LISP Working Group
Internet-Draft
Obsoletes: 8111 (if approved)
Updates: 9301 (if approved)
Intended status: Standards Track
Expires: 4 September 2025

D. Saucez, Ed.
Inria
L. Iannone, Ed.
Huawei

3 March 2025

Locator/ID Separation Protocol Delegated Database Tree (LISP-DDT)
draft-saucez-lisp-8111bis-01

Abstract

This document describes the Locator/ID Separation Protocol Delegated Database Tree (LISP-DDT), a hierarchical distributed database that 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 control plane elements generically 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.

This document obsoletes RFC 8111 and updates RFC 9301.

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 https://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 4 September 2025.

Copyright Notice

Table of Contents

1. Introduction

The Locator/ID Separation Protocol (LISP), defined in [RFC9300] and [RFC9301], specifies an architecture to create overlays networks leveraging on two separate namespaces, namely the Endpoint Identifiers (EIDs) used for end-to-end communications, and the Routing Locators (RLOCs) used for routing and forwarding.

[RFC9301] specifies an interface between a database storing EID-to-RLOC mappings and LISP devices that need such information to forward packets. The internal organization of such a database is beyond the scope of [RFC9301]. Multiple architectures of the database have been proposed, each having its advantages and disadvantages (see, for example, [RFC6836] and [RFC6837]).

This document specifies an architecture for a scalable distributed database of LISP EID-to-RLOC mappings: the LISP Delegated Database Tree (LISP-DDT). LISP-DDT is a hierarchical distributed database that 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 perform EID-to-RLOC mappings lookups. A Map-Resolver that requests a given mapping will follow a path through the tree-structured database and will contact, one after another, the nodes along that path, until it reaches the leaf node(s) authoritative for the mapping it is seeking.

In organizing a database of EID-to-RLOC mappings, this specification extends the definition of the EID numbering space by logically concatenating the following attributes in order to define the database index key:

* Database-ID (DBID) (16 bits)

* Instance Identifier (IID) (24 bits)

* Address Family Identifier (AFI) (16 bits)

* EID-Prefix (variable, according to the AFI value)

The resulting concatenation of these fields is termed an "Extended EID-Prefix", or XEID-Prefix.

LISP-DDT defines a new device type, the "DDT Node", that is configured as authoritative for one or more XEID-Prefixes. It is also 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 DDT Map-Server (MS) 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 [RFC9301] 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 the Map-Referral LISP message type, 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 5 for the definition of the Map-Referral message.

To find an EID-to-RLOC mapping, a LISP-DDT Client, usually a Map-Resolver, starts by sending an Encapsulated Map-Request to a pre-configured DDT Node RLOC. The DDT Node responds with a Map-Referral message indicating that either (1) it will find the requested mapping to complete processing of the request or (2) the DDT Client should contact another DDT Node that has more-specific information; in the latter case, the DDT Node then sends a new 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 [RFC1035].

This document obsoletes [RFC8111] and updates [RFC9301].

2. Requirements Language

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

3. Definitions of Terms

This documents assumes that the reader is familiar with LISP and the LISP terminology. For definitions of terms like Map-Request, Encapsulated Map-Request, Map-Reply, ITR, ETR, Map-Server, and Map-Resolver, please consult the LISP Data Plane specification [RFC9300] and the LISP Control Plane specification [RFC9301].

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.

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 (as defined by [RFC9301]), but it is also possible for an Ingress Tunnel Router (ITR) to implement DDT Client functionality.

DDT 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 DDT Map-Referral includes a "referral" consisting in a set of RLOCs for DDT Nodes that have information about the more-specific XEID-Prefix covering the requested XEID. See Section 5 for a complete definition of the message and Section 6.1 and Section 6.3 for details on its processing.

DDT Map-Referral Cache: Data structure to temporarily maintain previously received Map-Referral message results, containing RLOCs for DDT Nodes responsible for XEID-Prefixes.

DDT Map-Request: an ECM Map-Request sent by a DDT Client to a DDT Node, with the "DDT-originated" flag set. Section 6.3.1 describes how DDT Map-Requests are sent. [RFC9301] defines the position of the "DDT-originated" flag in the Encapsulated Control Message header.

DDT Map-Resolver: a Map-Resolver that also implements DDT Client

functionality. Map-Resolver functionality is defined by [RFC9301]. A DDT Map-Resolver accepts Map-Requests from ITRs, sends Map-Requests to DDT Nodes, and implements the iterative following of Map-Referrals. Note that Map-Resolvers, as of [RFC9301], do not respond to clients that sent Map-Requests; they only ensure that the Map-Request has been forwarded to a LISP device (ETR or proxy Map-Server) that will provide an authoritative response to the original requester. This document uses the terms "DDT Map-Resolver" and "Map-Resolver" interchangeably.

DDT Map-Server: a Map-Server that also implements DDT functionality. Map-Server functionality is defined in [RFC9301]. This document uses the terms "DDT Map-Server" and "Map-Server" interchangeably.

DDT Map-Server peers: a list of all DDT Map-Servers performing Map-Server functionality for the same prefix. If peers are configured on a DDT Map-Server, then the latter will provide complete information about the prefix in its Map-Replies; otherwise, the Map-Server will mark the returned reply as potentially incomplete.

DDT Node: a network infrastructure component responsible for specific XEID-Prefix(es) and for the delegation of more-specific sub-prefixes to other DDT Nodes.

Extended EID (XEID): a LISP EID extended with data uniquely identifying the address space to which it belongs (LISP IID, address family, etc.). See Section 4.1 for a detailed description of XEID data.

Extended EID-Prefix (XEID-Prefix): a LISP EID-Prefix prepended with XEID data. An XEID-Prefix is used as a key index into the DDT database. XEID-Prefixes are used to describe database organization and are not seen as a single entity in protocol messages, though messages contain individual fields constituting XEID-Prefixes.

Negative DDT Map-Referral: A Negative Map-Referral is 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 Requests List: Data structure storing the set of outstanding requests for which a DDT Map-Resolver has received ECM Map-Requests from its clients seeking EID-to-RLOC mapping for an XEID. Each entry in the list contains additional state needed by the referral-following process, including the XEID, requester(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 the last RLOC queried in the RLOC set), and any LISP-Security (LISP-SEC) information [RFC9303] that was included in the DDT Client Map-Request. An entry in the list may be interchangeably termed a "pending requests list entry" or simply a "pending request".

4. Delegated Database Tree Organization

This section firstly defines the DDT database index key, namely XEID-Prefixes, and then details of the DDT database organization and how to configure prefix delegation.

4.1. XEID-Prefixes

A DDT database is indexed by Extended EID-Prefixes (XEID-Prefixes). An XEID-Prefix is a LISP EID-Prefix that includes additional data to uniquely identify the address space associated with the prefix. An XEID-Prefix is composed of four binary-encoded concatenated attributes:

* DBID (16 bits),

* Instance ID (24 bits),

* AFI (16 bits),

* and EID-Prefix (variable, according to the AFI value).

The DBID is the LISP-DDT Database-ID, a 16-bit attribute that allows the definition of multiple databases. Implementations that are compliant with this document must always set this field to 0. Other values of the DBID are reserved for future use.

The Instance ID (IID) is a 24-bit value describing the context of the EID-Prefix, see Section 8 of [RFC9300] for more discussion of Instance IDs.

The AFI is a 16-bit value defining the syntax of the EID-Prefix. AFI values are assigned by IANA [AFI].

4.2. Structure of the DDT Database

The LISP-DDT Database is organized as a prefix tree structure that is indexed by XEID-Prefixes, where each node of the tree is a DDT Node which has delegated for a set of sub-prefixes (see Section 4.3 for details regarding delegation).

DDT Map-Requests sent by DDT Clients to DDT Nodes always contain specific values for the DBID, IID, and AFI; unspecified values or ranges of values MUST NOT be used for any of these attributes.

LISP-DDT does not store actual EID-to-RLOC mappings; it is, rather, a distributed index that can be used to find the devices (ETRs that registered their EIDs with DDT Map-Servers) that can be queried with LISP to obtain those mappings. Changes to EID-to-RLOC mappings are made on the ETRs that 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.

4.3. 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. A delegation 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 regarding security information exchange and its use, see Section 7). Those RLOCs are returned in Map-Referral messages when the DDT Node receives a DDT Map-Request with an XEID that matches a delegation.

DDT Nodes may also have a list of "hints", which are XEID-Prefixes, for which it is no authoritative. Each of such XEID-Prefixes "hints" includes the list of RLOCs of the DDT Nodes authoritative for the XEID-Prefix. The list of "hints" is statically configured similarly to delegations.

4.3.1. The Root DDT Node

The root DDT Node is the logical "top" of the distributed database hierarchy. It is authoritative for all XEID-Prefixes, namely, for all valid 3-tuples (DBID, IID, AFI) and their EID-Prefixes.

The root DDT Nodes is the starting point for a XEID-Prefix lookup via a DDT Map-Request. However, not all DDT Map-Request will go through a root DDT Nodes thanks to the DDT Map-Referral Cache maintained by intermediate nodes (see Section 6 for details). The root DDT Node in a particular instantiation of LISP-DDT MUST be configured, at a minimum, to cover all EID space of all <DBID, Instance IDs, AFI> tuples defined in the instantiation.

5. The DDT Map-Referral Message

This specification defines a new LISP message called the DDT Map-Referral message. A DDT Map-Referral message is sent by a DDT Node to a DDT Client in response to a DDT Map-Request message. The message consists of an action code along with delegation information about the XEID-Prefix that matches the requested XEID.

5.1. Map-Referral Message Format

The format of the Map-Referral message is depicted in Figure 1.

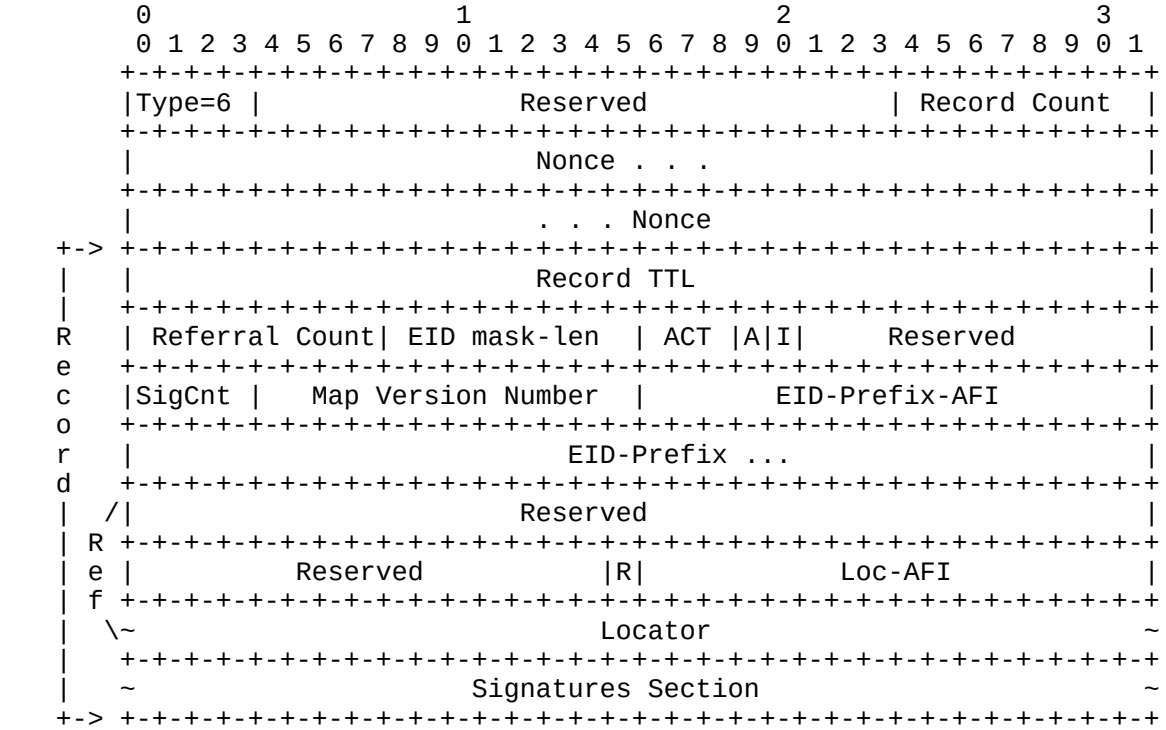


Figure 1: Map-Referral message format.

Type: DDT Map-Referral type 6, as allocated in [RFC9301].

Record Count: As defined in Section 5.4 of [RFC9301].

Nonce: As defined in Section 5.4 of [RFC9301].

Record TTL: As defined in Section 5.4 of [RFC9301].

Referral Count: This is the number of referral records in this message. A referral record is comprised of that portion of the packet labeled "Ref" and occurs the number of times equal to Referral Count.

EID mask-len: As defined in Section 5.4 of [RFC9301].

ACT: The DDT Action (ACT) field of the in a Map-Referral message encodes one of the action types defined in Section 5.2. Note that, despite the same position and same name as in Map-Reply messages defined in [RFC9301], the actions in DDT Map-Referrals message are different from the ones in Map-Replies.

A: The Authoritative bit CAN only be set to 1 by a DDT Node that is authoritative for the XEID-Prefix.

I; Incomplete (I) bit indicating that a DDT Node’s Referral Set of locators is incomplete and the receiver of this message SHOULD NOT cache the referral (see Section 5.2 for details).

SigCnt: Indicates the number of signatures sections present in the Record. If SigCnt is larger than 0, the signature information captured in a Signature section as described in Section 5.5 will be appended to the Record. The number of Signature sections at the end of the Record MUST match the SigCnt.

Map Version Number: As defined in [RFC9302].

EID-Prefix-AFI: As defined in Section 5.4 of [RFC9301].

XEID-Prefix: The requested XEID-Prefix. Uses the LCAF type 2 [RFC8060] to encode the IID and IP address prefix that form the XEID. It can also use natively IPv4 or IPv6 addresses, in this case the IID has the implicit default value of 0. The EID-Prefix-AFI MUST be set accordingly.

R bit: As defined in Section 5.4 of [RFC9301].

Loc-AFI: AFI of the Locator field. Values for this field include the value 16387 of the LISP Canonical Address Format (LCAF) [RFC8060]. LCAF Security Key Type 11 is used to store security material associated to the locator. DDT Nodes and DDT Map-Servers uses this LCAF Type to include public keys associated with their child DDT Nodes for an XEID-Prefix Map-Referral Record.

Locator: RLOC of a DDT Node to which the DDT Client is being referred. This is a variable-length field; its length is determined by the Loc-AFI setting.

Signatures Section: When present this section contain one (or more) signature sections containing a signature covering the whole DDT Map-Referral message and the related information necessary for its verification. See Section 5.5 for the details.

Fields marked as "reserved" MUST be sent as 0 (zero) and MUST be ignored on reception.

5.2. Action Codes

The possible values of the action field are defined hereafter, where the number in parenthesis indicated the decimal value of the 3-bit ACT filed.

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 record with additional information, most significantly, the set of RLOCs to which the XEID-Prefix has been delegated, which is used by a DDT Client 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 6.3.2).

MS-ACK (2): Indicates that the 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 a Map-Reply 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 that 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 6.1.2 for details. <!-- Also sent by a DDT Map-Server with authoritative configuration covering the requested EID but for which no specific site ETR is configured. LI: Not coherent. In this case an MS-Not-REGISTERED MUST be sent--.

NOT-AUTHORITATIVE (5): Indicates that the replying DDT Node received

a Map-Request for an XEID for which it is not authoritative. This can occur if a cached referral has become invalid due to a change in the database hierarchy. However, if such a DDT Node has a "hint" covering the requested EID, it MAY choose to return NODE-REFERRAL or MS-REFERRAL as appropriate. When returning action code NOT-AUTHORITATIVE, the DDT Node MUST provide the EID-Prefix received in the request and the TTL MUST be set to 0.

5.3. Referral Set

For "positive" action codes (NODE-REFERRAL, MS-REFERRAL, MS-ACK), a DDT Node MUST include in the Map-Referral message a list of RLOCs for DDT Nodes that are authoritative for the XEID-Prefix being returned. A DDT Client uses this information to contact one of those DDT Nodes as it "follows" a referral.

5.4. "Incomplete" Flag

A DDT Node sets the "Incomplete" (I) flag in a Map-Referral message if the Referral Set is incomplete; this is intended to prevent a DDT Client from caching, in the DDT Map-Referral Cache, a referral with incomplete information. A DDT node MUST set the "Incomplete" flag in the following cases:

* If returned action code is MS-ACK or MS-NOT-REGISTERED, but the matching XEID-Prefix is not flagged as "complete" in the configuration DDT Map-Server. The XEID-Prefix configuration on the DDT Map-Server SHOULD be marked as "complete" when the configuration of the XEID-Prefix includes all DDT Map-Server peers that are also authoritative for the same XEID-Prefix, or when a DDT Map-Server is the only authoritative node for the XEID-Prefix .

* If the returned action code is NOT-AUTHORITATIVE.

5.5. Signature Section

SigCnt counts the number of signature sections that appear at the end of the Record. The format of the signature section is described Figure 2.

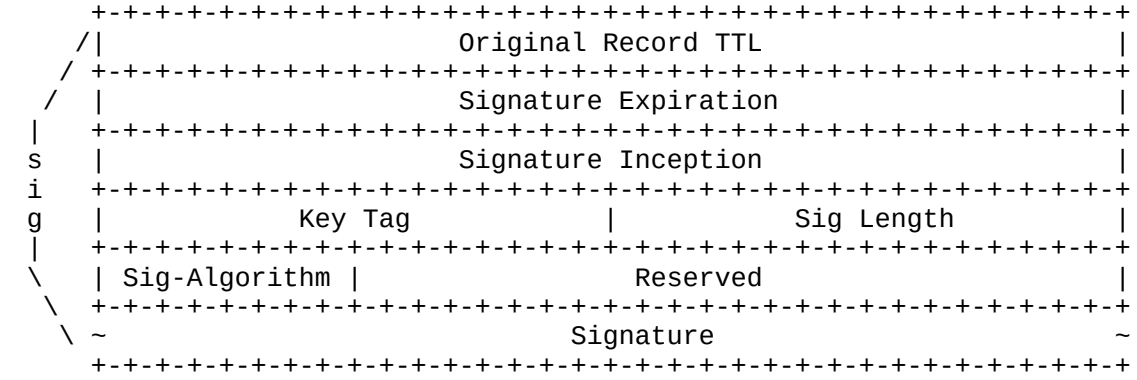


Figure 2: Map-Referral Signature Section format.

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

Signature Expiration and Signature Inception: Specify the validity period for the signature. The signature MUST NOT be used for authentication prior to the inception and MUST NOT be used for authentication after the expiration. 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 (see Section 8).

Sig Length: The length of the Signature field in bytes.

Sig-Algorithm: The identifier of the cryptographic algorithm used for the signature. Sig-Algorithm values defined in this specification are listed in Table 1. Implementations conforming to this specification MUST implement at least RSA-SHA256 for DDT signing. Sig-Algorithm type 1 (RSA-SHA1) is deprecated and SHOULD NOT be used.

Sig-Algorithm	Name	Reference	Notes
1	RSA-SHA1	[RFC8017]	DEPRECATED
2	RSA-SHA256	[RFC8017]	MANDATORY

Table 1: Sig-Algorithm Values

Reserved: MUST be set to 0 on transmit and MUST be ignored on receipt.

Signature: Contains the cryptographic signature that covers the entire Map-Referral Record to which this signature belongs. For the purpose of computing the signature, the Record TTL (Section 5.1) value is set to the value of Original Record TTL and the Signature field is filled with zeros.

6. DDT Network Elements and Their Operation

As described above, LISP-DDT introduces a new network element, namely the DDT Node and extends the functionality of Map-Servers and Map-Resolvers to send and receive DDT Map-Referral messages. The operation of each of these devices is described below.

6.1. 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 depending on whether or not there is a match, different actions are taken, as described in the following.

6.1.1. XEID Match

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 7 for details on security). If at least one DDT Node of the delegation is known to be a DDT Map-Server, then the Map-Referral message SHOULD be 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 SHOULD be used.

A DDT Node MAY also be configured with "hints" for XEID-Prefixes anywhere in the database hierarchy and for which it can provide referrals. This feature may be useful for reducing the number of iterations needed to find an EID-to-RLOC Mapping, particularly for private network deployments. However, the incorrect use of hints may create circular dependencies (or "referral loops") between DDT Nodes. A DDT Client MUST handle such circular referrals as described in Section 6.3.4. Furthermore, the use of hints in DDT deployments that span multiple administrative domains (i.e., different authorities manage DDT Nodes in the same DDT database) may not be a reliable information. Indeed, in this case, an operator managing a DDT Node may not be aware of the fact that the node is being referred to by hints. Locator addresses in hints may become stale when referred DDT Nodes are taken out of service or change their locator addresses.

6.1.2. XEID Miss

If the requested XEID did not match a configured delegation but does match an authoritative XEID-Prefix, then the DDT Node MUST return 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, which indicates that the XEID is not a LISP destination.

If the requested XEID did not match either a configured delegation, an authoritative XEID-Prefix, or a hint, then a Negative Map-Referral with action code NOT-AUTHORITATIVE MUST be returned.

6.2. 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 ETR RLOCs associated to the XEID-Prefix (or the Map-Server sends a Map-Reply, if it is providing a proxy Map-Reply service), and a Map-Referral with action code MS-ACK MUST be 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 MUST be returned to the sender of the DDT Map-Request. This Map-Reply contains the least-specific XEID-Prefix in the range for which this DDT Map-Server is authoritative and in which no registrations exist. In order to avoid long periods of lack of connectivity, the TTL value of the Negative Map-Reply should be short. The RECOMMENDED value is of 1 minute.

6.3. DDT Client

A DDT Client queries one or more DDT Nodes and uses an iterative process of following returned referrals until it receives one with action code MS-ACK (or an error indication). MS-ACK indicates that the Map-Request has reached a Map-Server that will forward it to an ETR that, in turn, will provide a Map-Reply to the locator address in the Map-Request.

DDT Client functionality is usually performed by DDT Map-Resolvers. Just as would 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. However, a DDT Map-Resolver implements DDT Client functionality to find the correct ETR to answer a Map-Request, which in turns requires a DDT Map-Resolver to maintain additional state, namely a DDT Map-Referral Cache and a Pending Requests List of XEIDs that are going through the iterative referral process.

6.3.1. Queuing and Sending DDT Map-Requests

When a DDT Client receives a A DDT Map-Request to resolve an XEID, it first performs a longest-prefix-match search for the XEID in its DDT Map-Referral Cache. Note that in normal use, the statically configured initial DDT Map-Referral Cache for a DDT Client should include a "default" entry with RLOCs for either the root DDT Node or one or more DDT Nodes that contain hints for the root DDT Node. Without such configuration, a DDT client would return a Negative Map-Reply for Map-Requests for an EID outside the subset of the mapping database known to it. There is no need to configure the same set of root DDT nodes on all DDT Clients. Additional entries are added when referrals are followed, as described below. If the longest-prefix-math for the XEID in the Map-Request does not return any result, or if a negative entry is found, then a Negative Map-Reply MUST be returned, and no further processing is performed by the DDT Client.

If a match is found, the DDT Client creates an entry in the Pending Requests LIst for the XEID, storing the original request (in the case of a DDT Map-Resolver, this will be the original Map-Request minus the encapsulation header) along with other information needed to track progress through the iterative referral process; the referral associated to the request is also initialized to indicate that no referral has yet been received.

The DDT Client then creates a DDT Map-Request, which is an ECM 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 request. It sends the DDT Map-Request to one of the RLOCs in the selected DDT Map-Referral Cache entry.

If DDT message exchanges are authenticated as described in Section 7, then the DDT Client MUST also be configured with public keys of DDT Nodes pointed to by the "default" cache entry. In this case, the "default" entry will typically be for the root DDT Node.

6.3.2. Receiving and Following DDT Map-Referrals

After sending a DDT Map-Request, a DDT Client expects to receive a DDT Map-Referral response. If none occurs within the timeout period, the DDT Client retransmits the request, sending it to the next RLOC in the referral cache entry if one is available. If all RLOCs have been tried and the maximum number of retransmissions has occurred for each, then the Pending Request List entry is discarded. In this case, the DDT Client returns no response to the sender of the original request.

When a DDT Client receives a DDT Map-Referral, it processes the message according to the action code it contains:

NODE-REFERRAL: The DDT Client checks for a possible referral loop as described in Section 6.3.4. If no loop is found, the DDT Client saves the returned XEID-Prefix in the DDT Map-Referral Cache. It also updates the XEID-Prefix and saved RLOCs in the corresponding 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 SHOULD NOT be included.

MS-REFERRAL: The DDT Client processes an MS-REFERRAL in the same manner as a NODE-REFERRAL, except that LISP-SEC security information saved with the original Map-Request MUST be included in the new DDT Map-Request sent to a Map-Server (see Section 7 for details on security).

MS-ACK: An MS-ACK is returned by a DDT Map-Server to indicate that it has one or more registered ETRs that can answer the DDT Map-Request for the XEID and the message has been forwarded to one of them (or, if the Map-Server is providing a proxy service for the prefix, then a reply has been sent to the querying ITR). If the pending request did not include saved LISP-SEC information (i.e., LISP-SEC is not used) or if that information was already included in the previous DDT Map-Request (sent by the DDT Client in response to either an MS-REFERRAL or a previous MS-ACK referral), then the pending request for the XEID is complete, processing of the request stops, and all request state can be discarded. Otherwise, when LISP-SEC information included in pending request, but has not yet been sent to the authoritative DDT Map-Server, the DDT Client MUST resend the DDT Map-Request with LISP-SEC information included, and the Pending Request List entry remains until another Map-Referral with action code MS-ACK is received. Unless the "Incomplete" flag is set, the XEID-Prefix and its referrals is saved in the DDT Map-Referral Cache.

MS-NOT-REGISTERED: The DDT Map-Server queried could not process the request because no ETRs registered the XEID-Prefix and returned a Negative DDT Map-Referral. If the DDT Client has not yet tried all of the RLOCs saved with the Pending Request List, 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 Client MUST return the Negative Map-Reply to the requester (or, in the case of a DDT Map-Resolver, to the sender of the original Map-Request). A negative DDT Map-Referral Cache entry is also created for the XEID-Prefix, whose TTL MUST be set to the value in the Map-Referral message, and the processing of the request stops.

DELEGATION-HOLE: The requested XEID-Prefix does not belong to the EID space and the queried DDT Node returned a Negative DDT Map-Referral. The DDT Client MUST return a Negative Map-Referral to the requester (or, in the case of a DDT Map-Resolver, a Negative Map-Reply to the sender of the original Map-Request);

NOT-AUTHORITATIVE: The DDT Node queried is not authoritative for the

requested XEID-Prefix. This can occur, for instance, if a cached referral has become invalid due to a change in the database hierarchy. If the DDT Client receiving this message used a cached information, it MAY choose to delete that cached information and retry the original Map-Request, starting from its "root" DDT Map-Referral Cache entry. If this action code is received in response to a query that did not use cached referral information, then it indicates a database synchronization problem or configuration error. In this case the DDT Client SHOULD log the event and the pending request MUST be silently discarded.

A DDT Client is not required to cache referrals, but doing so will decrease latency and reduce lookup delays.

6.3.3. Handling Referral Errors

If a DDT Client detects unexpected behavior by a DDT node, it MAY mark that node as unusable in its DDT Map-Referral Cache and update the pending request to try a different DDT Node, if more than one is listed. In any case, the DDT Client SHOULD log the error and any prefix contained in a DDT Map-Referral message that causes a referral error (including a referral loop) is not saved in the DDT Map-Referral Cache.

6.3.4. Referral Loop Detection

In response to a Map-Referral message with action code NODE-REFERRAL or MS-REFERRAL, a DDT Client 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 Client saves the last received referral XEID-Prefix for each pending request and checks to see if 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 6.3.3.

7. 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.

Each DDT Node is configured with one or more public/private key pairs that are used to digitally sign Map-Referral Records for XEID-Prefix(es) for which the DDT Node is authoritative. In other words,

each public/private key pair is associated with the combination of a DDT Node and an XEID-Prefix for which it is authoritative. Every DDT Node is also configured with the public keys of its child DDT Nodes. By including the 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 refer sub-prefixes its authoritative XEID-Prefixes for its child DDT nodes. A DDT Node configured to provide hints must also have the public keys of the DDT Nodes to which its hints point. DDT Node keys can be encoded using LCAF Type 11 to associate the key to the RLOC of the referred DDT Node. If a node has more than one public key, it should sign its Records with at least one of these keys. The revocation mechanism is described in Section 7.2.1.

Map-Resolvers are configured with one or more DDT Nodes' 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 by either (1) having it configured as a trust anchor or (2) 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's 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.

7.1. XEID-Prefix Delegation

In order to provide referrals to XEID sub-prefixes for its child DDT Nodes, a parent DDT Node signs its DDT Map-Referrals. Every signed Map-Referral MUST also include the public keys associated with each child DDT node. Such a signature indicates that the parent DDT Node delegated the specified XEID-Prefix to a given child DDT Node. The signature is also authenticating the public keys associated with the child DDT Nodes, and authorizing them to be used by the child DDT nodes, to provide origin authentication and integrity protection 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.

7.2. DDT Node Operation

Upon receiving a Map-Request, the DDT Node responds with a Map-Referral as specified in Section 6. 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 child DDT Nodes. Each Record contained in the Map-Referral is signed using the DDT Node's private key.

7.2.1. DDT Public Key Revocation

The node that owns a public key can also revoke that public key. For instance, if a parent DDT 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 mode, 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 Node is configured to advertise that key, the parent DDT 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. (See Section 4.7 of [RFC8060] for details regarding the R bit.) 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 MUST 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 MUST NOT 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.

7.3. 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 do not need to include keys in the Map-Referrals they generate.

7.4. Map-Resolver Operation

Upon receiving a Map-Referral, the Map-Resolver MUST first verify the signature(s) by using either 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 (Section 5.5) of the signature can be used to select the correct 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 to verify the signature with all matching keys. If a key is found, the Map-Resolver MUST use it to verify the associated signature in the Record. If (1) no matching key is found, or (2) 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 MAY try other siblings of this node if there is an alternate node that is authoritative for the same prefix. If not, the Map-Resolver CAN query the DDT Node’s parent to retrieve a valid key.

Once the signature is verified, the Map-Resolver has verified the XEID-Prefix delegation in the Map-Referral. This also means that public keys of the child DDT Nodes were authenticated; 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.

8. Security Considerations

This documents extends the LISP control plane defined in [RFC9301]; as such security considerations in that document apply here as well.

This document specifies how DDT security and LISP-SEC [RFC9303] complement one another to secure the DDT infrastructure, Map-Referral messages, and the Map-Request/Map-Reply protocols.

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,

where the path between the Map-Resolver and Map-Server needs to be protected. Such protection can be achieved with the mechanism described in Section 7, which proposes a DDT security architecture that provides data origin authentication, data integrity protection, and XEID-Prefix delegation within the DDT infrastructure. DDT security as described in Section 7 MUST be enabled by default.

9. Deployment Experience

TBD

10. IANA Considerations

10.1. LISP DDT Map-Referral Packet Type

IANA has made the an early temporary allocation for message type 6, "LISP DDT Map-Referral", in the "LISP Packet Types" registry group. IANA is requested to make the allocation permanent and modify the "LISP Packet Types" as shown in Table 2.

Code	Message	Reference
6	LISP DDT Map-Referral	[This document]

Table 2: LISP DDT Map-Referral Message Type Allocation

10.2. LISP DDT Action Codes

IANA is asked to create a registry named "LISP DDT Action Codes" under the "Locator/ID Separation Protocol (LISP) Parameters" group. Such registry should be populated as shown in Table 3:

Code	DDT Action	Reference
0	NODE-REFERRAL	[This document]
1	MS-REFERRAL	[This document]
2	MS-ACK	[This document]
3	MS-NOT-REGISTERED	[This document]
4	DELEGATION-HOLE	[This document]
5	NOT-AUTHORITATIVE	[This document]
6-7	Unallocated	[This document]

Table 3: LISP DDT Action Codes Allocation

Values can be assigned by IANA on the "IETF Review" basis according to [RFC8126].

11. References

11.1. Normative References

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

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

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

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

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/rfc/rfc8126>.

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

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

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

[RFC9302] Iannone, L., Saucez, D., and O. Bonaventure, "Locator/ID Separation Protocol (LISP) Map-Versioning", RFC 9302, DOI 10.17487/RFC9302, October 2022, <https://www.rfc-editor.org/rfc/rfc9302>.

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

11.2. Informative References

[AFI] IANA, "Address Family Numbers", <http://www.iana.org/assignments/address-family-numbers/>.

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

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

[RFC6837] Lear, E., "NERD: A Not-so-novel Endpoint ID (EID) to Routing Locator (RLOC) Database", RFC 6837, DOI 10.17487/RFC6837, January 2013, <https://www.rfc-editor.org/rfc/rfc6837>.

[RFC8017] Moriarty, K., Ed., Kaliski, B., Jonsson, J., and A. Rusch, "PKCS #1: RSA Cryptography Specifications Version 2.2", RFC 8017, DOI 10.17487/RFC8017, November 2016, <https://www.rfc-editor.org/rfc/rfc8017>.

Appendix A. Pseudo-code and Decision Tree Diagrams

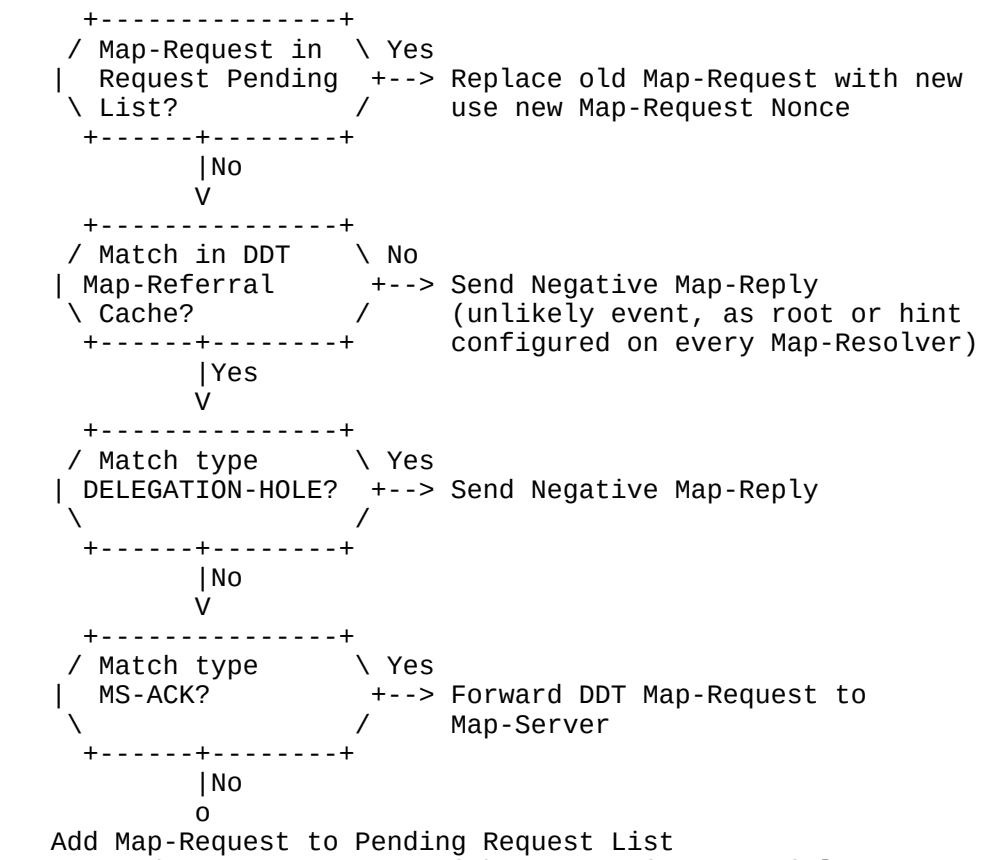
To illustrate the DDT algorithms described in this document and to aid in implementation, each of the major DDT Map-Server and DDT Map-Resolver functions are described first using simple "pseudo-code" and then in the form of a decision tree.

A.1. Map-Resolver Processing of Map-Request

A.1.1. Pseudo-code Summary

```
if (Map-Request already in Pending Request List) {
   replace old Map-Request with new
   use new Map-Request nonce

} else if ( no match in DDT Map-Referral Cache ) {
   return Negative Map-Reply to requester

} else if ( match type DELEGATION-HOLE ) {
   return Negative Map-Reply to requester

} else if ( match type MS-ACK ) {
   forward DDT Map-Request to Map-Server

} else {
   Add Map-Request to Pending Request List
   Forward DDT Map-Request w/o security material

}
```

A.1.2. Decision Tree Diagram



A.2. Map-Resolver Processing of Map-Referral Message

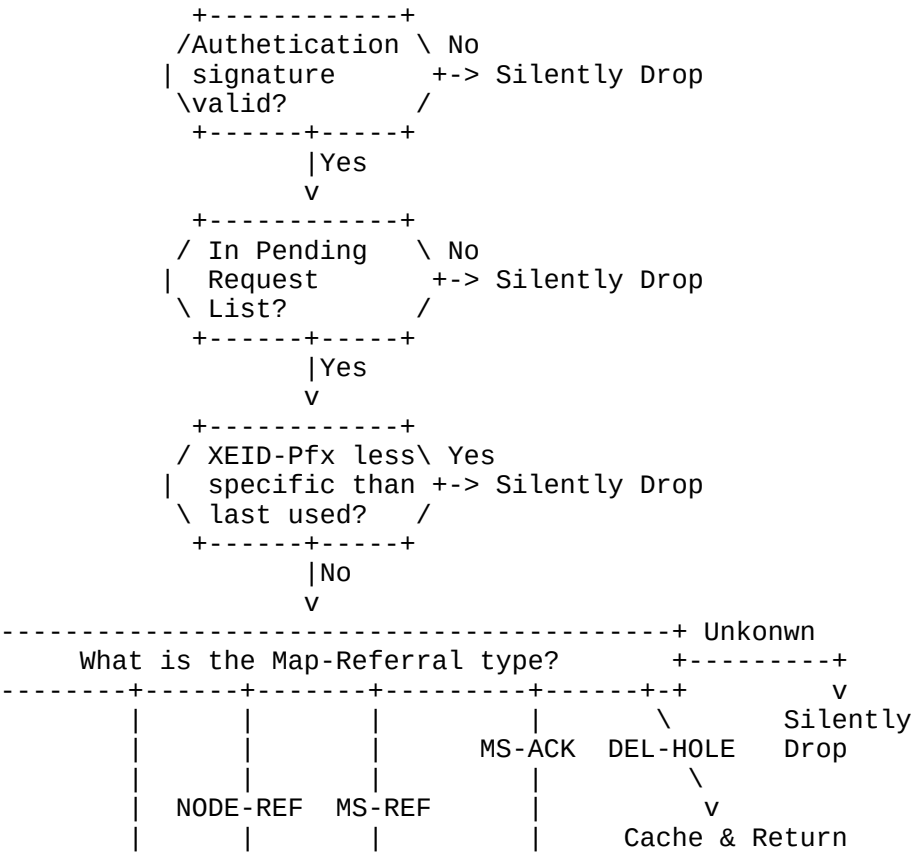
A.2.1. Pseudo-code Summary

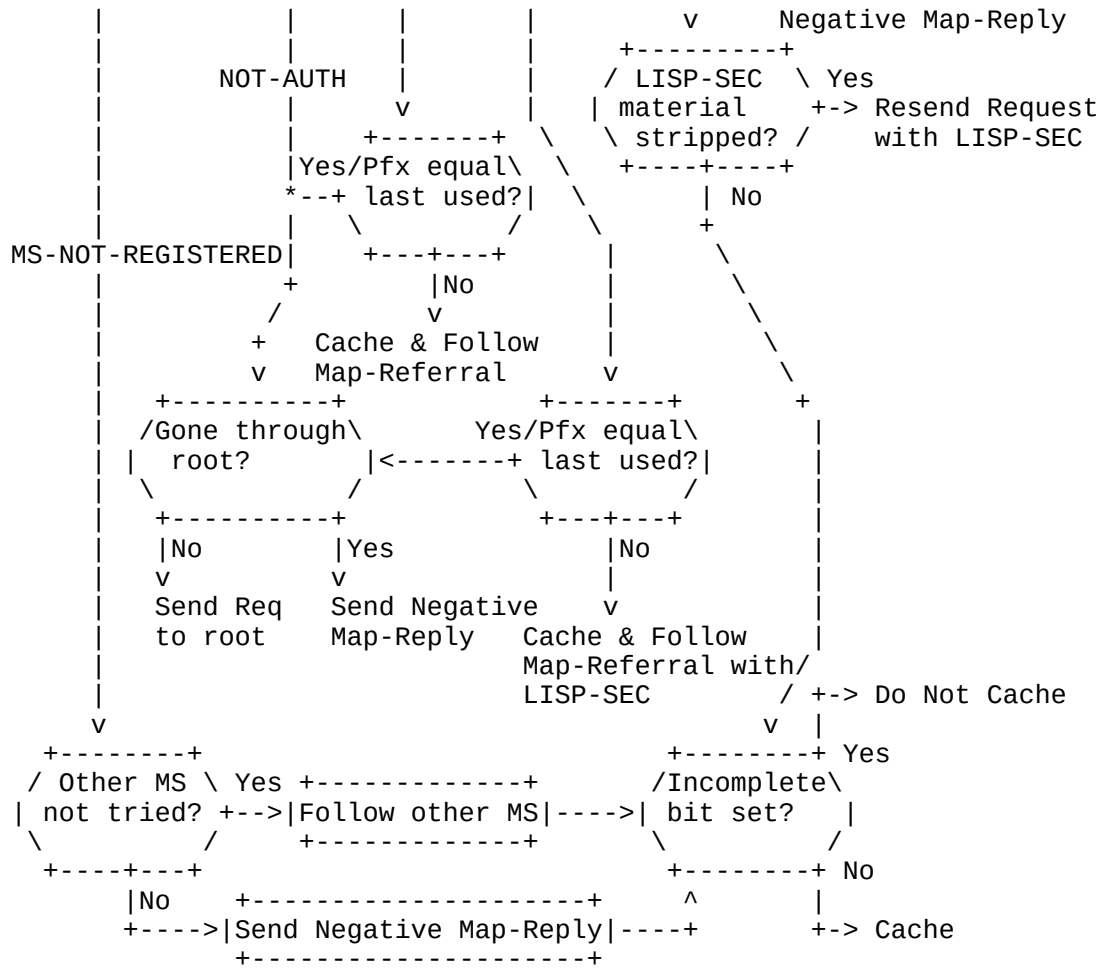
```
if ( authentication signature validation failed ) {
   silently drop
}

if ( no Pending Request List entry matched by Map-Referral Nonce ) {
   silently drop
}

if ( XEID-Prefix in Map-Referral less specific than last used ) {
   if ( gone through root ) {
      silently drop
   } else {
      send Map-Request to root
```

```
      }
   }

   switch (map_referral_type) {

      case NOT-AUTHORITATIVE:
         if ( gone through root ) {
            return Negative Map-Reply to requester
         } else {
            send Map-Request to root
         }

      case DELEGATION-HOLE:
         Store in DDT Map-Referral Cache
         send Negative Map-Reply to requester

      case MS-REFERRAL:
         if ( Map-Referral RLOC Set contains last used RLOC ) {
            if ( gone through root ) {
               return Negative Map-Reply to requester
            } else {
               send Map-Request to root
            }
         } else {
            Store in DDT Map-Referral Cache
            follow Map-Referral with LISP-SEC security material
         }

      case NODE-REFERRAL:
         if ( Map-Referral RLOCs Set contains last used RLOC ) {
            if ( gone through root ) {
               return Negative Map-Reply to requester
            } else {
               send Map-Request to root
            }
         } else {
            Store in DDT Map-Referral Cache
            follow Map-Referral without LISP-SEC security material
         }

      case MS-ACK:
         if ( Map-Referral has no LISP-SEC security material ) {
            resend Map-Request with LISP-SEC security material
            if { !incomplete } {
               Store in DDT Map-Referral Cache
            }
         }
```

```
      case MS-NOT-REGISTERED:
         if ( all Map-Server delegations not tried ) {
            follow Map-Referral without LISP-SEC security material
            if ( !incomplete ) {
               Store in DDT Map-Referral Cache
            }
         } else {
            send Negative Map-Reply to requester
            if { !incomplete } {
               Store in DDT Map-Referral Cache
            }
         }

      case DEFAULT:
         Silently Drop
   }
```

A.2.2. Decision Tree Diagram



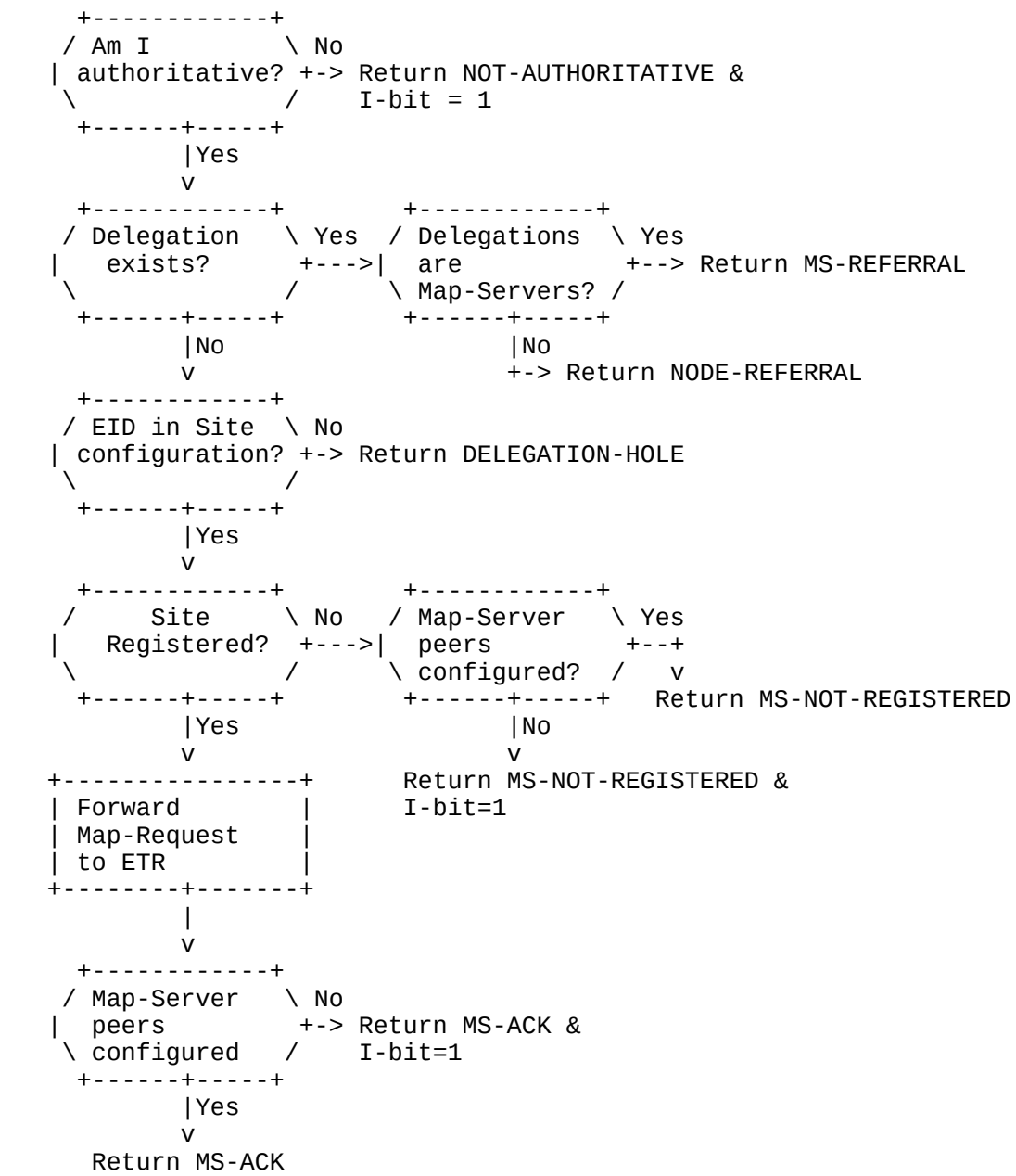


A.3. DDT Node Processing of DDT Map-Request Message

A.3.1. Pseudo-code Summary

```
   if ( I am not authoritative ) {
      send Map-Referral NOT_AUTHORITATIVE & set "Incomplete" bit

   } else if ( delegation exists ) {
      if ( delegated Map-Servers ) {
         send Map-Referral MS_REFERRAL

      } else {
         send Map-Referral NODE_REFERRAL
      }

   } else {
      if ( EID in site ) {
         if ( site registered ) {
            forward Map-Request to ETR

            if ( Map-Server peers configured ) {
               send Map-Referral MS_ACK

            } else {
               send Map-Referral MS_ACK & set"Incomplete" bit
            }

         } else {
            if ( Map-Server peers configured ) {
               send Map-Referral MS_NOT_REGISTERED

            } else {
               send Map-Referral MS_NOT_REGISTERED & set"Incomplete" bit
            }

         }

      } else {
         send Map-Referral DELEGATION_HOLE

      }

   }
```

A.3.2. Decision Tree Diagram


Am I authoritative?
No
Return NOT-AUTHORITATIVE & I-bit = 1
Yes
Delegation exists?
Yes
Delegations are Map-Servers?
Yes
Return MS-REFERRAL
No
No
Return NODE-REFERRAL
EID in Site configuration?
No
Return DELEGATION-HOLE
Yes
Site Registered?
No
Map-Server peers configured?
Yes
Return MS-NOT-REGISTERED
No
Forward Map-Request to ETR
Return MS-NOT-REGISTERED & I-bit=1
Map-Server peers configured?
No
Return MS-ACK & I-bit=1
Yes
Return MS-ACK

Appendix B. Generic DDT Example

This section shows an example of DDT tree and several possible scenarios of Map-Requests coming to a Map-Resolver and subsequent iterative DDT referrals. In this example, RLOCs of DDT Nodes are shown in the IPv4 address space while the EIDs are in the IPv6 address space.

B.1. Reference DDT Tree Topology

To show how referrals are followed to find the RLOCs for a number of different requests, the DDT topology in Figure 3 is used.

Figure 3: Reference DDT tree topology.

DDT Root nodes are configured with IP addresses 192.0.2.1 and 192.0.2.2. DDT Map-Resolvers are configured with default referral cache entries for these addresses.

The DDT Root nodes delegate 2001:db8::/32 to two DDT nodes, DDT-Node1 and DDT-Node2, having IP addresses 192.0.2.11 and 192.0.2.12 respectively.

DDT-Node1 and DDT-Node2 delegate 2001:db8:0100::/40 to a Map-Server1, who has RLOC 192.0.2.101. Map-Server1 is configured to allow ETRs to register the sub-prefixes 2001:db8:0103::/48 and 2001:db8:0104::/48.

DDT-Node1 and DDT-Node2 also delegate 2001:db8:0500::/40 to DDT-Node3 who uses the address 192.0.2.201. DDT-Node3 further delegates 2001:db8:0500::/48 to Map-Server2, who has RLOC 192.0.2.211, and 2001:db8:0501::/48 to Map-Server3, who has RLOC 192.0.2.221.

Map-Server2 is configured to allow ETRs to register the sub-prefixes 2001:db8:0500:1::/64 and 2001:db8:0500:2::/64.

Map-Server3 is configured to allow ETRs to register the sub-prefixes 2001:db8:0501:8::/64 and 2001:db8:0501:9::/64.

B.2. Lookup EID registered at with Map-Server1

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. The example assumes that DDT Map-Referral Caches contain only the default configuration.

ITR1 sends an ECM 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 a 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 the DDT Map-Referral Cache) the Map-Referral for EID-Prefix 2001:db8::/32, action code NODE-REFERRAL, RLOC set (192.0.2.11, 192.0.2.12).

3. Send a DDT Map-Request to 192.0.2.11 or 192.0.2.12.

4. Receive (and save in the DDT Map-Referral Cache) the Map-Referral for EID-Prefix 2001:db8:0100::/40, action code MS-REFERRAL, RLOC set (192.0.2.101).

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

6. The DDT Map-Server1 (192.0.2.101) decapsulates the DDT Map-Request and forwards the Map-Request to the registered Site1 ETR for 2001:db8:0103::/48.

7. The DDT Map-Server1 (192.0.2.101) sends a DDT Map-Referral message for EID-Prefix 2001:db8:0103::/48, action code MS-ACK, to the DDT Map-Resolver.

8. The DDT Map-Resolver receives the Map-Referral message with action code MS-ACK and removes the request for 2001:db8:0103:1::1 from the Pending Request List.

9. Site1's ETR for 2001:db8:0103::/48 receives the Map-Request forwarded by Map-Server1 and sends a Map-Reply to ITR1.

B.3. Lookup EID registered at with Map-Server3

This example shows a three-level delegation: root to first DDT node, first DDT Node to second DDT Node, and second DDT Node to DDT Map-Server. The example assumes that DDT Map-Referral Caches contain only the default configuration.

ITR2 sends an ECM 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 a 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 the DDT Map-Referral Cache) the Map-Referral for EID-Prefix 2001:db8::/32, action code NODE-REFERRAL, RLOC set (192.0.2.11, 192.0.2.12).

3. Send a DDT Map-Request to 192.0.2.11 or 192.0.2.12.

4. Receive (and save in the DDT Map-Referral Cache) the DDT Map-Referral for EID-Prefix 2001:db8:0500::/40, action code NODE-REFERRAL, RLOC set (192.0.2.201).

5. Send a DDT Map-Request to 192.0.2.201.

6. Receive (and save in the DDT Map-Referral Cache) the DDT Map-Referral for EID-Prefix 2001:db8:0501::/48, action code MS-REFERRAL, RLOC set (192.0.2.221).

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

8. Map-Server3 (192.0.2.221) decapsulates the DDT Map-Request and forwards the Map-Request to a registered Site5 ETR for 2001:db8:0501:8::/64.

9. Map-Server3 (192.0.2.221) sends a DDT Map-Referral message for EID-Prefix 2001:db8:0501:8::/64, action code MS-ACK, to the DDT Map-Resolver.

10. The DDT Map-Resolver receives a Map-Referral message with action code MS-ACK and removes the request for 2001:db8:0501:8:4::1 from the Pending Request List.

11. Site5's ETR for 2001:db8:0501:8::/64 receives the Map-Request forwarded by Map-Server3 and sends a Map-Reply to ITR2.

B.4. Lookups using cached DDT Map-Referrals to Map-Servers

This example shows a lookup for 2001:db8:0104:2::2, where the DDT Map-Resolver uses a saved DDT Map-Referral Cache entry to skip the iterative referral process and go directly to a DDT Map-Server.

In this case, ITR1 uses the same Map-Resolver used in the example in Appendix B.2. It sends an ECM 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 directly a DDT Map-Request (for 2001:db8:0104:2::2) to 192.0.2.101; if the ITR-originated ECM Map-Request had a LISP-SEC signature, it is included.

2. Map-Server1 (192.0.2.101) decapsulates the DDT Map-Request and forwards the Map-Request to the registered Site2 ETR for 2001:db8:0104::/48.

3. Map-Server1 (192.0.2.101) sends a DDT Map-Referral message for EID-Prefix 2001:db8:0104::/48, action code MS-ACK, to the DDT Map-Resolver.

4. The DDT Map-Resolver receives the DTT Map-Referral with action code MS-ACK and removes the request for 2001:db8:0104:2::2 from the Pending Request List.

5. Site2's ETR for 2001:db8:0104::/48 receives the Map-Request from Map-Server1 and sends a Map-Reply to ITR1.

B.5. Lookup using cached DDT Map-Referrals to DDT Nodes

This example shows how a DDT Map-Resolver uses a DDT Map-Referral Cache entry to start the referral process at a non-root, intermediate DDT node for the prefix 2001:db8:0500:2:4::1, which is similar to one previously requested.

In this case, ITR2 uses the same Map-Resolver used in the example in Appendix B.3. It sends an ECM 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 a DDT Map-Request (for 2001:db8:0500:2:4::1) to 192.0.2.201.

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

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

4. Map-Server2 (192.0.2.211) decapsulates the DDT Map-Request and forwards the Map-Request to the registered Site4 ETR for 2001:db8:0500:2::/64.

5. Map-Server3 (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. The DDT Map-Resolver receives the Map-Referral (MS-ACK) and removes the request for 2001:db8:0500:2:4::1 from the Pending Request List.

7. Site4’s ETR for 2001:db8:0500:2::/64 receives the Map-Request and sends a Map-Reply to ITR2.

B.6. Lookup of a non-existent EID

This example shows the lookup of 2001:db8:0500::1/128, which uses the cached MS-REFERRAL for 2001:db8:0500::/48 learned because of previous lookups at the DDT Map-Server at 192.0.2.211. The DDT Map-Resolver proceeds as follows:

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

2. Map-Server2 (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 DDT 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 the configured delegations (2001:db8:0500:1::/64 and 2001:db8:0500:2::/64).

3. The DDT Map-Resolver receives the delegation, adds a Negative DDT Map-Referral Cache entry for 2001:db8:0500::/64, removes the request for 2001:db8:0500::1 from the Pending Request list, and returns a Negative Map-Reply to ITR2.

Contributors

Vince Fuller
VAF.NET Internet Consulting
Email: vince.fuller@gmail.com

Darrel Lewis
ICANN
Los Angeles, CA 90292
United States of America
Email: darrel.lewis@icann.org

Vina Ermagan
Google
United States of America
Email: ermagan@gmail.com

Amit Jain
Juniper Networks
Email: atjain@juniper.net

Anton Smirnov
Cisco Systems, Inc.
De Kleetlaan 6a
1831 Diegem
Belgium
Email: as@cisco.com

Authors' Addresses

Damien Saucez (editor)
Inria
France
Email: damien.saucez@inria.fr

Luigi Iannone (editor)
Huawei Technologies France S.A.S.U.
18, Quai du Point du Jour
92100 Boulogne-Billancourt
France
Email: luigi.iannone@huawei.com