

LISP Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: April 4, 2019

A. Rodriguez-Natal  
Cisco Systems  
A. Cabellos-Aparicio  
Technical University of Catalonia  
S. Barkai  
Fermi Serverless  
V. Ermagan  
D. Lewis  
F. Maino  
Cisco Systems  
D. Farinacci  
lispers.net  
October 1, 2018

**LISP support for Multi-Tuple EIDs**  
**draft-rodrigueznatal-lisp-multi-tuple-eids-06**

Abstract

This document describes extensions for LISP to support EIDs based on tuples of multiple elements.

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 April 4, 2019.

Copyright Notice

Copyright (c) 2018 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 (<https://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

<a href="#">1.</a>	Introduction . . . . .	<a href="#">2</a>
<a href="#">1.1.</a>	Requirements Language . . . . .	<a href="#">2</a>
<a href="#">2.</a>	Definition of terms . . . . .	<a href="#">3</a>
<a href="#">3.</a>	Overview . . . . .	<a href="#">3</a>
<a href="#">4.</a>	Protocol Operation with Extended-EIDs . . . . .	<a href="#">4</a>
<a href="#">4.1.</a>	LISP Tunnel Routers . . . . .	<a href="#">4</a>
<a href="#">4.2.</a>	Mapping System . . . . .	<a href="#">4</a>
<a href="#">5.</a>	Extended-EIDs Encoding . . . . .	<a href="#">5</a>
<a href="#">5.1.</a>	5-Tuple . . . . .	<a href="#">5</a>
<a href="#">6.</a>	Extended-EID Lookups . . . . .	<a href="#">5</a>
<a href="#">6.1.</a>	5-Tuple (Coarse) . . . . .	<a href="#">5</a>
<a href="#">6.2.</a>	5-Tuple (Exact) . . . . .	<a href="#">6</a>
<a href="#">7.</a>	Mapping Databases for Extended-EIDs . . . . .	<a href="#">7</a>
<a href="#">7.1.</a>	5-Tuple (Coarse) . . . . .	<a href="#">7</a>
<a href="#">7.2.</a>	5-Tuple (Exact) . . . . .	<a href="#">7</a>
<a href="#">8.</a>	A Note on Instance-ID . . . . .	<a href="#">7</a>
<a href="#">9.</a>	Acknowledgments . . . . .	<a href="#">7</a>
<a href="#">10.</a>	IANA Considerations . . . . .	<a href="#">7</a>
<a href="#">11.</a>	Security Considerations . . . . .	<a href="#">8</a>
<a href="#">12.</a>	Normative References . . . . .	<a href="#">8</a>
	Authors' Addresses . . . . .	<a href="#">8</a>

## [1.](#) Introduction

This document describes how LISP handles lookups based on Extended-EIDs, i.e. tuples of  $n$  elements. Particularly it describes how the Tunnel Routers and the Mapping System operate when Extended-EIDs are in place, the different types of Extended-EIDs defined so far, how the lookup is performed for each Extended-EID type and which mapping databases are recommended to use depending on the kind of Extended-EIDs used.

### [1.1.](#) 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. Definition of terms

- o n-tuple: The term n-tuple is used in this document to describe the set of n elements present in a data packet (e.g. IP address, port, protocol) that can be used to identify unequivocally a packet or a set of packets.
- o 5-tuple: The term 5-tuple is used in this document to describe the set comprised by 5 elements, being these the source IP address, the destination IP address, the level 4 protocol number, the level 4 protocol source port and the level 4 protocol destination port of a data packet.
- o Extended-EID: This document uses the term Extended-EID to refer to any n-tuple (including a 5-tuple) used in a EID role. See as well [[RFC8111](#)]
- o Flow: The term flow is used in this document to refer to the sequence of packets identified by the same n-tuple.
- o MT-[xTR, RTR, MS, MR]: A LISP [xTR, RTR, MS, MR] that supports the enhanced operation for Multi-Tuple Extended-EIDs described in this document.
- o MT-TR: A LISP tunnel router (e.g. xTR, RTR) that supports the enhanced operation for Multi-Tuple Extended-EIDs described in this document, e.g. MT-xTR, MT-RTR.

The rest of the terms are defined in their respective documents. See the LISP specification [[I-D.ietf-lisp-rfc6833bis](#)] for most of the definitions, [[RFC8060](#)] for LCAF and [[I-D.ietf-lisp-te](#)] for RTR.

## 3. Overview

This document describes extensions for LISP to support Multi-Tuple Extended-EIDs. Protocol operation follows the specification defined on [[I-D.ietf-lisp-rfc6833bis](#)] except for the following. Besides of IP mappings, a Mapping System can store Extended-EID mappings. Being Extended-EID a n-tuple identifying a flow. LISP routers perform look-ups based on these Extended-EIDs, instead of on destination IPs. Apart from using n-tuples instead of IPs, retrieving information from the Mapping System follows LISP standard mechanisms (i.e. Map-Request, Map-Reply).



## **4. Protocol Operation with Extended-EIDs**

### **4.1. LISP Tunnel Routers**

LISP tunnel routers with enhanced operation for Multi-Tuple Extended-EIDs, or MT-TRs (MT-xTRs and MT-RTRs), behave as specified on and [\[I-D.ietf-lisp-rfc6833bis\]](#), with the particularity that MT-TRs perform mapping lookups based on Extended-EIDs (n-tuples).

Any MT-TR must keep an internal map-cache indexed by Extended-EIDs. When a MT-TR receives a packet to encapsulate, it extracts the fields required by the n-tuple lookup in use and stores them in an Extended-EID structure. In the case of a 5-tuple lookup, it will extract the source address, destination address, level 4 protocol, source port (if any) and destination port (if any) from the packet. The MT-TR uses the Extended-EID to perform a look-up into the map-cache. The lookup process must follow the procedure described in [section Section 6](#). If there is an entry on the map-cache that matches the Extended-EID, the MT-TR retrieves the mapping information, selects a destination RLOC and encapsulates the packet, as defined in [\[I-D.ietf-lisp-rfc6833bis\]](#)

If the map-cache of the MT-TR contains no entry for the Extended-EID, the MT-TR sends a Map-Request to a MT-MR. The MT-TR MUST be provisioned with the RLOC of at least one MT-MR. The Map-Request sent carries the Extended-EID (encoded in the specific LCAF for that Extended-EID type) in the EID-prefix field of the Map-Request. This Map-Request will eventually trigger a Map-Reply to be sent back the requester MT-TR, see [section Section 4.2](#). This Map-Reply carries an Extended-EID on the EID-prefix field. The MT-TR stores, as defined in [\[I-D.ietf-lisp-rfc6833bis\]](#), the mapping for the Extended-EID.

### **4.2. Mapping System**

Mapping System elements (comprising Map Servers and Map Resolvers) behave as specified on [\[I-D.ietf-lisp-rfc6833bis\]](#) when implementing enhanced Multi-Tuple Extended-EIDs operation, with the particularity that MT-MRs resolve Map-Requests based on Extended-EIDs and MT-MSs store mappings indexed by Extended-EIDs.

MT-MRs must be capable of processing Map-Requests with an Extended-EID on the EID-prefix field, of finding the appropriate MT-MS for the Extended-EID and of forwarding the Map-Request to it. This is done according to the lookup rules described in [section Section 6](#) and using the mapping database described in [section Section 7](#) which differs depending on the specific Extended-EID.



LISP elements must perform the mapping update mechanisms defined in [\[I-D.ietf-lisp-rfc6833bis\]](#) (e.g, SMR) using as EID the Extended-EID.

## **5. Extended-EIDs Encoding**

This section describes the Extended-EID types defined so far and the LCAFs to support them.

### **5.1. 5-Tuple**

The 5-tuple LCAF is a combination of Application Data Type 4 and Source/Dest Type 12 LCAFs. Experimental deployment may indicate that a specific 5-tuple type LCAF is necessary.

## **6. Extended-EID Lookups**

This section describes the lookup process to be followed when using Extended-EID. At this point, this document only covers 5-tuple kind of Extended-EID lookups (with options for coarse or exact lookup). It is expected to include lookup mechanism for n-tuple lookups with more complex protocol combinations.

### **6.1. 5-Tuple (Coarse)**

When using a coarse lookup for 5-tuple, the encoding described in [Section 5.1](#) is used to carry the Extended-EID. Note that a coarse lookup also covers exact lookups. The lookup is (logically) done at steps, one per each element of the tuple. The lookup MUST follow this strict order:

- (1) Destination address
- (2) Source address
- (3) Protocol number
- (4) Destination port
- (5) Source port

This means that for a given 5-tuple, the lookup process will first select from the available 5-tuples present in the system, the ones that match the destination address. Among them, those that also match the source address. This is iterated for the rest of the elements in the tuple. If a 5-tuple matches several entries, then the one with the longest prefix match or shortest port-range has priority. To clarify the process an example is provided below.





Suppose that a MT-MS stores the mappings indexed by the tuples (A), (B), (C), (D) and (E), and that it receives Map-Request messages carrying the Extended-EIDs (T), (U), (V), (W), (X) and (Y).

	dst-add	src-add	pr	dst-prt	src-prt
(A)	[1.1.1.0/24,	2.2.0.0/16,	17,	1000-3000,	1000-3000]
(B)	[1.1.0.0/16,	2.2.2.0/24,	17,	1000-3000,	1000-3000]
(C)	[3.3.3.0/24,	4.4.4.0/24,	6,	4000-4500,	7000-8000]
(D)	[3.3.3.0/24,	4.4.4.0/24,	6,	4000-6000,	7000-8000]
(E)	[5.5.5.0/24,	6.6.6.0/24,	17,	0-65535,	0-65535]
(T)	[ 1.1.1.8,	2.2.2.9,	17,	2000,	2000 ]
(U)	[ 1.1.8.8,	2.2.9.9,	17,	2000,	2000 ]
(V)	[ 1.1.8.8,	2.2.2.9,	17,	2000,	2000 ]
(W)	[ 3.3.3.3,	4.4.4.4,	6,	4300,	7500 ]
(X)	[ 3.3.3.3,	4.4.4.4,	6,	5000,	7500 ]
(Y)	[ 5.5.5.5,	6.6.6.6,	17,	6000,	6000 ]

#### 5-tuple example for coarse lookup

- (1) When (T) is Map-Requested, the lookup could match both (A) and (B), however destination address has preference over source address and therefore (A) is returned.
- (2) When (U) is Map-Requested, the lookup will return that no entry exists for the 5-tuple.
- (3) When (V) is Map-Requested, the lookup will return (B).
- (4) When (W) is Map-Requested, the lookup will return (C) and not (D), although both could match the tuple, since the destination port range is shorter in (C).
- (5) When (X) is Map-Requested, the lookup will return (D).
- (6) When (Y) is Map-Requested, the lookup will return (E). A port range of 0-65535 means any port.

### 6.2. 5-Tuple (Exact)

In scenarios where 5-tuple coarse lookup is not required, the lookup can be optimized to only account for exact matches. When using a exact lookup for 5-tuple, the encoding described in [Section 6.1](#) is used to carry the Extended-EID. The exact match lookup is performed by serializing the elements of the 5-tuple as a single vector of



bits. The order to serialize the elements is the same that is described in [Section 5.1](#). This (unique) vector of bits is then used as the key to perform an exact match lookup over the available entries.

## **7. Mapping Databases for Extended-EIDs**

The mapping database, i.e. the system to interconnect (MT-)MRs and (MT-)MSs should be optimal for each one of the different Extended-EIDs types. This section covers recommended mapping databases for each of the Extended-EIDs described in this document.

### **7.1. 5-Tuple (Coarse)**

The mapping database to be used for a coarse lookup of 5-tuples can leverage on the LISP-DDT mapping database [[RFC8111](#)] since it supports multi-tuple lookups. Note that a LISP-DDT based database can support also an exact lookup.

### **7.2. 5-Tuple (Exact)**

Although a LISP-DDT based mapping database supports both coarse and exact lookups, the particularities of the latter benefit of using a mapping database optimized for flat namespaces rather than one optimized for hierarchical data. In that sense, the exact match mechanism should be supported by a DHT-like mapping database, such as [[I-D.cheng-lisp-shdht](#)] or [LISP-DHT].

## **8. A Note on Instance-ID**

Instance-ID is a special case to be considered. If it is in use, its lookup is resolved before the lookup for the Extended-EID begins (regardless of the Extended-EID type). In terms of implementation this means that if the Instance-ID is present, it will have always more priority than any other field within the multi-tuple EID. In other words, Instance-ID is the high-order parts of the destination and source addresses and a longest match lookup should be applied to it before looking up the address itself.

## **9. Acknowledgments**

## **10. IANA Considerations**

This memo includes no request to IANA.



## **11. Security Considerations**

Security Considerations TBD

## **12. Normative References**

[I-D.cheng-lisp-shdht]

Cheng, L. and J. Wang, "LISP Single-Hop DHT Mapping Overlay", [draft-cheng-lisp-shdht-04](#) (work in progress), July 2013.

[I-D.ietf-lisp-rfc6833bis]

Fuller, V., Farinacci, D., and A. Cabellos-Aparicio, "Locator/ID Separation Protocol (LISP) Control-Plane", [draft-ietf-lisp-rfc6833bis-15](#) (work in progress), September 2018.

[I-D.ietf-lisp-te]

Farinacci, D., Kowal, M., and P. Lahiri, "LISP Traffic Engineering Use-Cases", [draft-ietf-lisp-te-02](#) (work in progress), April 2018.

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

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

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

### Authors' Addresses

Alberto Rodriguez-Natal  
Cisco Systems  
170 Tasman Drive  
San Jose, CA  
USA

Email: natal@cisco.com



Albert Cabellos-Aparicio  
Technical University of Catalonia  
Barcelona  
Spain

Email: [acabello@ac.upc.edu](mailto:acabello@ac.upc.edu)

Sharon Barkai  
Fermi Serverless  
CA  
USA

Email: [sharon@fermicloud.io](mailto:sharon@fermicloud.io)

Vina Ermagan  
Cisco Systems  
170 Tasman Drive  
San Jose, CA  
USA

Email: [vermagan@cisco.com](mailto:vermagan@cisco.com)

Darrel Lewis  
Cisco Systems  
170 Tasman Drive  
San Jose, CA  
USA

Email: [darlewis@cisco.com](mailto:darlewis@cisco.com)

Fabio Maino  
Cisco Systems  
170 Tasman Drive  
San Jose, CA  
USA

Email: [fmaino@cisco.com](mailto:fmaino@cisco.com)





Dino Farinacci  
lispers.net  
San Jose, CA  
USA

Email: farinacci@gmail.com