Network Working Group
Internet-Draft
Intended status: Experimental
Expires: September 13, 2012

G. Wiley, Ed.
Verisign, Inc.

March 12, 2012

LISP-DDT Database Transfer
draft-wiley-lisp-ddtxfer-01

Abstract

This draft describes a protocol for transferring a LISP Delegated Database Tree (LISP-DDT) between DDT nodes and for notifying DDT nodes that the database has changed. In the absence of a formally defined protocol the LISP-DDT database is transferred using files containing device configuration commands. This protocol provides for transferring a complete database or the deltas between two versions of the database. This document does not describe the use of DDT as part of the LISP mapping service.

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 September 13, 2012.

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.

1. Introduction

References to the LISP-DDT database in this document are intended to refer specifically to the portion of the LISP-DDT database that has been delegated to the DDT node, not the entire database.

An operator that offers a LISP-DDT service faces a few problems (similar to those faces by the operator of a DNS resolution service):

- How to replicate the LISP-DDT database from one host/device to another to support redundant platforms or to support operational maintenance.
- Transfering the LISP-DDT between dissimilar platforms, for example between network devices offered by competing network device vendors that rely on different configuration commands.
- How to transfer portions of the LISP-DDT database rather than the entire database (to deal with the eventual large database).
- How to notify an interested DDT node that the LISP-DDT database has changed

The focus for this document is to define a protocol for the transfer of the LISP-DDT database between DDT XFR nodes that addresses these problems. The local representation of the data in the LISP-DDT database on the DDT node is outside the scope of this document.

In the absence of a well described line protocol an operator might send a list of OS commands to run on the DDT node that would populate the LISP DDT database, however even this approach requires an operational protocol to ensure a reliable/repeatable transfer of the LISP DDT database.

There are two types of transfers, both are limited to only the portion of the database that has been delegated to the DDT node. A full transfer sends the entire set of entries in the database for which the DDT node is authoritative. An incremental transfer sends only the portion of that set of entries in the database that has changed (based on the version of the database on the receiving node).

LISP-DDT bears some resemblance to the Domain Name Service (DNS) in the sense that it provides an indirect vector to the target data.

Think of a LISP-DDT query as the analog to a DNS name server (NS) query, and a LISP map request as the analog to a DNS address (A) query (LISP-DDT does not store the EID to RLOC mappings returned in a map request).

DDT XFR clients that need to be notified of changes to the DDT database may subscribe to the appropriate DDT XFR server and are asynchronously notified of changes. A DDT XFR client always initiates a transfer by sending a request to the authoritative DDT XFR server.

LISP-DDT database transfer introduces the use of a monotonically increasing 64-bit serial number that identifies a version of a portion of a LISP-DDT database. Each delegated portion of the database (identified by a unique Extended EID-prefix) has a distinct serial number which is maintained by the authoritative DDT XFR server. One or more changes to the contents of the LISP-DDT database increments the serial number. Once a serial number is used it may never be reused. This feature ensures that a DDT XFR client can reliably determine whether its copy of the LISP-DDT database is consistent with the one offered by the DDT XFR server. This serial number is analogous to the DNS SOA serial number.

1.1. LISP-DDT

LISP-DDT defines a database key for identifying EID mappings to RLOCs in the EID numbering space. This key (the Extended EID prefix) is comprised of the following fields:

Field	Size
Key-ID	16 bits
Instance Identifier (IID)	32 bits
Address Family Identifier (AFI)	16 bits
EID-prefix	variable length per AFI

See [LISP-DDT] for a more complete discussion of the key fields.

The root DDT XFR server is the apex of the hierarchy and is identified by the key: Key-ID=0, IID=0, AFI=0, EID-prefix=0/0

It is useful to note that a DDT node does not necessarily imply a single host, in fact it is more likely implemented as a collection of hosts operating behind a VIP or via some other mechanism that allows for active redundant components.

1.2. Network Transport

TCP is used exclusively as the transport for LISP-DDT transfer sessions as opposed to other LISP messages which are sent via UDP.

1.3. Terminology

Note that for the sake of consistency many of these definitions were lifted directly from [LISP-DDT].

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

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

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

Authoritative DDT node: The DDT map server that is authoritative for a portion of the LISP-DDT database is identified as an "authoritative DDT node/server" and is the exclusive source for an authoritative copy of the portion of the LISP-DDT database that has been delegated to that DDT map server.

DDT XFR client: The DDT node that is requesting a LISP-DDT database transfer from a DDT XFER server. A DDT XFR client would likely be a device/host that may serve as a DDT server following the transfer.

DDT XFR server: The DDT node that receives a request for a LISP-DDT database transfer and functions as the source of the LISP-DDT database is the DDT XFR server. DDT XFR servers never initiate a transfer, however they may send notification messages to DDT XFR clients that have subscribed to the portion of the database for which the DDT server is authoritative.

Full Transfer: A full LISP-DDT database transfer (FDDTXFR) describes the transfer of the entire portion of the LISP-DDT database for which a DDT-node is authoritative. The transfer does not include portions of the database that are delegated to other DDT nodes.

Incremental Transfer: An incremental LISP-DDT database transfer (IDDTXFR) is a transfer of the differences between two serial numbers of the LISP-DDT database for which a DDT XFR server is authoritative. The current serial number and the serial number provided by the DDT XFR client determine the context of the differences.

serial number: A monotonically increasing number that indicates a specific version of the portion of the LISP-DDT database identified by a unique Extended EID-prefix. There may be more than one material difference between versions of the database even though the serial number is incremented once.

session: A LISP-DDT database transfer session ("session") includes all communication between the DDT XFR client and DDT XFR server involved in requesting and responding to a request for the transfer of a portion of the LISP-DDT database.

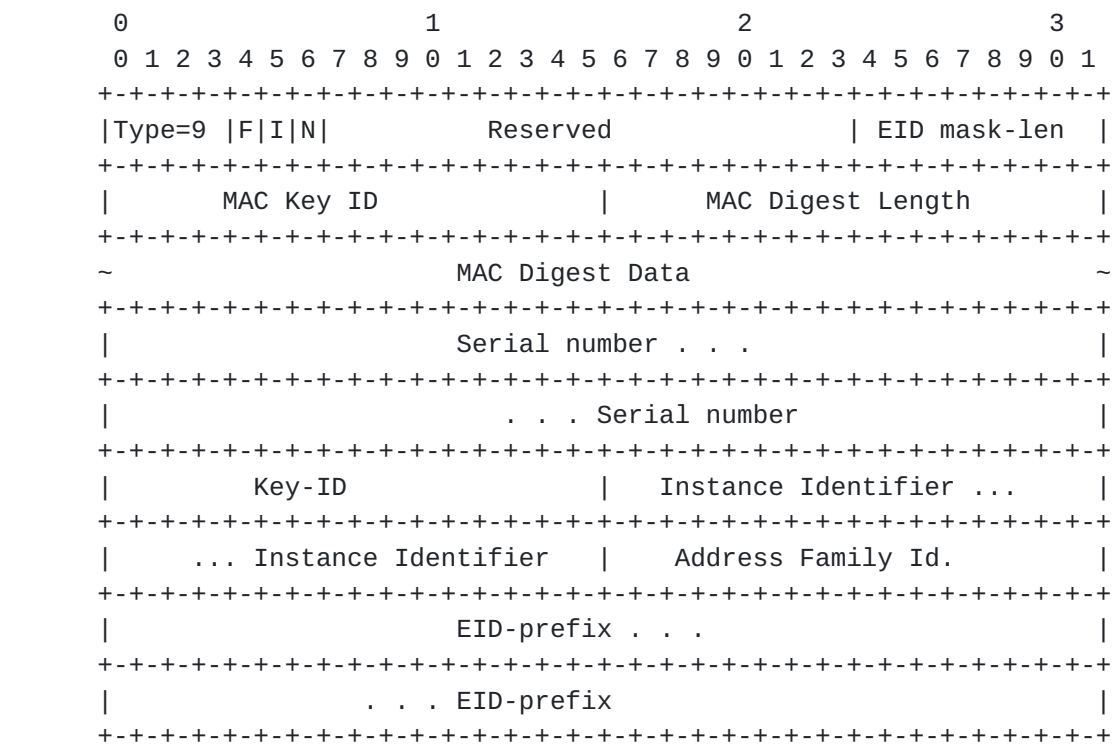
For definitions of other acronyms (EID, RLOC, etc.) see [LISP] and [LISP-DDT].

1.4. 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 RFC 2119 [RFC2119].

2. LISP-DDT Database Transfer Request

Immediately after the network connection is established from a DDT XFR client to a DDT XFR server the DDT XFR client sends a transfer request message. The LISP-DDT transfer message is very similar to the LISP Map-Request.



LISP-DDT Transfer Request Message Format

Packet field descriptions:

Type=9: DDT Transfer request, this four bit value identifies the packet type (other LISP packet types are defined in the respective Internet Drafts).

F (bit flag): If this bit is set then this is a full transfer request.

I (bit flag): If this bit is set then this is an incremental transfer request.

N (bit flag): If this bit is set then this is a notification that the serial number for the portion of the database identified by the Extended EID has changed.

EID mask-len: The EID mask length (in bits) of the EID-prefix, not including the three fields that prefix the EID-prefix.

MAC Key ID: Identifies the Message Authentication Code (MAC) digest algorithm per [LISP].

MAC Digest length: Length in bytes of the following MAC digest data.

MAC Digest Data: The MAC digest data depends on the MAC Key ID specified and is calculated on all of the fields that follow in this message.

Serial number: In the case of an incremental transfer the DDT XFR client populates this 64-bit value with the current serial number so that the DDT XFR server can attempt to determine which database changes must be sent. This field is ignored for full transfers.

Key-ID: 16 bit Key-ID for the X-EID prefix being requested.

Instance Identifier: 32 bit Instance Identifier for the X-EID prefix being requested.

AFI: 16 bit Address Family Identifier for the X-EID prefix being requested.

EID-prefix: The variable length EID-prefix (per AFI) for the XEID prefix being requested. This length is also available in the header of the packet.

3. Full LISP-DDT Database Transfer

A full LISP-DDT database transfer is initiated by a DDT XFR client as a FDDTXFR request sent to a DDT XFR server. A DDT XFR server may choose to ignore full transfer requests received from the same DDT XFR client more than once in a 12 hour period.

This request is answered by successive DDTDAT messages until the entire portion of the database is sent from the DDT XFR server.

4. Incremental LISP-DDT Database Transfer

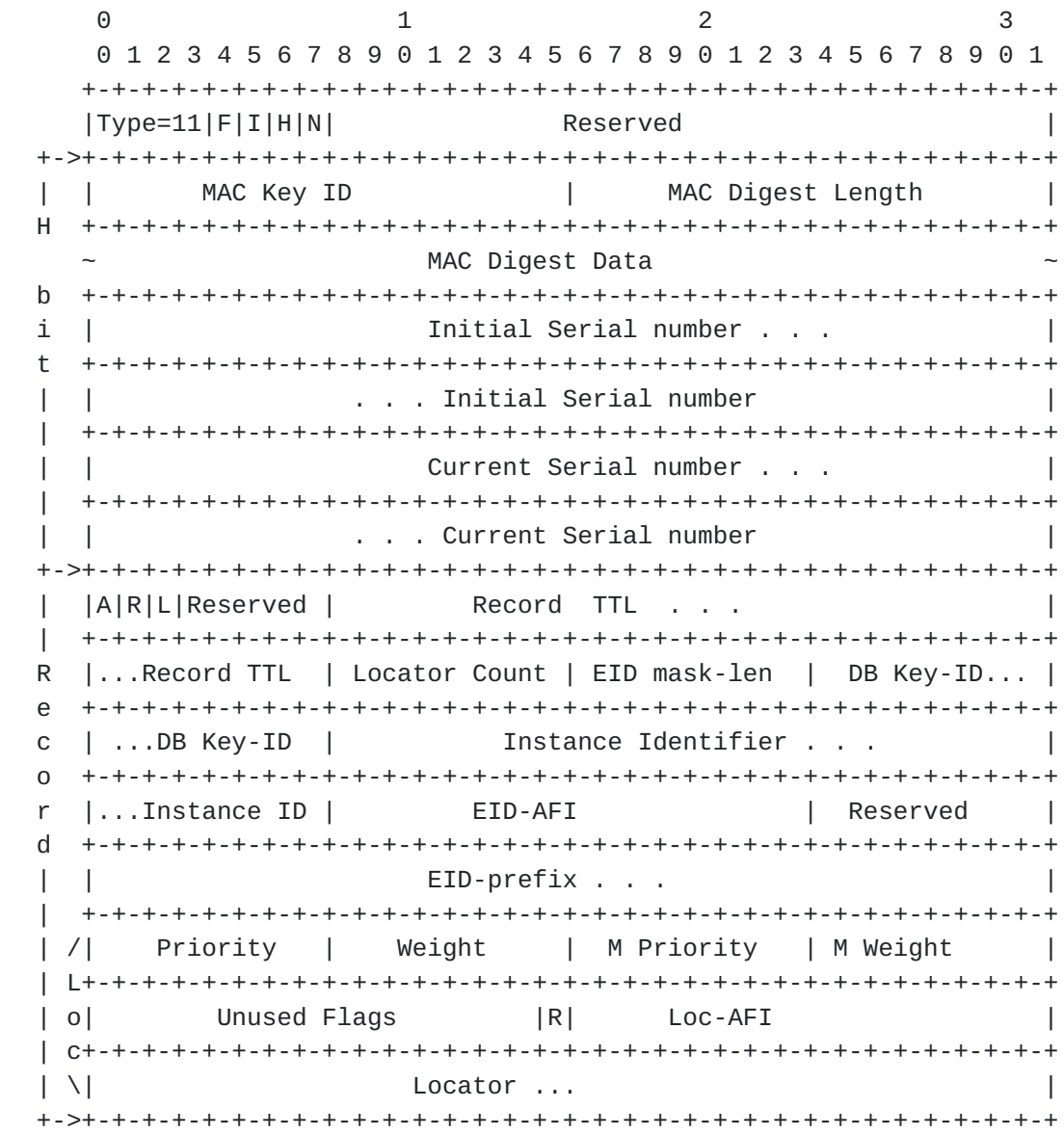
An incremental LISP-DDT database transfer is initiated by a DDT XFR client as a IDDTXFER request sent to a DDT XFR server. The DDT XFR client specifies a serial number which is used by the DDT XFR server to determine which changes must be sent in the reply.

The DDT XFR server should respond to the IDDTXFER request with an IDDTXFER reply that includes only the changes between the specified serial number and the current serial number. The DDT XFR server may

choose to respond with an FDDTXFER reply if the changes between the serial numbers are no longer available to the DDT XFR server.

The DDT XFR client must be prepared to receive an FDDTXFER reply in response to a IDDTXFER request.

5. LISP-DDT Transfer Data Message



LISP-DDT Transfer Data Message Format

Packet field descriptions are identical to those for the LISP Map

Reply, new fields are described below:

Type=11: DDT Transfer Data Message, this four bit value identifies the packet type (other LISP packet types are defined in the respective Internet Drafts).

F (bit flag): If this bit is set then this is a response to a full transfer request.

I (bit flag): If this bit is set then this is a response to an incremental transfer request.

H (bit flag): If this bit is set then the data header is included in the message. This header includes the message digest and serial numbers of the versions of the database. The header typically is included only in the first data message sent in response to a request.

N (bit flag): NXDB, this bit is set in a response if the portion of the database requested is not delegated to the DDT XFR server responding to the request. If this bit is set then the Extended EID-prefix is populated with the values specified in the original request.

MAC Key ID: Identifies the Message Authentication Code (MAC) digest algorithm per [LISP].

MAC Digest length: Length in bytes of the following MAC digest data.

MAC Digest Data: The MAC digest data depends on the MAC Key ID specified and is calculated on all of the fields that follow in this message.

Initial Serial number: In an IDDTXFER reply, the initial serial number is the serial number specified in the IDDTXFER request provided that serial number was used to generate the changes specified in the data messages. In a FDDTXFER request or in the case that the DDT XFR server could not determine the changes between the specified and current serial numbers then this field is filled with 0's. If the H bit is not set then this serial number is not present and the Extended EID prefix begins at this location in the packet.

Current Serial number: In both the IDDTXFER and FDDTXFER reply, the current serial number is provided in this field. If the H bit is not set then this serial number is not present.

A (bit flag): For incremental transfers this bit is set to indicate that the Extended EID prefix is to be added to the database. If this bit is not set then the Extended EID prefix is to be removed from the database. In cases where the Extended EID prefix is duplicated by the add, the record is replaced in the database.

R (bit flag): For incremental transfers this bit is set to indicate that the Extended EID prefix is to be removed from the database. The A and R bits are mutually exclusive.

L (bit flag): This bit is set only in the last record of the DDTXFER reply to indicate that no more records follow.

6. LISP-DDT Database Transfer Notify

A DDT XFR server maintains a list of DDT XFR clients that have subscribed to the portion of the LISP-DDT database tree delegated to that DDT XFR server. A DDT XFR client subscribes to a DDT XFR server by communicating with the DDT XFR server operator who then configures the DDT XFR server to send notifications. When the DDT XFR server increments the serial number for the delegated portion of the database, all DDT XFR clients are notified of the change via a DDTNTFY message which is identical to a DDTXFER request message with the N bit set.

Notifications are sent to DDT XFR clients within five minutes of the change to the serial number. A DDT XFR server may choose to only send the most recent/current serial number rather than all of the serial numbers that were used between notifications to a DDT XFR client.

Notifications are delivered via a reliable transport (TCP) and the DDT XFR server must attempt to connect with the DDT XFR client and send the notification at least three times within a three minute window no more frequently than once per minute. The DDT XFR server may choose a higher number of attempts at sending the notification, but not more frequently than once per minute.

7. Acknowledgments

Special thanks to folks that offered their considerable experience in building and operating large scale DNS platforms and helping translate this experience to LISP-DDT database transfers including Dave Blacka, Piet Barber and Sean Mountcastle. I also appreciate the time that Neel Goyal and Ramin Ali Dousti have put into this effort.

8. IANA Considerations

This memo includes no request to IANA.

9. Security Considerations

There are a few Security considerations specific to LISP-DDT Database Transfers that are incremental to the LISP-DDT specification. Most of the security related requirements are satisfied by the use of TLS. Although TLS as described in RFC 5246 [RFC5246] is intended to be transparent to the application protocol, certain details should be addressed to ensure consistent implementation of LISP-DDT Transfer. These details are addressed in a separate document.

Two of the approaches to implementing security related features in the protocol are to offer the secure service on an alternate port or to allow session endpoints to upgrade an existing session via TLS.

The preferred approach to securing LISP-DDT Transfer is to offer the secure service on an alternate port from the "un-secure" service. The primary motivation for this approach is to improve operational efficiency, for example this makes it easier to deploy a footprint that efficiently allocates hardware used to accelerate TLS termination.

An alternative to offering the secure service on an different network port is to support session upgrades. In this case either endpoint of a LISP-DDT transfer session may upgrade that session via TLS similar to the approach taken to upgrade an HTTP session via TLS as described in RFC 2817 [RFC2817].

The use of TLS provides for the three primary security considerations:

- Host authentication, both DDT XFR server and DDT XFR client can be reasonably confident of the identity of the other endpoint.

- Transfer integrity, the contents of the transfer can be reasonably assured to be unadulterated. The use of a message digest as is currently specified for the protocol affords message integrity as well.

- Transfer confidentiality, the contents of the transfer will be essentially opaque to hosts outside the session.

10. References

10.1. Normative References

[LISP] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol (LISP), draft-ietf-lisp-22.txt", February 2012.

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

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2817] Khare, R. and S. Lawrence, "Upgrading to TLS Within HTTP/1.1", RFC 2817, May 2000.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

10.2. Informative References

[LISP-DDT] Fuller, V., "LISP Delegated Database Tree, draft-ietf-lisp-ddt-01.txt (work in progress)", October 2011.

Appendix A. Open Issues

- A subscription message to manage DDT XFR client subscriptions to DDT XFR servers for portions of the database would be very useful. This requires a robust mechanism for authenticating subscribe/unsubscribe requests as part of the application protocol rather than relying entirely on TLS.

- Compression should be included in the specification as an optional feature. This will become more important as LISP is more widely adopted and the size of the LISP-DDT increases.

- Do we need to support some sort of discovery protocol, for example should a DDT XFR client be able to not specify some portion of the extended EID prefix so that a DDT XFR server can reply with a "default" database or offer some means to walk the portion of the tree delegated to the DDT XFR server?

Author's Address

Glen Wiley (editor)
Verisign, Inc.
12061 Bluemont Way
Reston, Va 20190
USA

Phone:
Email: gwiley@verisign.com