Network Working Group
Internet-Draft
Intended status: Experimental
Expires: December 2, 2016

V. Fuller

D. Lewis
V. Ermagan
Cisco Systems
A. Jain
Juniper Networks
A. Smirnov
Cisco Systems

May 31, 2016

# LISP Delegated Database Tree
# draft-ietf-lisp-ddt-07

Abstract

This document 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 as "authoritative" for one or more EID-prefixes, along with the set of RLOCs for Map Servers or "child" DDT nodes to which more-specific EID-prefixes are delegated.

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 December 2, 2016.

Copyright Notice

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

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

Table of Contents

1. Introduction . . . . . 3
2. Requirements Language . . . . . 4
3. Definition of Terms . . . . . 5
4. Database organization . . . . . 7
4.1. EID-prefix tree structure and instance IDs . . . . . 7
4.2. Configuring prefix delegation . . . . . 7
4.2.1. The root DDT node . . . . . 7
5. DDT Map-Request . . . . . 8
6. The Map-Referral message . . . . . 8
6.1. Action codes . . . . . 9
6.2. Referral set . . . . . 9
6.3. Incomplete flag . . . . . 10
6.4. Map-Referral Message Format . . . . . 10
6.4.1. SIG section . . . . . 12
7. DDT network elements and their operation . . . . . 13
7.1. DDT node . . . . . 14
7.1.1. Match of a delegated prefix (or sub-prefix) . . . . . 14
7.1.2. Missing delegation from an authoritative prefix . . . 14
7.2. DDT Map Server . . . . . 14
7.3. DDT Map Resolver . . . . . 15
7.3.1. Queuing and sending DDT Map-Requests . . . . . 15
7.3.2. Receiving and following referrals . . . . . 16
7.3.3. Handling referral errors . . . . . 17
7.3.4. Referral loop detection . . . . . 18
8. Pseudo Code and Decision Tree diagrams . . . . . 18
8.1. Map Resolver processing of ITR Map-Request . . . . . 19
8.1.1. Pseudo-code summary . . . . . 19
8.1.2. Decision tree diagram . . . . . 19
8.2. Map Resolver processing of Map-Referral message . . . . . 20
8.2.1. Pseudo-code summary . . . . . 20
8.2.2. Decision tree diagram . . . . . 22
8.3. DDT Node processing of DDT Map-Request message . . . . . 24
8.3.1. Pseudo-code summary . . . . . 24
8.3.2. Decision tree diagram . . . . . 25
9. Example topology and request/referral following . . . . . 25
9.1. Lookup of 2001:db8:0103:1::1/128 . . . . . 27

9.2. Lookup of 2001:db8:0501:8:4::1/128 . . . . . 28
9.3. Lookup of 2001:db8:0104:2::2/128 . . . . . 29
9.4. Lookup of 2001:db8:0500:2:4::1/128 . . . . . 30
9.5. Lookup of 2001:db8:0500::1/128 (non-existent EID) . . . . 30
10. Securing the database and message exchanges . . . . . 31
10.1. XEID-prefix Delegation . . . . . 31
10.2. DDT node operation . . . . . 32
10.2.1. DDT public key revocation . . . . . 32
10.3. Map Server operation . . . . . 33
10.4. Map Resolver operation . . . . . 33
11. Open Issues and Considerations . . . . . 34
12. IANA Considerations . . . . . 34
13. Security Considerations . . . . . 34
14. References . . . . . 35
14.1. Normative References . . . . . 35
14.2. Informative References . . . . . 35
Appendix A. Acknowledgments . . . . . 36
Authors' Addresses . . . . . 36

## 1. Introduction

LISP [RFC6830] specifies an architecture and mechanism for replacing the addresses currently used by IP with two separate name spaces: Endpoint Identifiers (EIDs), used end-to-end for terminating transport-layer associations, and Routing Locators (RLOCs), which 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: Database-ID (DBID, 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 DBID 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 as authoritative for one or more XEID-prefixes. It also is configured with the set of more-specific sub-prefixes 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 either points to a Map Server (sometimes termed a "terminal DDT node") to which an Egress Tunnel Router (ETR) has registered that sub-prefix or points to another DDT node in the database tree that further delegates the sub-prefix. See [RFC6833] 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. See Section 6 for the definition of the Map-Referral message.

To find an EID-to-RLOC mapping, a LISP-DDT client, usually a DDT Map Resolver, starts by sending an Encapsulated Map-Request to a preconfigured DDT node RLOC. The DDT node responds with a Map-Referral message that either indicates that it will find the requested mapping to complete processing of the request or that the DDT client should contact another DDT node that has more-specific information; in the latter case, the DDT node then sends a new Encapsulated Map-Request to the next DDT node and the process repeats in an iterative manner.

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. 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].

## 3. 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 [RFC1918] address space is used. See "Using Virtualization and Segmentation with LISP" in [RFC6830] for more discussion of Instance IDs.

XEID-prefix: a LISP EID-prefix with 16-bit LISP-DDT DBID (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. Encoding of the prefix, its AFI and the instance ID (IID) are specified by [I-D.ietf-lisp-lcaf]. 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 Map-Reply 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 pending request list, then queries one or more DDT nodes for the requested EID, following returned referrals until it receives one with action code MS-ACK (or an error indication). MS-ACK 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") and a pending request list 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 [RFC6833].

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 7.3.1 describes how DDT Map-Requests are sent. Section 5 defines position of the "DDT-originated" flag in the Encapsulated Control Message header.

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 in response to a DDT Map-Request for an XEID that matches a configured XEID-prefix delegation. A non-negative Map-Referral 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 7.1 and Section 7.3.2 for details on the sending and processing of Map-Referral messages.

Negative Map-Referral: a Map-Referral sent in response to a DDT Map-Request that matches an authoritative XEID-prefix but for which there is no delegation configured (or no ETR registration if sent by a DDT Map-Server).

Pending Request List: the set of outstanding requests for which a DDT Map Resolver has received encapsulated Map-Requests from a DDT client for an XEID. Each entry in the list contains additional state needed by the referral following process, including the requestor(s) of the XEID (typically, one or more ITRs), saved information about the last referral received and followed (matching XEID-prefix, action code, RLOC set, index of last RLOC queried in the RLOC set), and any LISP-SEC information ([I-D.ietf-lisp-sec]) that was included in the DDT client Map-Request. An entry in the list may be interchangeably termed a "pending request list entry" or simply a "pending request".

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 [RFC6830] and the LISP Mapping Service specification [RFC6833].

# 4. Database organization

## 4.1. 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: DBID (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.

## 4.2. Configuring prefix delegation

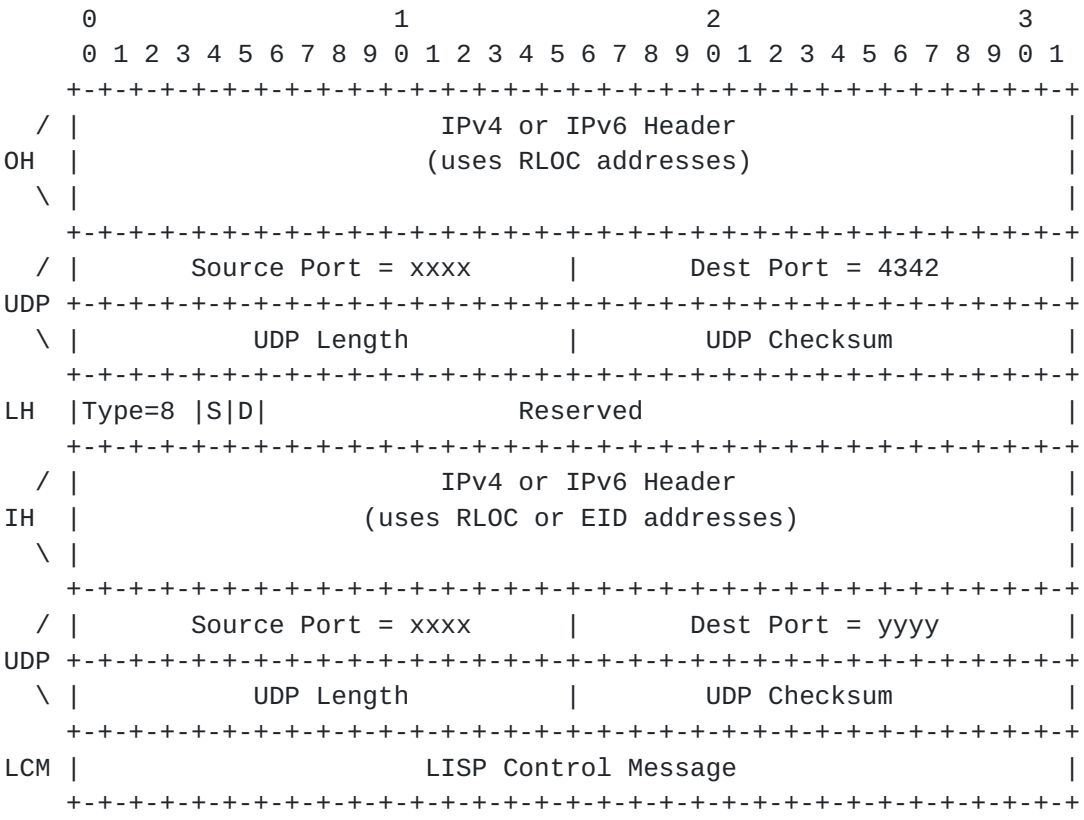
Every DDT node is configured with one or more XEID-prefixes for which it is authoritative along with a list of delegations of XEID-prefixes to other DDT nodes. A DDT node is required to maintain a list of delegations for all sub-prefixes of its authoritative XEID-prefixes; it also may list "hints", which 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, a set of RLOCs for DDT nodes that have more detailed knowledge of the XEID-prefix, and accompanying security information (for details of security infomation exchange and its use see Section 10). 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 provides proxy Map-Reply service) Map-Requests.

### 4.2.1. The root DDT node

The root DDT node is the logical "top" of the database hierarchy: DBID=0, IID=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.

## 5. DDT Map-Request

A DDT client (usualy a Map Resolver) uses LISP Encapsulated Control Message (ECM) to send Map-Request to a DDT node. Format of the ECM is defined by [RFC6830]. This specification adds to ECM flag "DDT-originated".




D: The "DDT-originated" flag. It is set by a DDT client to indicate that the receiver SHOULD return Map-Referral messages as appropriate. Use of the flag is further described in Section 7.3.1. This bit is allocated from LISP message header bits marked as Reserved in [RFC6830].

## 6. The Map-Referral message

This specification defines a new LISP message, the Map-Referral. It is sent by a DDT node to a DDT client in response to a DDT Map-Request message. See Section 6.4 for a detailed layout of the Map-Referral message fields.

The message consists of an action code along with delegation information about the XEID-prefix that matches the requested XEID.

### 6.1. Action codes

The action codes are as follows:

NODE-REFERRAL (0): indicates that the replying DDT node has delegated an XEID-prefix that matches the requested XEID to one or more other DDT nodes. The Map-Referral message contains a "map-record" with additional information, most significantly the set of RLOCs to which the prefix has been delegated, that is used by a DDT Map Resolver to "follow" the referral.

MS-REFERRAL (1): indicates that the replying DDT node has delegated an XEID-prefix that matches the requested XEID to one or more DDT Map Servers. It contains the same additional information as a NODE-REFERRAL, but is handled slightly differently by the receiving DDT client (see Section 7.3.2).

MS-ACK (2): indicates that a replying DDT Map Server received a DDT Map-Request that matches an authoritative XEID-prefix for which it has one or more registered ETRs. This means that the request has been forwarded to one of those ETRs to provide an answer to the querying ITR.

MS-NOT-REGISTERED (3): indicates that the replying DDT Map Server received a Map-Request for one of its configured XEID-prefixes which has no ETRs registered.

DELEGATION-HOLE (4): indicates that the requested XEID matches a non-delegated sub-prefix of the XEID space. This is a non-LISP "hole", which has not been delegated to any DDT Map Server or ETR. See Section 7.1.2 for details.

NOT-AUTHORITATIVE (5): indicates that the replying DDT node received a Map-Request for an XEID-request for which it is not authoritative. This can occur if a cached referral has become invalid due to a change in the database hierarchy.

### 6.2. Referral set

For "positive" action codes (NODE-REFERRAL, MS-REFERRAL, MS-ACK), a DDT node includes in the Map-Referral message a list of RLOCs for all DDT nodes that are authoritative for the XEID-prefix being returned; a DDT Map Resolver uses this information to contact one of those DDT nodes as it "follows" a referral.

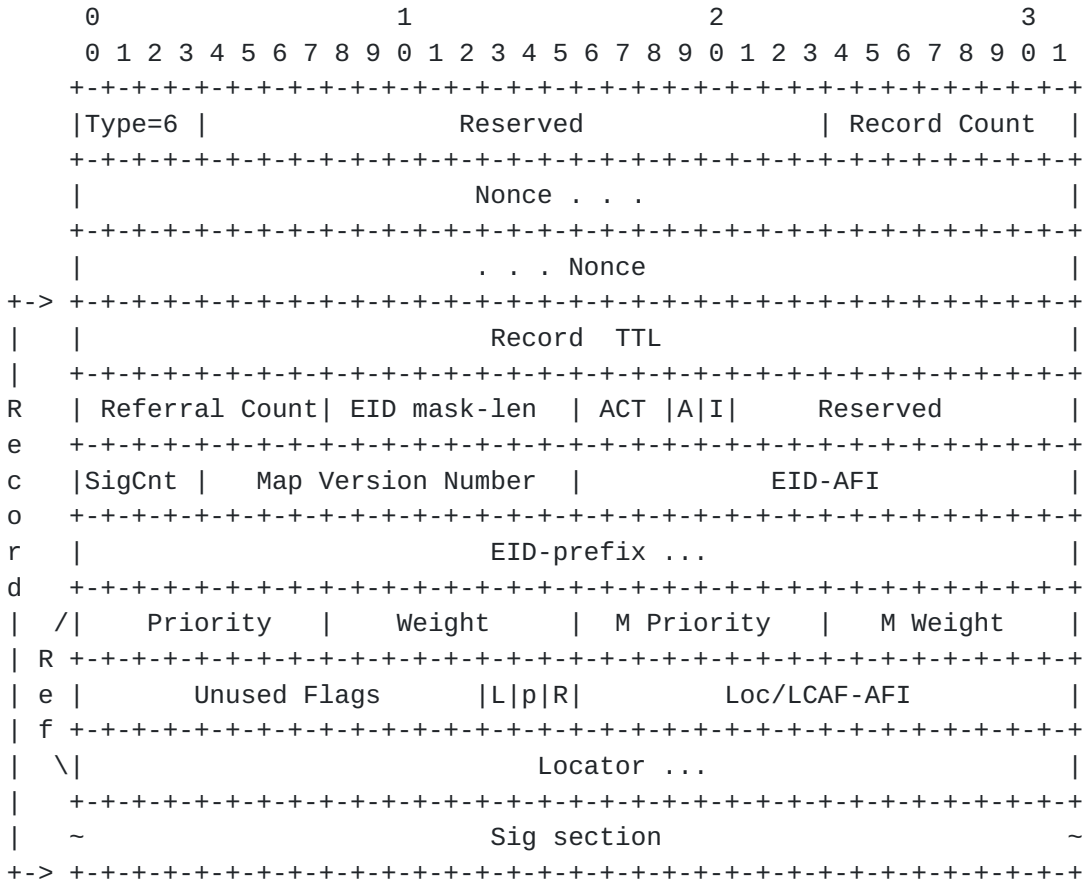
### 6.3. Incomplete flag

A DDT node sets the "Incomplete" flag in a Map-Referral message if the Referral Set is incomplete; this is intended to prevent a DDT Map Resolver from caching a referral with incomplete information. A DDT node must set the "incomplete" flag in the following cases:

- o If it is setting action code MS-ACK or MS-NOT-REGISTERED but does not have configuration for other "peer" DDT nodes that are also authoritative for the matched XEID-prefix.

- o If it is setting action code NOT-AUTHORITATIVE.

### 6.4. Map-Referral Message Format




Type: Type value 6 was reserved for future use in RFC6830, this document allocates this value to identify Map-Referral messages.

ACT: The "action" field of the mapping record in a Map-Referral message encodes 6 action types. The values for the action types are:

NODE-REFERRAL (0): Sent by a DDT node with a child delegation, which is authoritative for the EID.

MS-REFERRAL (1): Sent by a DDT node that has information about Map Server(s) for the EID but it is not one of the Map Servers listed, i.e. the DDT-Node sending the referral is not a Map Server.

MS-ACK (2): Sent by a DDT Map Server that has one or more ETR registered for the EID.

MS-NOT-REGISTERED (3): Sent by a DDT Map Server that is configured for the EID-prefix, but for which no ETRs are registered.

DELEGATION-HOLE (4): Sent by an intermediate DDT node with authoritative configuration covering the requested EID but without any child delegation for the EID. Also sent by a DDT Map Server with authoritative configuration covering the requested EID, but for which no specific site ETR is configured.

NOT-AUTHORITATIVE (5): Sent by a DDT node that does not have authoritative configuration for the requested EID. The EID-prefix returned MUST be the original requested EID and the TTL MUST be set to 0. However, if such a DDT node has a "hint" delegation covering the requested EID, it MAY choose to return NODE-REFERRAL or MS-REFERRAL as appropriate.

Incomplete: The "I" bit indicates that a DDT node's referral-set of locators is incomplete and the receiver of this message SHOULD NOT cache the referral. A DDT sets the "incomplete" flag, the TTL, and the Action Type field as follows:

| Type | (Action field) | Incomplete | Referral-set | TTL values |
|---|---|---|---|---|
| 0 | NODE-REFERRAL | NO | YES | 1440 |
| 1 | MS-REFERRAL | NO | YES | 1440 |
| 2 | MS-ACK | * | * | 1440 |
| 3 | MS-NOT-REGISTERED | * | * | 1 |
| 4 | DELEGATION-HOLE | NO | NO | 15 |
| 5 | NOT-AUTHORITATIVE | YES | NO | 0 |

*: The "Incomplete" flag setting on Map Server originated referral of MS-REFERRAL and MS-NOT-REGISTERED types depend on whether the Map Server has the full peer Map Server configuration for the same prefix and has encoded the information in the mapping record. Incomplete bit is not set when the Map Server has encoded the information, which means the referral-set includes all the RLOCs of all Map Servers that serve the prefix. It is set when the Map Server has not encoded the Map Server set information.

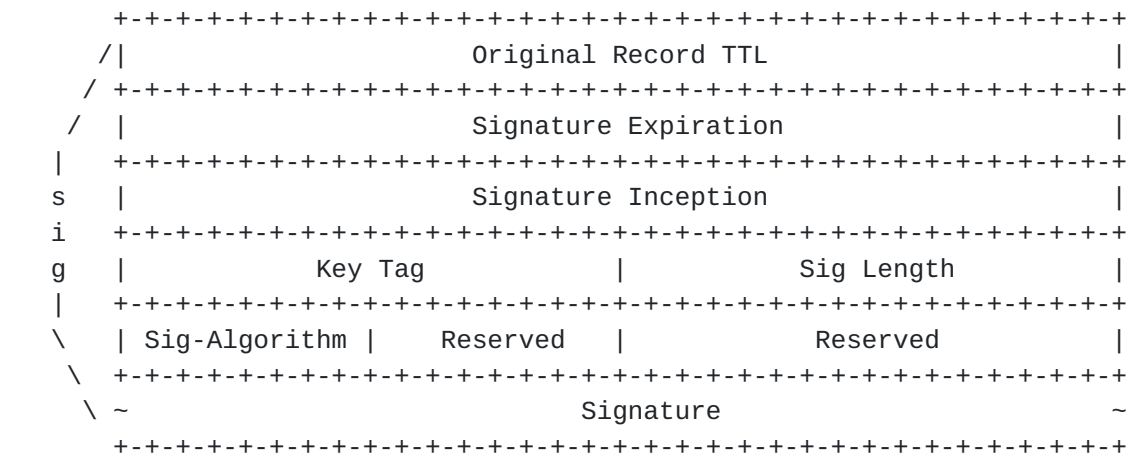
SigCnt: Indicates the number of signatures (sig section) present in the Record. If SigCnt is larger than 0, the signature information captured in a sig section as described in Section 6.4.1 will be appended to the end of the record. The number of sig sections at the end of the Record must match the SigCnt. Note that bits occupied by SigCnt were Reserved in Records embedded into messages defined by [RFC6830] and were required to be set to zero.

Loc/LCAF-AFI: If this is a Loc AFI, keys are not included in the record. If this is a LCAF AFI, the contents of the LCAF depend on the Type field of the LCAF. Security material are stored in LCAF Type 11. DDT nodes and Map Servers can use this LCAF Type to include public keys associated with their Child DDT nodes for a XEID-prefix referral record. LCAF types and formats are defined in [I-D.ietf-lisp-lcaf].

All other fields and their descriptions are equivalent to those in the Map-Reply message, as defined in LISP [RFC6830]. 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.

### 6.4.1. SIG section

If SigCnt field in the Map-Referral is not 0, the signature information is included at the end of captured in a sig section as described below. SigCnt counts the number of sig sections that appear at the end of the Record.




Original Record TTL: The original Record TTL for this record that is covered by the signature. Record TTL is in minutes.

Signature Expiration and Inception: Specify the validity period for the signature. The signature MUST NOT be used for authentication prior to the inception date and MUST NOT be used for authentication after the expiration date. Each field specifies a date and time in the form of a 32-bit unsigned number of seconds elapsed since 1 January 1970 00:00:00 UTC, ignoring leap seconds, in network byte order.

Key Tag: An identifier to specify which key is used for this signature if more than one valid key exists for the signing DDT node.

Sig Length: The length of the Signature field.

Sig-Algorithm: The identifier of the cryptographic algorithm used for the signature. This specification defines only one algorithm: Sig-Algorithm type 1 is RSA-SHA1 (see [RFC3447]).

Reserved: This field must be set to 0 on transmit and must be ignored on receipt.

Signature: Contains the cryptographic signature that covers the entire record. The Record TTL and the sig fields are set to zero for the purpose of computing the Signature.

## 7. DDT network elements and their operation

As described above, DDT introduces a new network element, the "DDT node", extends the functionality of Map Servers and Map Resolvers to send and receive Map-Referral messages. The operation of each of these devices is described as follows.

### 7.1. DDT node

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:

#### 7.1.1. 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 including associated security information (see Section 10 for details on security). If the delegation is known to be a DDT Map Server, then the Map-Referral message is sent with action code MS-REFERRAL to indicate to the receiver that LISP-SEC information (if saved in the pending request) should be included in the next DDT Map-Request; otherwise, the action code NODE-REFERRAL is used.

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.

#### 7.1.2. 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 uses the least-specific XEID-prefix that does not match any XEID-prefix delegated by the DDT node. The action code is set to DELEGATION-HOLE; 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 and a negative Map-Referral with action code NOT-AUTHORITATIVE is returned.

### 7.2. DDT Map Server

When a DDT Map Server receives a DDT Map-Request, its operation is similar to that of a DDT node with additional processing as follows:

- 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 Map-Reply service) and a Map-Referral with the MS-ACK action is returned to the sender of the DDT Map-Request.

- If the requested XEID matches a configured XEID-prefix for which no ETR registration has been received then a negative Map-Referral with action code MS-NOT-REGISTERED is returned to the sender of the DDT Map-Request.

### 7.3. DDT Map Resolver

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 (see [RFC6836]), however, a DDT Map Resolver uses an iterative process of following referrals to find the correct ETR to answer a Map-Request; this requires a DDT Map Resolver to maintain additional state: a Map-Referral cache and pending request list of XEIDs that are going through the iterative referral process.

#### 7.3.1. Queuing and sending DDT Map-Requests

When a DDT Map Resolver receives an encapsulated Map-Request, it first performs a longest-match search for the XEID in its referral cache. If no match 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 performed by the DDT Map Resolver.

If a match is found, the DDT Map Resolver creates a pending request list entry for the XEID and saves the original Map-Request (minus the encapsulation header) along with other information needed to track progress through the iterative referral process; the "referral XEID-prefix" is also initialized to the null value since no referral has yet been received. The Map Resolver then creates a DDT Map-Request (which is 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 doing so decreases latency and reduces 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.

### 7.3.2. Receiving and following referrals

After sending a DDT Map-Request, a DDT Map Resolver expects to receive a Map-Referral response. If none occurs within the timeout period, the DDT Map Resolver retransmits the request, sending to the next RLOC in the referral cache entry if one is available. If the maximum number of retransmissions has occurred and all RLOCs have been tried, then the pending request list entry is dequeued.

A DDT Map Resolver processes a received Map-Referral message according to each action code:

NODE-REFERRAL: The DDT Map Resolver checks for a possible referral loop as as described in Section 7.3.4. If no loop is found, the DDT Map Resolver saves the prefix returned in the Map-Referral message in the referral cache, updates the saved prefix and saved RLOCs in the pending request list entry, and follows the referral by sending a new DDT Map-Request to one of the DDT node RLOCs listed in the Referral Set; security information saved with the original Map-Request is not included.

MS-REFERRAL: The DDT Map Resolver follows an MS-REFERRAL in the same manner as a NODE-REFERRAL except that that security information saved with the original Map-Request is included in the new Map-Request sent to a Map Server (see Section 10 for details on security).

MS-ACK: This is returned by a DDT Map Server to indicate that it has one or more registered ETRs that can answer a Map-Request for the XEID. If the pending request did not include saved LISP-SEC information or if that information was already included in the previous DDT Map-Request (sent by the DDT Map Resolver in response to either an MS-REFERRAL or a previous MS-ACK referral), then the pending request for the XEID is complete and is dequeued. Otherwise, LISP-SEC information is required and has not yet been sent to the authoritative DDT Map-Server; the DDT Map Resolver re-sends the DDT Map-Request with LISP-SEC information included and

the pending request queue entry remains until another Map-Referral with MS-ACK action code is received. If the "incomplete" flag is not set, the prefix is saved in the referral cache.

MS-NOT-REGISTERED: The DDT Map Server queried could not process the request because it did not have any ETRs registered for a matching, authoritative XEID-prefix. If the DDT Map Resolver has not yet tried all of the RLOCs saved with the pending request, then it sends a Map-Request to the next RLOC in that list. If all RLOCs have been tried, then the destination XEID is not registered and is unreachable. The DDT Map Resolver returns a negative Map-Reply to the original Map-Request sender; this Map-Reply contains the non-registered XEID-prefix with TTL value of one minute. A negative referral cache entry is created for the prefix (also with TTL of one minute) and the pending request is dequeued.

DELEGATION-HOLE: The DDT Map Server queried did not have an XEID-prefix defined that matched the requested XEID so it does not exist in the mapping database. The DDT Map Resolver returns a negative Map-Reply to the original Map-Request sender; this Map-Reply will indicate the least-specific XEID-prefix matching the requested XEID for which no delegations exist and will have a TTL value of 15 minutes. A negative referral cache entry is created for the prefix (also with TTL of 15 minutes) and the pending request is dequeued.

NOT-AUTHORITATIVE: The DDT Map Server queried is not authoritative for the requested XEID. This can occur if a cached referral has become invalid due to a change in the database hierarchy. If the DDT Map Resolver receiving this message can determine that it is using old cached information, it MAY choose to delete that cached information and re-try the original Map-Request, starting from its "root" cache entry. If this action code is received in response to a query that did not use a cached referral information, then it indicates a database synchronization problem or configuration error. The pending request list entry that caused this answer is removed, with no answer returned to the original requestor.

### 7.3.3. 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 remove the pending request list 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 pending request to try a different DDT node if more than one is listed in the referral cache. In any case, any prefix contained in a Map-Referral message that causes a referral error (including a referral loop) is not saved in the DDT Map-Resolver referral cache.

### 7.3.4. Referral loop detection

In response to a Map-Referral message with action code NODE-REFERRAL or MS-REFERRAL, a DDT Map Resolver is directed to query a new set of DDT node RLOCs that are expected to have more-specific XEID-prefix information for the requested XEID. To prevent a possible "iteration loop" (following referrals back-and-forth among a set of DDT nodes without ever finding an answer), a DDT Map Resolver saves the last received referral XEID-prefix for each pending request and checks that a newly received NODE-REFERRAL or MS-REFERRAL message contains a more-specific referral XEID-prefix; an exact or less-specific match of the saved XEID-prefix indicates a referral loop. If a loop is detected, the DDT Map Resolver handles the request as described in Section 7.3.3. Otherwise, the Map Resolver saves the most recently received referral XEID-prefix with the pending request when it follows the referral.

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, but is bounded by the maximum length of the XEID.

Note that when a DDT Map Resolver adds an entry to its lookup queue and sends an initial Map-Request for an XEID, the queue entry 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 on all DDT Map Resolvers.

## 8. Pseudo Code and Decision Tree diagrams

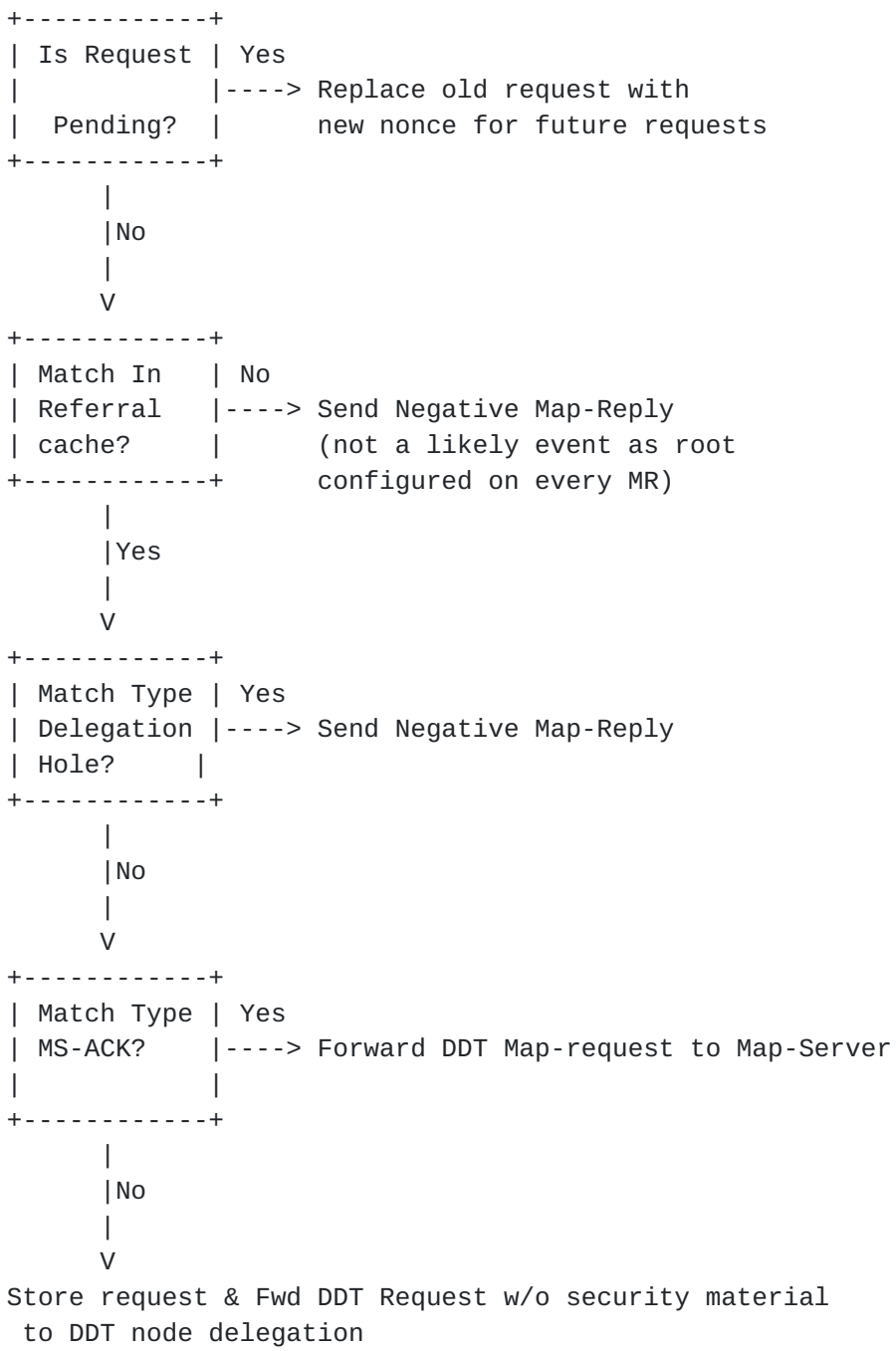
To aid in implementation, each of the major DDT Map Server and DDT Map Resolver functions are described below, first using simple "psuedo-code" and then in the form of a decision tree.

## 8.1. Map Resolver processing of ITR Map-Request

### 8.1.1. Pseudo-code summary

```
if ( request pending i.e., (ITR,EID) of request same ) {
    replace old request with new & use new request nonce
     for future requests
} else if ( no match in refcache ) {
    return negative map-reply to ITR
} else if ( match type delegation hole ) {
    return negative map-reply to ITR
} else if ( match type ms-ack ) {
    fwd DDT request to map-server
} else {
    store & fwd DDT request w/o security material to node delegation
}
```

### 8.1.2. Decision tree diagram




## 8.2. Map Resolver processing of Map-Referral message

### 8.2.1. Pseudo-code summary

```
if ( no request pending matched by referral nonce ) {
    silently drop
}

if ( pfx in referral less specific than last referral used ) {
```

```
          if ( gone through root ) {
              silently drop
          } else {
              send request to root
          }
      }

      switch (map_referral_type) {

          case NOT_AUTHORITATIVE :
              if ( gone through root ) {
                  return negative map-reply to ITR
              } else {
                  send request to root
              }

          case DELEGATION_HOLE:
              cache & send negative map-reply to ITR

          case MS_REFERRAL:
              if ( referral equal to last used ) {
                  if ( gone through root ) {
                      return negative map-reply to ITR
                  } else {
                      send request to root
                  }
              } else {
                  cache
                  follow the referral, include security material
              }

          case NODE_REFERRAL:
              if ( referral equal to last used ) {
                  if ( gone through root ) {
                      return negative map-reply to ITR
                  } else {
                      send request to root
                  }
              } else {
                  cache
                  follow the referral, strip security material
              }

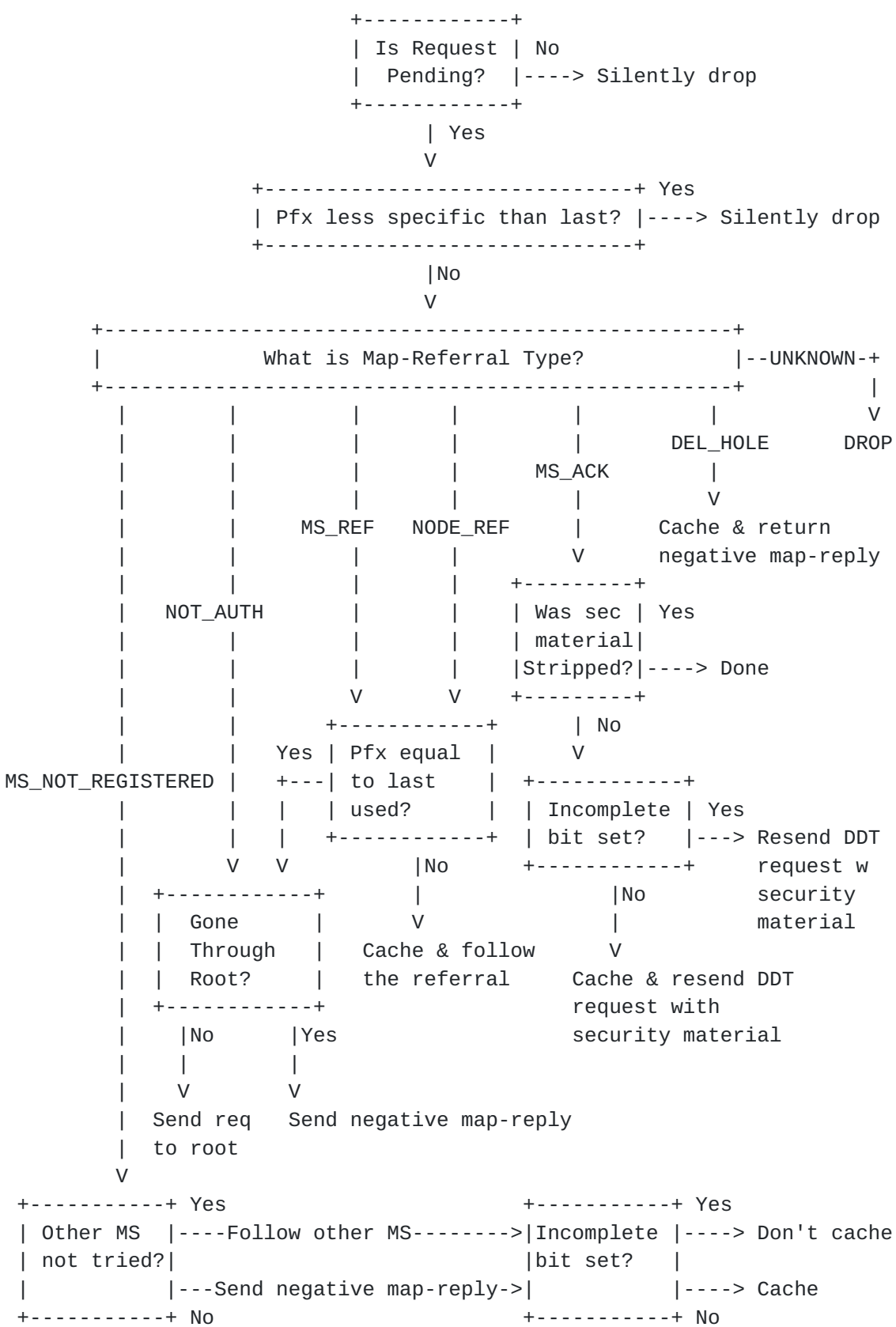
          case MS_ACK:
              if ( security material stripped ) {
                  resend request with security material
                  if { !incomplete } {
                      cache
```

```
                }
            }

        case MS_NOT_REGISTERED:
            if { all map-server delegations not tried } {
                follow delegations not tried
                if ( !incomplete ) {
                    cache
                }
            } else {
                send negative map-reply to ITR
                if { !incomplete } {
                    cache
                }
            }

        case DEFAULT:
            drop
        }
    }
```

### 8.2.2. Decision tree diagram


Is Request Pending?
No
Silently drop
Yes
Pfx less specific than last?
Yes
Silently drop
No
What is Map-Referral Type?
UNKNOWN
DEL_HOLE
DROP
MS_ACK
Cache & return negative map-reply
MS_REF
NODE_REF
NOT_AUTH
Was sec material Stripped?
Yes
Done
No
MS_NOT_REGISTERED
Yes
Pfx equal to last used?
Incomplete bit set?
Yes
Resend DDT request w security material
No
Gone Through Root?
No
Yes
Cache & follow the referral
Cache & resend DDT request with security material
Send req to root
Send negative map-reply
Other MS not tried?
Yes
Follow other MS
Send negative map-reply
No
Incomplete bit set?
Yes
Don't cache
Cache
No

### 8.3. DDT Node processing of DDT Map-Request message

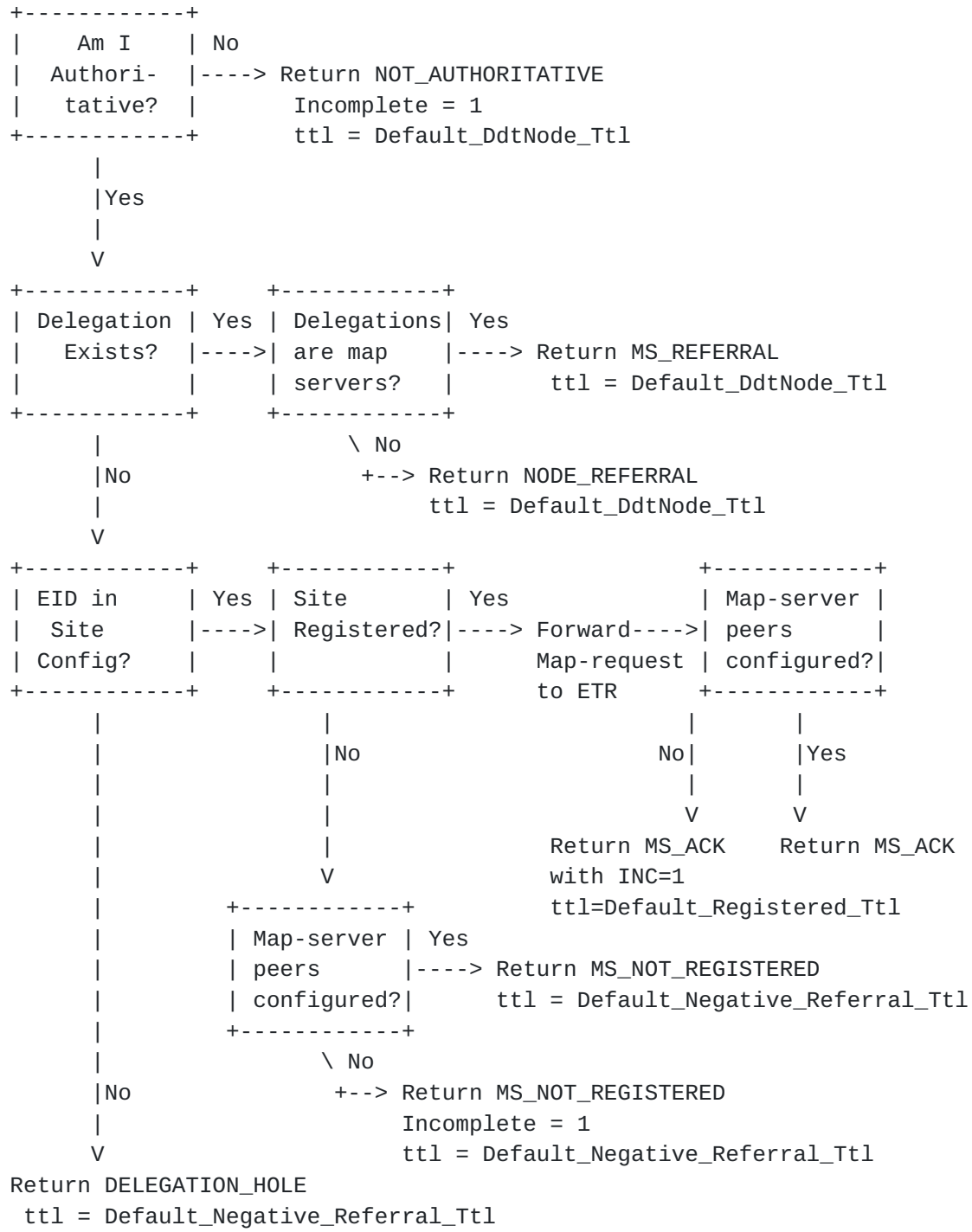
#### 8.3.1. Pseudo-code summary

```
    if ( I am not authoritative ) {
        send map-referral NOT_AUTHORITATIVE with
         incomplete bit set and ttl 0
    } else if ( delegation exists ) {
        if ( delegated map-servers ) {
            send map-referral MS_REFERRAL with
              ttl 'Default_DdtNode_Ttl'
        } else {
            send map-referral NODE_REFERRAL with
              ttl 'Default_DdtNode_Ttl'
        }
    } else {
        if ( eid in site) {
            if ( site registered ) {
                forward map-request to etr
                if ( map-server peers configured ) {
                    send map-referral MS_ACK with
                     ttl 'Default_Registered_Ttl'
                } else {
                    send map-referral MS_ACK with
                     ttl 'Default_Registered_Ttl' and incomplete bit set
                }
            } else {
                if ( map-server peers configured ) {
                    send map-referral MS_NOT_REGISTERED with
                     ttl 'Default_Configured_Not_Registered_Ttl'
                } else {
                    send map-referral MS_NOT_REGISTERED with
                     ttl 'Default_Configured_Not_Registered_Ttl'
                     and incomplete bit set
                }
            }
        } else {
            send map-referral DELEGATION_HOLE with
             ttl 'Default_Negative_Referral_Ttl'
        }
    }
```

where architectural constants for TTL are set as follows:

```
Default_DdtNode_Ttl                  1440 minutes
Default_Registered_Ttl               1440 minutes
Default_Negative_Referral_Ttl        15 minutes
Default_Configured_Not_Registered_Ttl 1 minute
```

### 8.3.2. Decision tree diagram

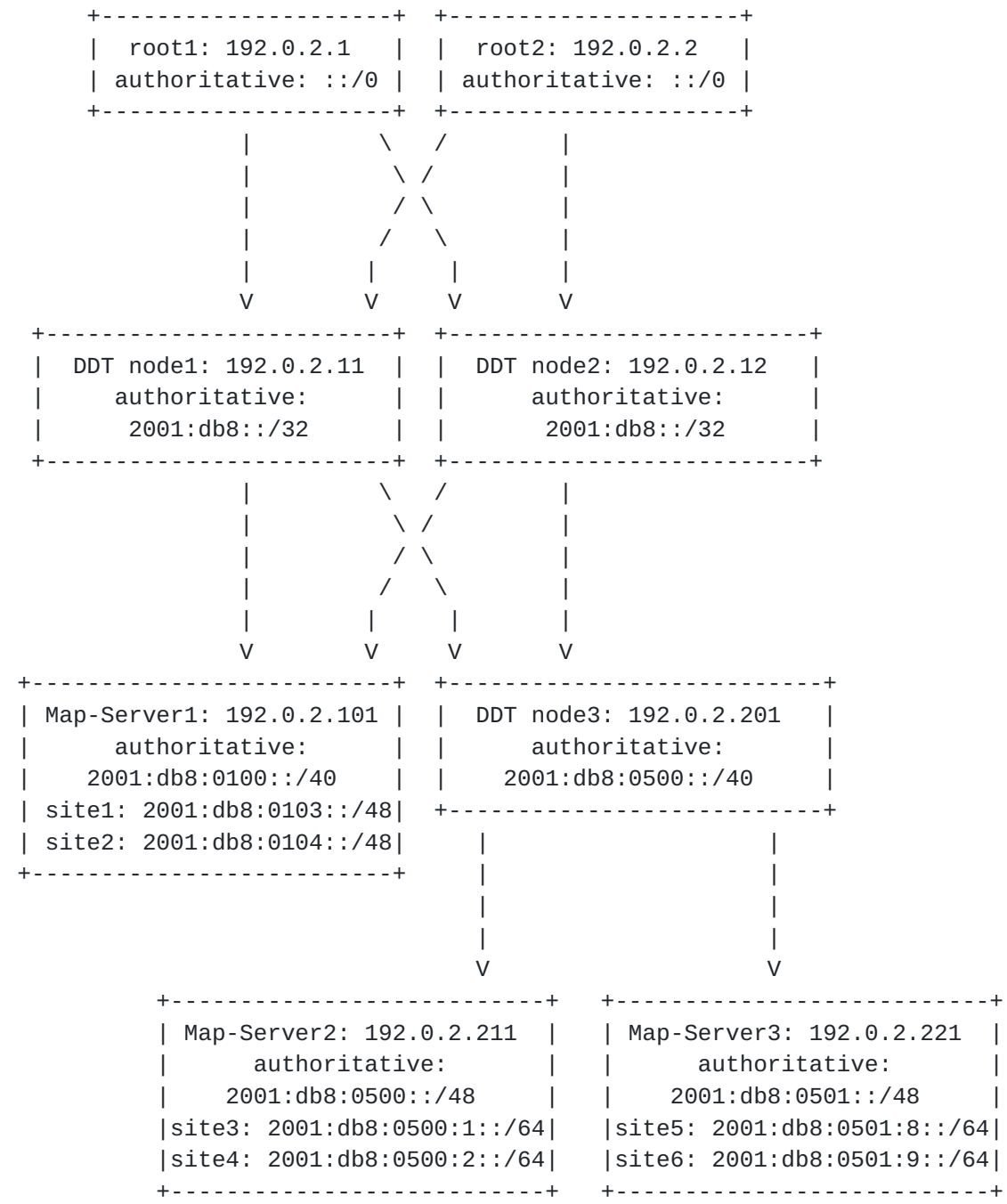



## 9. Example topology and request/referral following

This chapter shows example DDT tree and several possible scenarios of Map-Requests coming to a Map Resolver and subsequent iterative DDT referrals. For the sake of example RLOCs of DDT nodes are shown in

IPv4 address space while the EIDs are in IPv6 AF. The same principle of hierarchical delegation and pinpointing referrals is equally applicable to any AF whose address hierarchy can be expressed as a bitstring with associated length. DDT tree of IPv4 prefixes is another AF with immediate practical value.

To show how referrals are followed to find the RLOCs for a number of EIDs, consider the following example EID topology for DBID=0, IID=0, AFI=2, and EID=0/0



DDT nodes are configured for this "root" at IP addresses 192.0.2.1 and 192.0.2.2. DDT Map Resolvers are configured with default referral cache entries to these addresses.

The root DDT nodes delegate 2001:db8::/32 to two DDT nodes with IP addresses 192.0.2.11 and 192.0.2.12.

The DDT nodes for 2001:db8::/32 delegate 2001:db8:0100::/40 to a DDT Map Server with RLOC 192.0.2.101

The DDT Map Server for 2001:db8:0100::/40 is configured to allow ETRs to register the sub-prefixes 2001:db8:0103::/48 and 2001:db8:0104::/48

The DDT nodes for 2001:db8::/32 also delegate 2001:db8:0500::/40 to a DDT node with RLOC 192.0.2.201

The DDT node for 2001:db8:0500::/40 is further configured to delegate 2001:db8:0500::/48 to a DDT Map Server with RLOC 192.0.2.211 and 2001:db8:0501::/48 to a DDT Map Server with RLOC 192.0.2.221

The DDT Map Server for 2001:db8:0500::/48 is configured to allow ETRs to register the sub-prefixes 2001:db8:0500:1::/64 and 2001:db8:0500:2::/64

The DDT Map Server for 2001:db8:0501::/48 is configured to allow ETRs to register the sub-prefixes 2001:db8:0501:8::/64 and 2001:db8:0501:9::/64

### 9.1. Lookup of 2001:db8:0103:1::1/128

The first example shows a DDT Map Resolver following a delegation from the root to a DDT node followed by another delegation to a DDT Map Server.

ITR1 sends an Encapsulated Map-Request for 2001:db8:0103:1::1 to one of its configured (DDT) Map Resolvers. The DDT Map Resolver proceeds as follows:

1. Send DDT Map-Request (for 2001:db8:0103:1::1) to one of the root DDT nodes, 192.0.2.1 or 192.0.2.2

2. Receive (and save in referral cache) Map-Referral for EID-prefix 2001:db8::/32, action code NODE-REFERRAL, RLOC set (192.0.2.11, 192.0.2.12)

3. Send DDT Map-Request to 192.0.2.11 or 192.0.2.12

4. Receive (and save in referral cache) Map-Referral for EID-prefix 2001:db8:0100::/40, action code MS-REFERRAL, RLOC set (192.0.2.101)

5. Send DDT Map-Request to 192.0.2.101; if the ITR-originated Encapsulated Map-Request had a LISP-SEC signature, it is included

6. DDT Map Server at 192.0.2.101 decapsulates the DDT Map-Request and forwards to a registered site1 ETR for 2001:db8:0103::/48

7. DDT Map Server at 192.0.2.101 sends a Map-Referral message for EID-prefix 2001:db8:0103::/48, action code MS-ACK to the DDT Map Resolver

8. DDT Map Resolver receives Map-Referral message and dequeues the pending request for 2001:db8:0103:1::1

9. site1 ETR for 2001:db8:0103::/48 receives Map-Request forwarded by DDT Map Server and sends Map-Reply to ITR1

### 9.2. Lookup of 2001:db8:0501:8:4::1/128

The next example shows a three-level delegation: root to first DDT node, first DDT node to second DDT node, second DDT node to DDT Map Server.

ITR2 sends an Encapsulated Map-Request for 2001:db8:0501:8:4::1 to one of its configured (DDT) Map Resolvers, which are different from those for ITR1. The DDT Map Resolver proceeds as follows:

1. Send DDT Map-Request (for 2001:db8:0501:8:4::1) to one of the root DDT nodes, 192.0.2.1 or 192.0.2.2

2. Receive (and save in referral cache) Map-Referral for EID-prefix 2001:db8::/32, action code NODE-REFERRAL, RLOC set (192.0.2.11, 192.0.2.12)

3. Send DDT Map-Request to 192.0.2.11 or 192.0.2.12

4. Receive (and save in referral cache) Map-Referral for EID-prefix 2001:db8:0500::/40, action code NODE-REFERRAL, RLOC set (192.0.2.201)

5. Send DDT Map-Request to 192.0.2.201

6. Receive (and save in referral cache) Map-Referral for EID-prefix 2001:db8:0501::/48, action code MS-REFERRAL, RLOC set (192.0.2.221)

7. Send DDT Map-Request to 192.0.2.221; if the ITR-originated Encapsulated Map-Request had a LISP-SEC signature, it is included

8. DDT Map Server at 192.0.2.221 decapsulates the DDT Map-Request and forwards to a registered site5 ETR for 2001:db8:0501:8::/64

9. DDT Map Server at 192.0.2.221 sends a Map-Referral message for EID-prefix 2001:db8:0501:8::/64, action code MS-ACK, to the DDT Map Resolver

10. DDT Map Resolver receives Map-Referral(MS-ACK) message and dequeues the pending request for 2001:db8:0501:8:4::1

11. site5 ETR for 2001:db8:0501:8::/64 receives Map-Request forwarded by DDT Map Server and sends Map-Reply to ITR2

### 9.3. Lookup of 2001:db8:0104:2::2/128

This example shows how a DDT Map Resolver uses a saved referral cache entry to skip the referral process and go directly to a DDT Map Server for a prefix that is similar to one previously requested.

In this case, ITR1 uses the same Map Resolver used in example Section 9.1. It sends an Encapsulated Map-Request for 2001:db8:0104:2::2 to that (DDT) Map Resolver. The DDT Map-Resolver finds an MS-REFERRAL cache entry for 2001:db8:0100::/40 with RLOC set (192.0.2.101) and proceeds as follows:

1. Send DDT Map-Request (for 2001:db8:0104:2::2) to 192.0.2.101; if the ITR-originated Encapsulated Map-Request had a LISP-SEC signature, it is included

2. DDT Map Server at 192.0.2.101 decapsulates the DDT Map-Request and forwards to a registered site2 ETR for 2001:db8:0104::/48

3. DDT Map Server at 192.0.2.101 sends a Map-Referral message for EID-prefix 2001:db8:0104::/48, action code MS-ACK to the DDT Map Resolver

4. DDT Map Resolver receives Map-Referral(MS-ACK) and dequeues the pending request for 2001:db8:0104:2::2

5. site2 ETR for 2001:db8:0104::/48 receives Map-Request and sends Map-Reply to ITR1

### 9.4. Lookup of 2001:db8:0500:2:4::1/128

This example shows how a DDT Map Resolver uses a saved referral cache entry to start the referral process at a non-root, intermediate DDT node for a prefix that is similar to one previously requested.

In this case, ITR2 asks the same Map Resolver used in example Section 9.2. It sends an Encapsulated Map-Request for 2001:db8:0500:2:4::1 to that (DDT) Map Resolver, which finds a NODE-REFERRAL cache entry for 2001:db8:0500::/40 with RLOC set (192.0.2.201). It proceeds as follows:

1. Send DDT Map-Request (for 2001:db8:0500:2:4::1) to 192.0.2.201

2. Receive (and save in referral cache) Map-Referral for EID-prefix 2001:db8:0500::/48, action code MS-REFERRAL, RLOC set (192.0.2.211)

3. Send DDT Map-Request to 192.0.2.211; if the ITR-originated Encapsulated Map-Request had a LISP-SEC signature, it is included

4. DDT Map Server at 192.0.2.211 decapsulates the DDT Map-Request and forwards to a registered site4 ETR for 2001:db8:0500:2::/64

5. DDT Map Server at 192.0.2.211 sends a Map-Referral message for EID-prefix 2001:db8:0500:2::/64, action code MS-ACK to the DDT Map Resolver

6. DDT Map Resolver receives Map-Referral(MS-ACK) and dequeues the pending request for 2001:db8:0500:2:4::1

7. site4 ETR for 2001:db8:0500:2::/64 receives Map-Request and sends Map-Reply to ITR2

### 9.5. Lookup of 2001:db8:0500::1/128 (non-existent EID)

This example uses the cached MS-REFERRAL for 2001:db8:0500::/48 learned above to start the lookup process at the DDT Map-Server at 192.0.2.211. The DDT Map Resolver proceeds as follows:

1. Send DDT Map-Request (for 2001:db8:0500::1) to 192.0.2.211; if the ITR-originated Encapsulated Map-Request had a LISP-SEC signature, it is included

2. DDT Map Server at 192.0.2.211, which is authoritative for 2001:db8:0500::/48, does not have a matching delegation for 2001:db8:0500::1. It responds with a Map-Referral message for 2001:db8:0500::/64, action code DELEGATION-HOLE to the DDT Map

Resolver. The prefix 2001:db8:0500::/64 is used because it is the least-specific prefix that does match the requested EID, but does not match one of configured delegations (2001:db8:0500:1::/64 and 2001:db8:0500:2::/64).

3. DDT Map Resolver receives the delegation, adds a negative referral cache entry for 2001:db8:0500::/64, dequeues the pending request for 2001:db8:0500::1, and returns a negative Map-Reply to ITR2.

## 10. Securing the database and message exchanges

This section specifies the DDT security architecture that provides data origin authentication, data integrity protection, and XEID-prefix delegation. Global XEID-prefix authorization is out of the scope of this document.

Each DDT node is configured with one or more public/private key pair(s) that are used to digitally sign referral records for XEID-prefix(es) that the DDT node is authoritative for. In other words, each public/private key pair is associated with the combination of a DDT node and the XEID-prefix that it is authoritative for. Every DDT node is also configured with the public keys of its children DDT nodes. By including public keys of target child DDT nodes in the Map-Referral records, and signing each record with the DDT node's private key, a DDT node can securely delegate sub-prefixes of its authoritative XEID-prefixes to its children DDT nodes.

Map Resolvers are configured with one or more trusted public keys referred to as trust anchors. Trust anchors are used to authenticate the DDT security infrastructure. Map Resolvers can discover a DDT node's public key either by having it configured as a trust anchor, or by obtaining it from the node's parent as part of a signed Map-Referral. When a public key is obtained from a node's parent, it is considered trusted if it is signed by a trust anchor, or if it is signed by a key that was previously trusted. Typically, in a Map Resolver, the root DDT node public keys should be configured as trust anchors. Once a Map Resolver authenticates a public key it locally caches the key along with the associated DDT node RLOC and XEID-prefix for future use.

### 10.1. XEID-prefix Delegation

In order to delegate XEID sub-prefixes to its children, a parent DDT node signs its Map-Referrals. Every signed Map-Referral also includes the public keys associated with each child DDT node. Such a signature indicates that the parent node is delegating the specified XEID -prefix to a given child DDT node. The signature is also

authenticating the public keys associated with the children nodes, and authorizing them to be used by the children DDT nodes to provide origin authentication and integrity protection for further delegations and mapping information of the XEID-prefix allocated to the DDT node.

As a result, for a given XEID-prefix, a Map Resolver can form an authentication chain from a configured trust anchor (typically the root DDT node) to the leaf nodes (Map Servers). Map Resolvers leverage this authentication chain to verify the Map-Referral signatures while walking the DDT tree until they reach a Map Server authoritative for the given XEID-prefix.

### 10.2. DDT node operation

Upon receiving a Map-Request, the DDT node responds with a Map-Referral as specified in Section 7. For every record present in the Map-Referral, the DDT node also includes the public keys associated with the record's XEID-prefix and the RLOCs of the children DDT nodes. Each record contained in the Map-Referral is signed using the DDT node's private key.

#### 10.2.1. DDT public key revocation

The node that owns a public key can also revoke that public key. For instance if a parent node advertises a public key for one of its child DDT nodes, the child DDT node can at a later time revoke that key. Since DDT nodes do not keep track of the Map Resolvers that query them, revocation is done in a pull model, where the Map Resolver is informed of the revocation of a key only when it queries the node that owns that key. If the parent DDT is configured to advertise this key, the parent node must also be signaled to remove the key from the records it advertises for the child DDT node; this is necessary to avoid further distribution of the revoked key.

To securely revoke a key, the DDT node creates a new Record for the associated XEID-prefix and locator, including the revoked key with the R bit set. The DDT node must also include a signature in the Record that covers this record; this is computed using the private key corresponding to the key being revoked. Such a record is termed a "revocation record". By including this record in its Map-Referrals, the DDT node informs querying Map Resolvers about the revoked key. A digital signature computed with a revoked key can only be used to authenticate the revocation, and SHOULD NOT be used to validate any data. To prevent a compromised key from revoking other valid keys, a given key can only be used to sign a revocation for that specific key; it cannot be used to revoke other keys. This prevents the use of a compromised key to revoke other valid keys as

described in [RFC5011]. A revocation record must be advertised for a period of time equal to or greater than the TTL value of the Record that initially advertised the key, starting from the time that the advertisement of the key was stopped by removal from the parent DDT node.

### 10.3. Map Server operation

Similar to a DDT node, a Map Server is configured with one (or more) public/private key pairs that it must use to sign Map-Referrals.

However unlike DDT nodes, Map Servers do not delegate prefixes and as a result they do not need to include keys in the Map-Referrals they generate.

### 10.4. Map Resolver operation

Upon receiving a Map-Referral, the Map Resolver must first verify the signature(s) by using a trust anchor, or a previously authenticated public key, associated with the DDT node sending the Map-Referral. If multiple authenticated keys are associated with the DDT node sending this Map-Referral, the Key Tag field of the signature can be used to select the right public key for verifying the signature. If the key tag matches more than one key associated with that DDT node, the Map Resolver must try verifying the signature with all matching keys. For every matching key that is found the Map Resolver must also verify that the key is authoritative for the XEID-prefix in the Map-Referral record. If such a key is found, the Map Resolver must use it to verify the associated signature in the record. If no matching key is found, or if none of the matching keys is authoritative for the XEID-prefix in the Map-Referral record, or if such a key is found but the signature is not valid the Map-Referral record is considered corrupted and must be discarded. This may be due to expired keys. The Map Resolver can try other siblings of this node if there is an alternative node authoritative for the same prefix. If not, the Map Resolver can query the DDT node's parent to retrieve a valid key. It is good practice to use a counter or timer to avoid repeating this process if the resolver cannot verify the signature after several trials.

Once the signature is verified, the Map Resolver has verified the XEID-prefix delegation in the Map-Referral, and authenticated the public keys of the children DDT nodes. The Map Resolver must add these keys to the authenticated keys associated with each child DDT node and XEID-prefix. These keys are considered valid for the duration specified in the record's TTL field.

## 11. Open Issues and Considerations

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

- Defining an interface to implement interconnection and/or interoperability with other mapping databases, such as LISP+ALT.
- Additional key structures for use with LISP-DDT, such as to support additional EID formats as defined in [I-D.ietf-lisp-lcaf]
- Management of the DDT Map Resolver referral cache, in particular, detecting and removing outdated entries.

Operational experience will help answer open questions surrounding these and other issues.

## 12. IANA Considerations

This document makes no request of the IANA.

## 13. Security Considerations

Section 10 describes a DDT security architecture that provides data origin authentication, data integrity protection, and XEID-prefix delegation within the DDT Infrastructure.

Global XEID-prefix authorization is beyond the scope of this document, but the SIDR working group [RFC6480] is developing an infrastructure to support improved security of Internet routing. Further work is required to determine if SIDR's public key infrastructure (PKI) and the distributed repository system it uses for storing and disseminating PKI data objects may also be used by DDT devices to verifiably assert that they are the legitimate holders of a set of XEID prefixes.

This document specifies how DDT security and LISP-SEC ([I-D.ietf-lisp-sec]) complement one another to secure the DDT infrastructure, Map-Referral messages, and the Map-Request/Map-Reply protocols. In the future other LISP security mechanisms may be developed to replace LISP-SEC. Such future security mechanisms should describe how they can be used together with DDT to provide similar levels of protection.

LISP-SEC can use the DDT public key infrastructure to secure the transport of LISP-SEC key material (the One-Time Key) from a Map-Resolver to the corresponding Map-Server. For this reason, when LISP-SEC is deployed in conjunction with a LISP-DDT mapping database

and the path between Map-Resolver and Map-Server needs to be protected, DDT security should be enabled as well.

## 14. References

### 14.1. 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>.

[RFC6833] Fuller, V. and D. Farinacci, "Locator/ID Separation Protocol (LISP) Map-Server Interface", RFC 6833, DOI 10.17487/RFC6833, January 2013, <http://www.rfc-editor.org/info/rfc6833>.

### 14.2. Informative References

[I-D.ietf-lisp-lcaf] Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format (LCAF)", draft-ietf-lisp-lcaf-12 (work in progress), March 2016.

[I-D.ietf-lisp-sec] Maino, F., Ermagan, V., Cabellos-Aparicio, A., and D. Saucez, "LISP-Security (LISP-SEC)", draft-ietf-lisp-sec-09 (work in progress), October 2015.

[LISP-TREE] Jakab, L., Cabellos-Aparicio, A., Coras, F., Saucez, D., and O. Bonaventure, "LISP-TREE: a DNS hierarchy to support the lisp mapping system", Selected Areas in Communications, IEEE Journal , 2010, <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5586446>.

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <http://www.rfc-editor.org/info/rfc1918>.

[RFC3447] Jonsson, J. and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, DOI 10.17487/RFC3447, February 2003, <http://www.rfc-editor.org/info/rfc3447>.

[RFC5011] StJohns, M., "Automated Updates of DNS Security (DNSSEC) Trust Anchors", STD 74, RFC 5011, DOI 10.17487/RFC5011, September 2007, <http://www.rfc-editor.org/info/rfc5011>.

[RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, DOI 10.17487/RFC6480, February 2012, <http://www.rfc-editor.org/info/rfc6480>.

[RFC6836] Fuller, V., Farinacci, D., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol Alternative Logical Topology (LISP+ALT)", RFC 6836, DOI 10.17487/RFC6836, January 2013, <http://www.rfc-editor.org/info/rfc6836>.

## Appendix A. Acknowledgments

The authors would like to express their thanks to Lorand Jakab, Albert Cabellos-Asparicio, Florin Coras, Damien Saucez, and Olivier Bonaventure for their work on LISP-TREE [LISP-TREE] and LISP iterable mappings that inspired the hierarchical database structure and lookup iteration approach described in this document. Thanks also go to Dino Farinacci and Isidor Kouvelas for their implementation work; to Selina Heimlich and Srin Subramanian for testing; to Fabio Maino for work on security processing; and to Job Snijders, Glen Wiley, Neel Goyal, and Mike Gibbs for work on operational considerations and initial deployment of a prototype database infrastructure. Special thanks go to Jesper Skriver, Andrew Partan, and Noel Chiappa; all of whom have participated in (and put up with) seemingly endless hours of discussion of mapping database ideas, concepts, and issues.

Authors' Addresses

Vince Fuller

Email: vaf@vaf.net

Darrel Lewis
Cisco Systems

Email: darlewis@cisco.com

Vina Ermagan
Cisco Systems

Email: vermagan@cisco.com

Amit Jain
Juniper Networks

Email: atjain@juniper.net

Anton Smirnov
Cisco Systems

Email: as@cisco.com