

LISP Working Group
Internet-Draft
Intended status: Experimental
Expires: September 24, 2015

V. Ermagan
Cisco Systems
A. Rodriguez-Natal
F. Coras
A. Cabellos-Aparicio
Technical University of Catalonia
F. Maino
Cisco Systems
March 23, 2015

YANG model for LISP
draft-ermagan-lisp-yang-00

Abstract

This document describes a YANG data model to use with the Locator/ID Separation Protocol (LISP).

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 24, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in [Section 4.e](#) of

Internet-Draft

LISP-YANG

March 2015

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Tree view	2
3.	YANG model	11
4.	Acknowledgments	25
5.	IANA Considerations	25
6.	Security Considerations	25
7.	Normative References	25
	Authors' Addresses	26

[1.](#) Introduction

The Locator/ID Separation Protocol (LISP) defines several network elements subject to be configured. This document presents a YANG data model to define the basic configuration of all major LISP elements.

[2.](#) Tree view

The tree view of the model is depicted below. The following notation is used to describe elements within the tree. For readability purposes, the tree depth is limited to 9 levels.

Each node is printed as:

<status> <flags> <name> <opts> <type> <if-features>

<status> is one of:

- + for current
- x for deprecated
- o for obsolete

<flags> is one of:

- rw for configuration data
- ro for non-configuration data
- x for rpcs
- n for notifications

<name> is the name of the node

- (<name>) means that the node is a choice node
- :(<name>) means that the node is a case node

If the node is augmented into the tree from another module, its name is printed as <prefix>:<name>.

<opts> is one of:

- ? for an optional leaf or choice
- ! for a presence container
- * for a leaf-list or list
- [<keys>] for a list's keys

<type> is the name of the type for leafs and leaf-lists

<if-features> is the list of features this node depends on, printed within curly brackets and a question mark "{...}?"

```
module: lisp
  +--rw itr-cfg! {itr}?
```

```

|   +--rw rloc-probing!
|   |   +--rw interval?          uint16
|   |   +--rw retries?          uint8
|   |   +--rw retries-interval?  uint16
|   +--rw itr-rlocs
|   |   +--rw itr-rloc* [id]
|   |   |   +--rw id            string
|   |   |   +--rw address
|   |   |   |   +--rw afi?          enumeration
|   |   |   |   +--rw instance-id? instance-id-type
|   |   |   |   +--rw (address)?
|   |   |   |   +--:(ipv4)

```

```

|   |   |   +--rw ipv4?          inet:ipv4-address
|   |   |   +--:(ipv6)
|   |   |   |   +--rw ipv6?      inet:ipv6-address
|   |   |   |   +--:(mac-address)
|   |   |   |   +--rw mac-address? yang:mac-address
|   |   |   |   +--:(lcaf)
|   |   |   |   +--rw lcaf
|   |   |   |   |   +--rw lcaf-type?          enumeration
|   |   |   |   |   +--rw (address)?
|   |   |   |   |   |   +--:(as-number)
|   |   |   |   |   |   |   ...
|   |   |   |   |   |   +--:(sourc-dest-key)
|   |   |   |   |   |   |   ...
|   |   |   |   |   |   +--:(explicit-locator-path)
|   |   |   |   |   |   ...
|   +--rw local-eids
|   |   +--rw local-eid* [id]
|   |   |   +--rw id            eid-id
|   |   |   +--rw eid-address
|   |   |   |   +--rw afi?          enumeration
|   |   |   |   +--rw instance-id? instance-id-type
|   |   |   |   +--rw (address)?
|   |   |   |   +--:(ipv4)
|   |   |   |   |   +--rw ipv4?      inet:ipv4-address
|   |   |   |   |   +--:(ipv6)
|   |   |   |   |   +--rw ipv6?      inet:ipv6-address
|   |   |   |   |   +--:(mac-address)
|   |   |   |   |   +--rw mac-address? yang:mac-address
|   |   |   |   |   +--:(lcaf)

```

```

+--rw lcaf
+--rw lcaf-type? enumeration
+--rw (address)?
+--:(as-number)
|   ...
+--:(sourc-dest-key)
|   ...
+--:(explicit-locator-path)
|   ...
+--rw map-resolvers
+--rw map-resolver* [id]
+--rw id eid-id
+--rw eid-address
|   +--rw afi? enumeration
|   +--rw instance-id? instance-id-type
|   +--rw (address)?
|       +--:(ipv4)
|       |   +--rw ipv4? inet:ipv4-address
|       +--:(ipv6)

```

```
| | | | +--rw ipv6?          inet:ipv6-address
| | | | +---:(mac-address)
| | | | |   +--rw mac-address?    yang:mac-address
| | | | +---:(lcaf)
| | | | |   +--rw lcaf
| | | | |       +--rw lcaf-type?      enumeration
| | | | |       +--rw (address)?
| | | | |           +---:(as-number)
| | | | |               |
| | | | |               ...
| | | | |           +---:(sourc-dest-key)
| | | | |               |
| | | | |               ...
| | | | |           +---:(explicit-locator-path)
| | | | |               ...
+--rw map-resolver-address*    inet:ip-address
+--rw proxy-etrs
|   +--rw proxy-etr* [id]
|   +--rw id              eid-id
|   +--rw eid-address
|   |   +--rw afi?        enumeration
|   |   +--rw instance-id? instance-id-type
|   |   +--rw (address)?
|   |       +---:(ipv4)
```



```

|         |   +---:(negative-mapping)
|         |   |   +---rw map-reply-action?   map-reply-action
|         |   +---:(positive-mapping)
|         |   +---rw rlocs
|         |       +---rw rloc* [name]
|         |           +---rw name                               string
|         |           +---rw (address-type)?
|         |           |   +---:(interface-address)
|         |           |   |   ...
|         |           |   +---:(lisp-address)
|         |           |   |   ...
|         |           +---rw priority?                        uint8
|         |           +---rw weight?                          uint8
|         |           +---rw multicast-priority?             uint8
|         |           +---rw multicast-weight?               uint8
|         |   +---rw static?                                boolean
+---rw etr-cfg! {etr}?
|   +---rw local-eids
|       +---rw local-eid* [id]
|           +---rw id                                eid-id
|           +---rw eid-address
|               +---rw afi?                            enumeration
|               +---rw instance-id?                    instance-id-type
|               +---rw (address)?
|                   +---:(ipv4)
|                   |   +---rw ipv4?                    inet:ipv4-address
|                   +---:(ipv6)
|                   |   +---rw ipv6?                    inet:ipv6-address
|                   +---:(mac-address)
|                   |   +---rw mac-address?            yang:mac-address
|                   +---:(lcaf)
|                       +---rw lcaf
|                           +---rw lcaf-type?            enumeration

```

```

|         |   +---rw (address)?
|         |       +---:(as-number)
|         |       |   ...
|         |       +---:(sourc-dest-key)
|         |       |   ...
|         |       +---:(explicit-locator-path)
|         |       |   ...
+---rw map-servers

```

```

|         | +---rw map-server* [address]
|         |         +---rw address          inet:ip-address
|         |         +---rw auth-key?        string
|         |         +---rw auth-key-type?   auth-key-type
|         +---rw rlocs
|         | +---rw rloc* [name]
|         |         +---rw name              string
|         |         +---rw (address-type)?
|         |         | +---:(interface-address)
|         |         | | +---rw interface?    interface-name
|         |         | +---:(lisp-address)
|         |         |         +---rw locator-address
|         |         |         +---rw afi?      enumeration
|         |         |         +---rw instance-id? instance-id-type
|         |         |         +---rw (address)?
|         |         |         ...
|         |         +---rw priority?          uint8
|         |         +---rw weight?            uint8
|         |         +---rw multicast-priority? uint8
|         |         +---rw multicast-weight?   uint8
|         +---rw record-ttl?                  uint32
|         +---rw want-map-notify?             boolean
|         +---rw proxy-reply?                 boolean
|         +---rw registration-interval?       uint16
+---rw map-server-cfg! {map-server}?
|   +---rw sites
|   |   +---rw site* [site-id]
|   |   |   +---rw site-id    uint64
|   |   |   +---rw devices
|   |   |   |   +---rw device* [device-id]
|   |   |   |   |   +---rw device-id    uint64
|   |   |   |   |   +---rw auth-key
|   |   |   |   |   |   +---rw auth-key-value? string
|   |   |   |   |   |   +---rw auth-key-type? auth-key-type
|   |   |   +---rw eids
|   |   |   |   +---rw eid* [id]
|   |   |   |   |   +---rw id              eid-id
|   |   |   |   |   +---rw eid-address
|   |   |   |   |   |   +---rw afi?          enumeration
|   |   |   |   |   |   +---rw instance-id? instance-id-type

```

```

|   |   |   |   |   |   +---rw (address)?

```



```

|                                     |
|                                     | ...
|                                     | +--rw more-specifics-accepted?      boolean
|                                     | +--rw mapping-expiration-timeout?  int16
+--rw registered-mappings
|   +--rw mapping* [id]
|     +--rw id                               eid-id
|     +--rw eid
|       +--rw afi?                           enumeration
|       +--rw instance-id?                   instance-id-type
|       +--rw (address)?
|         +--:(ipv4)
|           | +--rw ipv4?                     inet:ipv4-address
|         +--:(ipv6)
|           | +--rw ipv6?                     inet:ipv6-address
|         +--:(mac-address)
|           | +--rw mac-address?              yang:mac-address
|         +--:(lcaf)
|           +--rw lcaf
|             +--rw lcaf-type?                 enumeration
|             +--rw (address)?
|               +--:(as-number)
|                 | ...
|               +--:(sourc-dest-key)
|                 | ...
|               +--:(explicit-locator-path)
|                 | ...
|             +--rw ttl?                       uint32
|             +--rw (locator-list)?
|               +--:(negative-mapping)
|                 | +--rw map-reply-action?    map-reply-action
|               +--:(positive-mapping)
|                 +--rw rlocs
|                   +--rw rloc* [name]
|                     +--rw name                string
|                     +--rw (address-type)?
|                       | +--:(interface-address)
|                       | | ...
|                       | +--:(lisp-address)
|                       | | ...
|                     +--rw priority?            uint8
|                     +--rw weight?              uint8
|                     +--rw multicast-priority?  uint8
|                     +--rw multicast-weight?    uint8
+--rw (mapping-system)
|   +--:(ddt-mapping-system)
|     +--rw ddt-mapping-system!
|       +--rw ddt-root*                       inet:ip-address

```

```

|         +---rw authoritative-eids
|         |   +---rw authoritative-eid* [id]
|         |   |   +---rw id                eid-id
|         |   |   +---rw eid-address
|         |   |   |   +---rw afi?            enumeration
|         |   |   |   +---rw instance-id?    instance-id-type
|         |   |   |   +---rw (address)?
|         |   |   |   |   +--:(ipv4)
|         |   |   |   |   |   ...
|         |   |   |   |   +--:(ipv6)
|         |   |   |   |   |   ...
|         |   |   |   |   +--:(mac-address)
|         |   |   |   |   |   ...
|         |   |   |   |   +--:(lcaf)
|         |   |   |   |   |   ...
|         |   |   +--:(alt-mapping-system)
|         |   |   |   +---rw alt-mapping-system!
+---rw map-resolver-cfg! {map-resolver}?
|   +---rw (mapping-system)
|   |   +--:(ddt-mapping-system)
|   |   |   +---rw ddt-mapping-system!
|   |   |   |   +---rw ddt-root*    inet:ip-address
|   |   +--:(alt-mapping-system)
|   |   |   +---rw alt-mapping-system!
+---rw proxy-itr-cfg! {proxy-itr}?
|   +---rw servicing-eids
|   |   +---rw eid* [id]
|   |   |   +---rw id                eid-id
|   |   |   +---rw eid-address
|   |   |   |   +---rw afi?            enumeration
|   |   |   |   +---rw instance-id?    instance-id-type
|   |   |   |   +---rw (address)?
|   |   |   |   |   +--:(ipv4)
|   |   |   |   |   |   +---rw ipv4?    inet:ipv4-address
|   |   |   |   |   |   +--:(ipv6)
|   |   |   |   |   |   |   +---rw ipv6?    inet:ipv6-address
|   |   |   |   |   |   |   +--:(mac-address)
|   |   |   |   |   |   |   |   +---rw mac-address?    yang:mac-address
|   |   |   |   |   |   |   |   +--:(lcaf)
|   |   |   |   |   |   |   |   |   +---rw lcaf
|   |   |   |   |   |   |   |   |   |   +---rw lcaf-type?    enumeration
|   |   |   |   |   |   |   |   |   |   +---rw (address)?
|   |   |   |   |   |   |   |   |   |   |   +--:(as-number)
|   |   |   |   |   |   |   |   |   |   |   |   ...
|   |   |   |   |   |   |   |   |   |   |   +--:(sourc-dest-key)
|   |   |   |   |   |   |   |   |   |   |   |   ...

```

```
| | +--:(explicit-locator-path)
| | ...
```

Internet-Draft

LISP-YANG

March 2015

```

+--rw map-resolvers
|   +--rw map-resolver* [id]
|   |   +--rw id          eid-id
|   |   +--rw eid-address
|   |   |   +--rw afi?          enumeration
|   |   |   +--rw instance-id?  instance-id-type
|   |   |   +--rw (address)?
|   |   |   |   +--:(ipv4)
|   |   |   |   |   +--rw ipv4?          inet:ipv4-address
|   |   |   |   +--:(ipv6)
|   |   |   |   |   +--rw ipv6?          inet:ipv6-address
|   |   |   |   +--:(mac-address)
|   |   |   |   |   +--rw mac-address?  yang:mac-address
|   |   |   |   +--:(lcaf)
|   |   |   |   |   +--rw lcaf
|   |   |   |   |   |   +--rw lcaf-type?          enumeration
|   |   |   |   |   |   +--rw (address)?
|   |   |   |   |   |   |   +--:(as-number)
|   |   |   |   |   |   |   |   ...
|   |   |   |   |   |   |   +--:(sourc-dest-key)
|   |   |   |   |   |   |   |   ...
|   |   |   |   |   |   |   +--:(explicit-locator-path)
|   |   |   |   |   |   |   |   ...
|   |   +--rw map-resolver*  inet:ip-address
+--rw map-cache
|   +--rw mapping* [id]
|   |   +--rw id          eid-id
|   |   +--rw eid
|   |   |   +--rw afi?          enumeration
|   |   |   +--rw instance-id?  instance-id-type
|   |   |   +--rw (address)?
|   |   |   |   +--:(ipv4)
|   |   |   |   |   +--rw ipv4?          inet:ipv4-address
|   |   |   |   +--:(ipv6)
|   |   |   |   |   +--rw ipv6?          inet:ipv6-address
|   |   |   |   +--:(mac-address)
|   |   |   |   |   +--rw mac-address?  yang:mac-address
|   |   |   |   +--:(lcaf)
|   |   |   |   |   +--rw lcaf

```



```

+--rw lcaf-type?          enumeration
+--rw (address)?
  +--:(as-number)
  |   ...
  +--:(sourc-dest-key)
  |   ...
  +--:(explicit-locator-path)
  |   ...

```

3. YANG model

This section contains the YANG model for lisp configuration and the companion lisp-address-types module.

<CODE BEGINS> file "lisp-address-types@2015-03-23.yang"

Ermagan, et al.

Expires September 24, 2015

[Page 11]

Internet-Draft

LISP-YANG

March 2015

```

module lisp-address-types {
  namespace "urn:ietf:params:xml:ns:yang:lisp-address-types";
  prefix lisp;

  import ietf-inet-types {
    prefix inet;
  }
  import ietf-yang-types {
    prefix yang;
  }

  organization "IETF LISP (Locator/ID Separation Protocol) Working Group";
  contact
    "lisp@ietf.org";
  description
    "This YANG module defines the LISP Canonical Address Formats (LCAF)
    for LISP. The module can be extended by vendors
    to define vendor-specific parameters."

```

Copyright (c) 2015 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License

set forth in [Section 4.c](#) of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of [RFC 6338](#); see the RFC itself for full legal notices.

```
";

revision 2015-03-23 {
  description
    "Initial revision.";
}

typedef instance-id-type {
  type uint32 {
    range "0..16777214";
  }
}

typedef simple-address {
  type union {
    type inet:ip-address;
```

```
  type yang:mac-address;
}

grouping lcac-address {
  leaf lcac-type {
    type enumeration {
      enum "null";
      enum "afi-list";
      enum "instance-id";
      enum "as-number";
      enum "application-data";
      enum "geo-coordinates";
      enum "opaque-key";
      enum "nat-ttraversal";
      enum "nonce-locator";
      enum "multicast-info";
      enum "explicit-locator-path";
```

```

        enum "security-key";
        enum "source-dest-key";
        enum "replication-list";
        enum "json-data-model";
        enum "key-value-address";
        enum "encapsulation-format";
    }
}
choice address {
    container as-number {
        when "lcaf-type = as-number";
        leaf as {
            type inet:as-number;
        }
        leaf address {
            type simple-address;
        }
    }
    container sourc-dest-key {
        when "lcaf-type = source-dest-key";
        leaf source {
            type inet:ip-prefix;
        }
        leaf dest {
            type inet:ip-prefix;
        }
    }
}
container explicit-locator-path {
    when "lcaf-type = explicit-locator-path";
    list hop {

```

```

        key "address";
        leaf address {
            type simple-address;
        }
        leaf lrs-bits {
            type bits {
                bit lookup;
                bit rloc-probe;
                bit strict;
            }
        }
    }
}

```

```

    }
  }
}

grouping lisp-address {
  leaf afi {
    type enumeration {
      enum "ipv4" {
        value 1;
      }
      enum "ipv6" {
        value 2;
      }
      enum "mac-address" {
        value 6;
      }
      enum "lcaf" {
        value 16387;
      }
    }
  }
  leaf instance-id {
    type instance-id-type;
  }
  choice address {
    case ipv4 {
      when "afi = ipv4";
      leaf ipv4 {
        type inet:ipv4-address;
      }
    }
    case ipv6 {
      when "afi = ipv6";
      leaf ipv6 {
        type inet:ipv6-address;
      }
    }
  }
}

```

```

}
case mac-address {
  when "afi = mac-address";
  leaf mac-address {

```


Relating to IETF Documents

(<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of [RFC 6338](#); see the RFC itself for full legal notices.

```
";

revision 2015-03-23 {
  description
    "Initial revision.";
}

typedef interface-name {
  description
    "Name of a device interface";
  type string;
}

typedef map-reply-action {
  description
    "Defines the lisp map-cache ACT type";
  type enumeration {
    enum "no-action" {
      value 0;
    }
    enum "natively-forward" {
      value 1;
    }
    enum "send-map-request" {
      value 2;
    }
    enum "drop" {
      value 3;
    }
  }
}

typedef eid-id {
  type string;
}

typedef instance-id-type {
  type uint32 {
    range "0..16777214";
  }
}
```

Internet-Draft

LISP-YANG

March 2015

```
typedef auth-key-type {
  type enumeration {
    enum "none" {
      value 0;
    }
    enum "hmac-sha-1-96" {
      value 1;
    }
    enum "hmac-sha-256-128" {
      value 2;
    }
  }
}

feature itr {
  description
    "ITR operation supported";
}

feature etr {
  description
    "ETR operation supported";
}

feature proxy-itr {
  description
    "PITR operation supported";
}

feature proxy-etr {
  description
    "PETR operation supported";
}

feature map-server {
  description
    "MS operation supported";
}

feature map-resolver {
```

```
    description
      "MR operation supported";
  }
```

```
  grouping locators {
    list rloc {
      key "name";
      leaf name {
```

```
    type string;
  }
  choice address-type {
    case interface-address {
      leaf interface {
        type interface-name;
      }
    }
    case lisp-address {
      container locator-address {
        uses lcaf:lisp-address;
      }
    }
  }
  leaf priority {
    type uint8;
  }
  leaf weight {
    type uint8;
  }
  leaf multicast-priority {
    type uint8;
  }
  leaf multicast-weight {
    type uint8;
  }
}
}
```

```
grouping mappings {
  list mapping {
    key "id";
    leaf id {
```

```

    type eid-id;
  }
  container eid {
    uses lcaf:lisp-address;
  }
  leaf ttl {
    type uint32;
  }
  choice locator-list {
    case negative-mapping {
      leaf map-reply-action {
        type map-reply-action;
      }
    }
    case positive-mapping {

```

```

    container rlocs {
      uses locators;
    }
  }
  default "positive-mapping";
}
}
}

container itr-cfg {
  if-feature itr;
  presence "LISP ITR operation enabled";
  config true;
  container rloc-probing {
    presence "RLOC probing active";
    leaf interval {
      type uint16;
      units "seconds";
      description
        "Interval in seconds";
    }
    leaf retries {
      type uint8;
      description
        "Number of retries";
    }
  }
}

```

```

    leaf retries-interval {
      type uint16;
      units "seconds";
      description
        "Interval in seconds between retries";
    }
  }
  container itr-rlocs {
    description
      "List of RLOCs of the ITR included in Map-Requests";
    list itr-rloc {
      key "id";
      leaf id {
        type string;
      }
      container address {
        uses lcaf:lisp-address;
      }
    }
  }
  container local-eids {
    list local-eid {

```

```

      min-elements 1;
      key "id";
      leaf id {
        type eid-id;
      }
      container eid-address {
        uses lcaf:lisp-address;
      }
    }
  }
  container map-resolvers {
    list map-resolver {
      min-elements 1;
      key "id";
      leaf id {
        type eid-id;
      }
      container eid-address {
        uses lcaf:lisp-address;

```

```

    }
    leaf-list map-resolver-address {
        type inet:ip-address;
    }
}
}
container proxy-etr {
    list proxy-etr {
        key "id";
        leaf id {
            type eid-id;
        }
        container eid-address {
            uses lcaf:lisp-address;
        }
        leaf-list proxy-etr-address {
            type inet:ip-address;
        }
    }
}
}
container map-cache {
    uses mappings {
        augment "mapping" {
            leaf static {
                description
                    "A configured mapping is a static mapping. If the mapping
                     is learned, it is operational data and static is false.";
                type boolean;
                default "true";
            }
        }
    }
}

```

```

    }
    }
}
}
}
container etr-cfg {
    if-feature etr;
    presence "LISP ETR operation enabled";
    config true;
    container local-eids {
        list local-eid {
            min-elements 1;
        }
    }
}

```

```

key "id";
leaf id {
    type eid-id;
}
container eid-address {
    uses lcaf:lisp-address;
}
container map-servers {
    list map-server {
        key "address";
        leaf address {
            type inet:ip-address;
        }
        leaf auth-key {
            type string;
        }
        leaf auth-key-type {
            type auth-key-type;
        }
    }
}
container rlocs {
    uses locators;
}
leaf record-ttl {
    type uint32;
}
leaf want-map-notify {
    type boolean;
}
leaf proxy-reply {
    type boolean;
}
leaf registration-interval {
    units "seconds";
    type uint16;
}

```

```

        default "60";
    }
}
}
}

```



```

container map-server-cfg {
  if-feature map-server;
  presence "LISP Map Server operation enabled";
  config true;
  container sites {
    list site {
      key "site-id";
      leaf site-id {
        type uint64;
      }
    }
    container devices {
      list device {
        key "device-id";
        leaf device-id {
          type uint64;
        }
      }
      container auth-key {
        leaf auth-key-value {
          description
            "clear text authentication key";
          type string;
        }
        leaf auth-key-type {
          type auth-key-type;
        }
      }
    }
    container eids {
      list eid {
        key "id";
        leaf id {
          type eid-id;
        }
      }
      container eid-address {
        uses lcaf:lisp-address;
      }
      leaf more-specifics-accepted {
        type boolean;
      }
      leaf mapping-expiration-timeout {
        type int16;
        units "seconds";
        default "180";
      }
    }
  }
}

```

```

    }
  }
}
}
}
}
}
container registered-mappings {
  uses mappings;
}
choice mapping-system {
  mandatory true;
  container ddt-mapping-system {
    presence "DDT Mapping System in use";
    leaf-list ddt-root {
      type inet:ip-address;
    }
    container authoritative-eids {
      list authoritative-eid {
        key "id";
        leaf id {
          type eid-id;
        }
        container eid-address {
          uses lcaf:lisp-address;
        }
      }
    }
  }
  container alt-mapping-system {
    presence "ALT Mapping System in use";
  }
}
}
container map-resolver-cfg {
  if-feature map-resolver;
  presence "LISP Map-Resolver operation enabled";
  config true;
  choice mapping-system {
    mandatory true;
    container ddt-mapping-system {
      presence "DDT Mapping System in use";
      leaf-list ddt-root {
        type inet:ip-address;
      }
    }
    container alt-mapping-system {
      presence "ALT Mapping System in use";
    }
  }
}

```

Internet-Draft

LISP-YANG

March 2015

```
    }
  }
  container proxy-itr-cfg {
    if-feature proxy-itr;
    presence "LISP PITR operation enabled";
    config true;
    container servicing-eids {
      list eid {
        key "id";
        leaf id {
          type eid-id;
        }
        container eid-address {
          uses lcaf:lisp-address;
        }
      }
    }
  }
  container map-resolvers {
    list map-resolver {
      key "id";
      leaf id {
        type eid-id;
      }
      container eid-address {
        uses lcaf:lisp-address;
      }
      leaf-list map-resolver {
        min-elements 1;
        type inet:ip-address;
      }
    }
  }
  container map-cache {
    uses mappings;
  }
}
container proxy-etr-cfg {
  if-feature proxy-etr;
  presence "LISP PETR operation enabled";
  config true;
  container servicing-eids {
    list eid {
```

```
key "id";
leaf id {
    type eid-id;
}
container eid-address {
    uses lcaf:lisp-address;
```

```
    }
  }
}
}
```

<CODE ENDS>

[4.](#) Acknowledgments

The tree view and the YANG model shown in this document have been formatted with the 'pyang' tool.

[5.](#) IANA Considerations

This memo includes no request to IANA.

[6.](#) Security Considerations

Security Considerations TBD

[7.](#) Normative References

[I-D.ietf-lisp-ddt]

Fuller, V., Lewis, D., Ermagan, V., and A. Jain, "LISP Delegated Database Tree", [draft-ietf-lisp-ddt-02](#) (work in progress), October 2014.

[I-D.ietf-lisp-lcaf]

Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format (LCAF)", [draft-ietf-lisp-lcaf-04](#) (work in progress), January 2014.

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

- [RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", [RFC 6830](#), January 2013.
- [RFC6833] Fuller, V. and D. Farinacci, "Locator/ID Separation Protocol (LISP) Map-Server Interface", [RFC 6833](#), January 2013.
- [RFC6836] Fuller, V., Farinacci, D., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol Alternative Logical Topology (LISP+ALT)", [RFC 6836](#), January 2013.

Ermagan, et al.

Expires September 24, 2015

[Page 25]

Internet-Draft

LISP-YANG

March 2015

Authors' Addresses

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

Email: vermagan@cisco.com

Alberto Rodriguez-Natal
Technical University of Catalonia
Barcelona
Spain

Email: arnatal@ac.upc.edu

Florin Coras
Technical University of Catalonia
Barcelona
Spain

Email: fcoras@ac.upc.edu

Albert Cabellos-Aparicio

Technical University of Catalonia
Barcelona
Spain

Email: acabello@ac.upc.edu

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

Email: fmaino@cisco.com