue entry already exists for the destination XEID and the requesting ITR (which could happen if an ITR has retransmitted a Map-Request), the Map-Request is replaced to ensure that the ITR-generated nonce and OTK are updated.

8.3. DDT Map Resolver searches referral cache for XEID

Next, the DDT Map Resolver searches its referral cache for the XEID. If none is found or if a negative cache entry is found, then the XEID does not exist in the database; a negative Map-Reply is returned to the original sender and the lookup queue entry is dequeued.

If the referral cache entry found is for a DDT Map Server, then the DDT Map Resolver has found the appropriate terminal node in the DDT hierarchy. It finishes processing the lookup queue entry as described in Section 8.8.

At this point, the referral cache entry must be for a DDT node that can provide more-specific information for the requested XEID so a DDT Map-Request is created and sent (see below).

8.4. DDT Map Resolver creates and sends DDT Map-Request

To follow a referral and query the next DDT node, the DDT Map Resolver creates a new DDT Map-Request, an encapsulated Map-Request using one of the RLOCs of the target DDT node as the outer header destination IP address and itself as the outer header source IP address. The "DDT-originated" flag is set in the encapsulation header to inform the target DDT node that it should return referrals. The original Map-Request LISP-SEC information, if any, is NOT included. The original Map-Request destination XEID is used in the new Map-Request while the source is one of the DDT Map Resolver's RLOCs.

The new "DDT Map-Request" is transmitted to the destination DDT node. If no response is received within a timeout, it is re-transmitted, preferably using a different destination DDT node RLOC. If the maximum number of retransmissions is exceeded, the request is dequeued and a negative Map-Reply is returned to the ITR that sent the original Map-Request.

8.5. DDT node receives and processes DDT Map-Request

The destination DDT node searches its configured delegations and authoritative prefixes for the XEID in the received encapsulated Map-Request. If no match is found, then the DDT Map-Request is silently discarded and, optionally, an error is logged.

If a delegation is found, the DDT node sends a Map-Referral message back to the DDT Map Resolver with the matched XEID-prefix and the set of RLOCs for DDT nodes that can be used to resolve XEIDs within that prefix.

If no matching delegation was found and the XEID matches one of the DDT node's authoritative prefixes, then the destination is not a LISP XEID (or a configuration error has occurred); the DDT node returns a negative Map-Referral message to the DDT Map Resolver as described in Section 5.2.

8.6. DDT Map Resolver processes Map-Referral

When the DDT Map Resolver receives a Map-Referral from a DDT-node, it first verifies that it has a corresponding lookup queue entry; if none can be found, then the Map-Referral is silently ignored, with optional error logging.

If the received Map-Referral was negative, then the destination XEID is not in the database; a negative Map-Reply is returned to the original Map-Request sender, a negative referral cache entry is

created for the returned XEID-prefix (with TTL from the Map-Referral message), and the lookup queue entry is dequeued.

For a non-negative Map-Referral, the lookup queue entry is updated with the new referral XEID-prefix and new DDT-node RLOCs. At this point, it also checks to make sure that a referral loop has not occurred (see Section 7.2.2).

To speed processing of future Map-Requests for the same XEID-prefix, the DDT Map Resolver adds a new entry (or updates an existing, matching entry) in its referral cache for the XEID-prefix, RLOC set, and TTL value in the Map-Referral message. Finally, processing continues to Section 8.4 to query the new destination DDT-node.

8.7. DDT Map Server receives Map-Request

At this point, the DDT Map Resolver has found the DDT Map Server responsible for the destination XEID-prefix and has sent its Map-Request there. The DDT Map Server receives the DDT Map-Request, strips the encapsulation header, and searches for the destination XEID in its set of configured XEID-prefixes. If the XEID is found and an ETR has registered for it, then DDT Map Server returns a Map-Referral to the DDT Map Resolver indicating (by setting the "done" flag) that it has found the terminal DDT node. If no LISP-SEC header was included in the original Map-Request, then the Map-Request is forwarded to one of the registered ETRs for further processing (Section 8.10); otherwise, the Map-Request is discarded so that the DDT Map Resolver can re-send it to the DDT Map Server with LISP-SEC information included.

8.8. DDT Map Resolver finished

At this point, the DDT Map Resolver has finished the referral iteration process. If security processing was requested, the DDT Map Resolver now re-sends the DDT Map-Request to the DDT Map Server with the LISP-SEC information included in the encapsulation header. The DDT Map Resolver dequeues the lookup queue entry for the XEID and cleans-up any other saved state.

8.9. DDT Map Server receives LISP-SEC-enabled Map-Request

When the DDT Map Server receives the re-sent DDT Map-Request, with LISP-SEC information included, it decrypts the LISP-SEC information, performs normal LISP-SEC processing, and forwards the resulting Map-Request to the target ETR.

8.10. ETR sends Map-Reply to ITR

The ETR receives a Map-Request as documented in [LISP], performs any necessary processing of security information, as documented in [LISP-SEC], and sends a Map-Reply to the ITR that sent the original Map-Request.

9. Open Issues and Considerations

There are a number of issues with the organization of the mapping database that need further investigation. Among these are:

- Unlike in [LISP-ALT], DDT does not currently define a mechanism for propagating ETR-to-Map Server registration state. This requires DDT Map Servers to suppress returning negative Map-Reply messages for defined but unregistered XEID-prefixes to avoid loss of connectivity during partial ETR registration failures. Suppressing these messages may cause a delay for an ITR obtaining a mapping entry when such a failure is occurring.

- Defining an interface to implement interconnection and/or interoperability with other ased mapping databases, such as LIST+ALT.

- Additional key structures for use with LISP-DDT, such as to support additional EID formats as defined in [LCAF].

- Authentication of delegations between DDT nodes.

- Possibility of a new, more general format for the Map-Referral messages to facilitate the use of LISP-DDT with additional Key-ID/IID/EID combinations. Currently-defined packet formats should be considered to be preliminary and provisional until this issue has received greater attention.

- Management of the DDT Map Resolver referral cache, in particular, detecting and removing outdated entries.

The authors expect that experimentation on the LISP pilot network will help answer open questions surrounding these and other issues.

10. IANA Considerations

This document makes no request of the IANA.

11. Security Considerations

Future revisions of this document will add public/private key pairing and use between DDT nodes. DDT will not rely on external security infrastructure.

12. References

12.1. Normative References

[LCAF] Farinacci, D. and J. Snijders, "LISP Canonical Address Format", draft-ietf-lisp-lcaf-06.txt (work in progress), October 2011.

[LISP] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol (LISP)", draft-ietf-lisp-16.txt (work in progress), October 2011.

[LISP-MS] Fuller, V. and D. Farinacci, "LISP Map Server Interface", draft-ietf-lisp-ms-12.txt (work in progress), October 2011.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[RFC4634] Eastlake, D. and T. Hansen, "US Secure Hash Algorithms (SHA and HMAC-SHA)", RFC 4634, July 2006.

12.2. Informative References

[LISP-ALT] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "LISP Alternative Topology (LISP-ALT)", draft-ietf-lisp-alt-10.txt (work in progress), October 2011.

[LISP-SEC] Maino, F., Ermagan, V., Cabellos, A., Sanchez, D., and O. Bonaventure, "LISP-Security", draft-maino-lisp-sec-00.txt (work in progress), July 2011.

Appendix A. Acknowledgments

The authors with to express their thanks to Damien Saucez, Lorand Jakab, and Olivier Bonaventure for work on LISP-TREE and LISP iterable mappings that inspired the hierarchical database structure and lookup iteration approach described in this document. Special thanks also go to Vina Ermagan, Amit Jain, Isidor Kouvelas, Jesper Skriver, Andrew Partan, and Noel Chiappa, all of whom have participated in (and put up with) seemingly endless hours of discussion of LISP mapping database ideas and issues.

Appendix B. Map-Referral Message Format

```
        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
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |Type=6 |D|M|            Reserved               | Record Count  |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         Nonce . . .                           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         . . . Nonce                           |
   +-> +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   |                          Record  TTL                          |
   |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   R   | Locator Count | EID mask-len  | ACT |A|      Reserved         |
   e   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   c   |         Reserved             |            EID-AFI            |
   o   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   r   |                          EID-prefix ...                       |
   d   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  /|    Priority   |    Weight     |  M Priority   |   M Weight    |
   | L +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | o |        Unused Flags         |R|           Loc-AFI             |
   | c +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  \|                             Locator ...                       |
   +-> +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

D (the "done flag") is set when a DDT Map Server sends a Map-Referral to indicate that processing is "done". M (the "DDT Map Server flag") indicates that the a DDT Map Server has found a matching, registered XEID-prefix for the XEID in the original Map-Request.

All the field descriptions are equivalent to those in the Map-Reply message, as defined in [LISP]. Note, though, that the set of RLOCs correspond to the DDT node to be queried as a result of the referral not the RLOCs for an actual EID-to-RLOC mapping.

Appendix C. Encapsulated Control Message Format

```
        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
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |                       IPv4 or IPv6 Header                     |
   OH  |                      (uses RLOC addresses)                    |
     \ |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |       Source Port = xxxx      |       Dest Port = 4342        |
   UDP +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     \ |           UDP Length          |        UDP Checksum           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   LH  |Type=8 |S|D|                Reserved                           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |                       IPv4 or IPv6 Header                     |
   IH  |                  (uses RLOC or EID addresses)                 |
     \ |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |       Source Port = xxxx      |       Dest Port = yyyy        |
   UDP +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     \ |           UDP Length          |        UDP Checksum           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   LCM |                      LISP Control Message                     |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

"D" is the "DDT-originated" flag and is set by a DDT client to indicate that the receiver can and should return Map-Referral messages as appropriate.

Authors' Addresses

Vince Fuller
cisco Systems
Tasman Drive
San Jose, CA 95134
USA

Email: vaf@cisco.com

Darrel Lewis
cisco Systems
Tasman Drive
San Jose, CA 95134
USA

Email: darlewis@cisco.com

Dino Farinacci
cisco Systems
Tasman Drive
San Jose, CA 95134
USA

Email: dino@cisco.com