Network Working Group
Internet-Draft
Intended status: Experimental
Expires: June 19, 2016

V. Ermagan
Cisco Systems
A. Rodriguez-Natal
F. Coras
Technical University of Catalonia
C. Moberg
Cisco Systems
A. Cabellos-Aparicio
Technical University of Catalonia
F. Maino
Cisco Systems
December 17, 2015

LISP Configuration YANG Model
draft-ietf-lisp-yang-01

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 June 19, 2016.

Copyright Notice

Table of Contents

1. Introduction

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

[RFC6830]elements. The models also capture some essential operational data elements as well.

2. LISP Module

This module is the base LISP module that is augmented in multiple models to represent various LISP device roles.

2.1. Module Structure

```
module: ietf-lisp
   +--rw lisp
   |  +--rw devices
   |     +--rw device* [type id]
   |        +--rw type       device-ref
   |        +--rw id         string
   |        +--rw enabled?   boolean
   +--ro lisp-state
      +--ro devices
         +--ro device* [type id]
            +--ro type    device-ref
            +--ro id      string
```

2.2. Module Definition

```
<CODE BEGINS> file "ietf-lisp@2014-12-19.yang"
module ietf-lisp {
  namespace "urn:ietf:params:xml:ns:yang:ietf-lisp";
  prefix lisp;
  import ietf-interfaces {
    prefix if;
  }
  import ietf-lisp-address-types {
    prefix lcaf;
  }
  organization
    "IETF LISP (Locator/ID Separation Protocol) Working Group";
  contact
    "lisp@ietf.org";
```

```
  description
    "This YANG module defines the generic configuration
     and operational data for LISP. The module can be extended by
     vendors to define vendor-specific LISP configuration
     parameters and policies.

     Copyright (c) 2014 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 2014-12-19 {
    description
      "Initial revision.";
    reference
      "https://tools.ietf.org/html/rfc6830";
  }
  identity lisp-device {
    description
      "LISP network device.";
  }
  typedef device-ref {
    type identityref {
      base lisp-device;
    }
    description
      "LISP device reference";
  }
  typedef map-reply-action {
    type enumeration {
      enum no-action {
        value 0;
```

```
          description
            "Mapping is kept alive and no encapsulation occurs.";
        }
        enum natively-forward {
          value 1;
          description
            "Matching packets are not encapsulated or dropped but
             natively forwarded.";
        }
        enum send-map-request {
          value 2;
          description
            "Matching packets invoke Map-Requests.";
        }
        enum drop {
          value 3;
          description
            "Matching packets are dropped.";
        }
```



```
      }
      description
        "Defines the lisp map-cache ACT type";
      reference "https://tools.ietf.org/html/rfc6830#section-6.1.4";
    }
    typedef eid-id {
      type string;
      description
        "Type encoding of lisp-addresses to be generally used in EID
         keyed lists.";
    }
    typedef auth-key-type {
      type enumeration {
        enum none {
          value 0;
          description
            "No authentication.";
        }
        enum hmac-sha-1-96 {
          value 1;
          description
            "HMAC-SHA-1-96 (RFC2404) authentication is used.";
```

```
        }
        enum hmac-sha-256-128 {
          value 2;
          description
            "HMAC-SHA-256-128 (RFC4868) authentication is used.";
        }
      }
      description
        "Enumeration of the authentication mechansms supported by
         LISP.";
      reference
        "https://tools.ietf.org/html/rfc6830#section-6.1.6";
    }
    grouping locators {
      description
        "Group that defines a list of LISP locators.";
      list rloc {
        key "id";
        description
          "List of routing locators";
        leaf id {
          type string;
          description
            "Locator id";
        }
        // FC need to be sure we don't use interface-name in itr
```



```
        // cached mappings
        choice address-type {
          description
            "The address type of the locator";
          case interface-name {
            leaf interface {
              type if:interface-state-ref;
              description
                "The name of the interface supporting the locator.";
            }
          }
          case address {
            container locator-address {
              uses lcaf:lisp-address;
              description
```

```
                "The locator address provided in LISP canonincal
                 address format.";
            }
          }
        }
        leaf priority {
          type uint8;
          description
            "Locator priority.";
        }
        leaf weight {
          type uint8;
          description
            "Locator weight.";
        }
        leaf multicast-priority {
          type uint8;
          description
            "Locator's multicast priority";
        }
        leaf multicast-weight {
          type uint8;
          description
            "Locator's multicast weight";
        }
      }
    }
    grouping mappings {
      description
        "Group that defines a list of LISP mappings.";
      list mapping {
        key "id";
        description
```



```
          "List of EID to RLOCs mappings.";
        leaf id {
          type eid-id;
          description
            "Id that uniquely identifies a mapping.";
        }
        container eid {
          uses lcaf:lisp-address;
```

```
        description
          "End-host Identifier (EID) to be mapped to a list of
           locators";
      }
      leaf ttl {
        type uint32;
        description
          "Mapping validity period.";
      }
      leaf authoritative {
        type bits {
          bit A {
            description
              "Authoritative bit.";
          }
        }
        description
          "Bit that indicates if mapping comes from an
           authoritative source.";
      }
      choice locator-list {
        case negative-mapping {
          leaf map-reply-action {
            type map-reply-action;
            description
              "Forwarding action for a negative mapping.";
          }
        }
        case positive-mapping {
          container rlocs {
            uses locators;
            description
              "List of locators for a positive mapping.";
          }
        }
        default "positive-mapping";
        description
          "Choice of locator list based on type of mapping.";
      }
    }
```



```
}
```

```
/* Configuration Data */
container lisp {
  description
    "Configuration parameters for LISP subsystem.";
  container devices {
    description
      "Configuration of LISP devices.";
    list device {
      key "type id";
      description
        "Each entry contains configuration of a lisp-device.";
      leaf type {
        type device-ref;
        description
          "The type of LISP device - identity derived from the
           'lisp-device' base identity.";
      }
      leaf id {
        type string;
        description
          "Arbitrary device name.";
      }
      leaf enabled {
        type boolean;
        default "true";
        description
          "Enable/disable the lisp-device.";
      }
    }
  }
}

/* Operational state data */
container lisp-state {
  config false;
  description
    "Operational state of the LISP subsystem.";
  container devices {
    description
      "Operational state of lisp-devices.";
    list device {
      key "type id";
      description
        "Each entry contains operational data of a lisp-device.";
      leaf type {
        type device-ref;
        description
```

```
              "Type of LISP device.";
          }
          leaf id {
            type string;
            description
              "Name of LISP device.";
          }
        }
      }
    }

  }
  <CODE ENDS>
```

3. LISP-ITR Module

This module captures the configuration data model of a LISP ITR. The model also captures some operational data elements.

3.1. Module Structure

```
module: ietf-lisp-itr
augment /lisp:lisp/lisp:devices/lisp:device:
   +--rw itr-cfg!
      +--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                    lisp-address-family-ref
      |        +--rw instance-id?           instance-id-type
      |        +--rw (address)?
      |           +--:(ipv4)
      |           |  +--rw ipv4?                  inet:ipv4-address
      |           +--:(ipv6)
      |           |  +--rw ipv6?                  inet:ipv6-address
      |           +--:(mac)
      |           |  +--rw mac?                   yang:mac-address
      |           +--:(distinguished-name)
      |           |  +--rw distinguished-name?    distinguished-name
      |           +--:(lcaf)
      |              +--rw lcaf
      |                 +--rw lcaf-type                lcaf-ref
```

```
|                  +--rw (address)?
|                     +--:(null-address)
```



```
|                     |  +--rw null-address
|                     |     +--rw address?   empty
|                     +--:(afi-list)
|                     |  +--rw afi-list
|                     |     +--rw address-list*   simple-address
|                     +--:(instance-id)
|                     |  +--rw instance-id
|                     |     +--rw instance-id?   instance-id-type
|                     |     +--rw mask-length?   uint8
|                     |     +--rw address?       simple-address
|                     +--:(as-number)
|                     |  +--rw as-number
|                     |     +--rw as?        inet:as-number
|                     |     +--rw address?   simple-address
|                     +--:(application-data)
|                     |  +--rw application-data
|                     |     +--rw address?            simple-address
|                     |     +--rw protocol?           uint8
|                     |     +--rw ip-tos?             int32
|                     |     +--rw local-port-low?     inet:port-number
|                     |     +--rw local-port-high?    inet:port-number
|                     |     +--rw remote-port-low?    inet:port-number
|                     |     +--rw remote-port-high?   inet:port-number
|                     +--:(geo-coordinates)
|                     |  +--rw geo-coordinates
|                     |     +--rw latitude?            bits
|                     |     +--rw latitude-degrees?    uint8
|                     |     +--rw latitude-minutes?    uint8
|                     |     +--rw latitude-seconds?    uint8
|                     |     +--rw longitude?           bits
|                     |     +--rw longitude-degrees?   uint16
|                     |     +--rw longitude-minutes?   uint8
|                     |     +--rw longitude-seconds?   uint8
|                     |     +--rw altitude?            int32
|                     |     +--rw address?             simple-address
|                     +--:(nat-traversal)
|                     |  +--rw nat-traversal
|                     |     +--rw ms-udp-port?        uint16
|                     |     +--rw etr-udp-port?       uint16
```

```
|                    |     +--rw global-etr-rloc?    simple-address
|                    |     +--rw ms-rloc?            simple-address
|                    |     +--rw private-etr-rloc?   simple-address
|                    |     +--rw rtr-rlocs*          simple-address
|                    |  +--:(explicit-locator-path)
|                    |  |  +--rw explicit-locator-path
|                    |  |     +--rw hop* [address]
|                    |  |        +--rw address     simple-address
|                    |  |        +--rw lrs-bits?   bits
```



```
|                    +--:(sourc-dest-key)
|                       +--rw sourc-dest-key
|                          +--rw source?   inet:ip-prefix
|                          +--rw dest?     inet:ip-prefix
|  +--rw local-eids
|  |  +--rw local-eid* [id]
|  |     +--rw id              lisp:eid-id
|  |     +--rw eid-address
|  |        +--rw afi                    lisp-address-family-ref
|  |        +--rw instance-id?           instance-id-type
|  |        +--rw (address)?
|  |           +--:(ipv4)
|  |           |  +--rw ipv4?                 inet:ipv4-address
|  |           +--:(ipv6)
|  |           |  +--rw ipv6?                 inet:ipv6-address
|  |           +--:(mac)
|  |           |  +--rw mac?                  yang:mac-address
|  |           +--:(distinguished-name)
|  |           |  +--rw distinguished-name?   distinguished-name
|  |           +--:(lcaf)
|  |              +--rw lcaf
|  |                 +--rw lcaf-type                lcaf-ref
|  |                 +--rw (address)?
|  |                    +--:(null-address)
|  |                    |  +--rw null-address
|  |                    |     +--rw address?   empty
|  |                    +--:(afi-list)
|  |                    |  +--rw afi-list
|  |                    |     +--rw address-list*   simple-address
|  |                    +--:(instance-id)
|  |                    |  +--rw instance-id
|  |                    |     +--rw instance-id?   instance-id-type
```

```
|                    |     +--rw mask-length?   uint8
|                    |     +--rw address?       simple-address
|                    +--:(as-number)
|                    |  +--rw as-number
|                    |     +--rw as?        inet:as-number
|                    |     +--rw address?   simple-address
|                    +--:(application-data)
|                    |  +--rw application-data
|                    |     +--rw address?            simple-address
|                    |     +--rw protocol?           uint8
|                    |     +--rw ip-tos?             int32
|                    |     +--rw local-port-low?     inet:port-number
|                    |     +--rw local-port-high?    inet:port-number
|                    |     +--rw remote-port-low?    inet:port-number
|                    |     +--rw remote-port-high?   inet:port-number
|                    +--:(geo-coordinates)
```



```
|                    |  +--rw geo-coordinates
|                    |     +--rw latitude?            bits
|                    |     +--rw latitude-degrees?    uint8
|                    |     +--rw latitude-minutes?    uint8
|                    |     +--rw latitude-seconds?    uint8
|                    |     +--rw longitude?           bits
|                    |     +--rw longitude-degrees?   uint16
|                    |     +--rw longitude-minutes?   uint8
|                    |     +--rw longitude-seconds?   uint8
|                    |     +--rw altitude?            int32
|                    |     +--rw address?             simple-address
|                    +--:(nat-traversal)
|                    |  +--rw nat-traversal
|                    |     +--rw ms-udp-port?        uint16
|                    |     +--rw etr-udp-port?       uint16
|                    |     +--rw global-etr-rloc?    simple-address
|                    |     +--rw ms-rloc?            simple-address
|                    |     +--rw private-etr-rloc?   simple-address
|                    |     +--rw rtr-rlocs*          simple-address
|                    +--:(explicit-locator-path)
|                    |  +--rw explicit-locator-path
|                    |     +--rw hop* [address]
|                    |        +--rw address     simple-address
|                    |        +--rw lrs-bits?   bits
|                    +--:(sourc-dest-key)
```

```
|                        +--rw sourc-dest-key
|                           +--rw source?   inet:ip-prefix
|                           +--rw dest?     inet:ip-prefix
+--rw map-resolvers
|  +--rw map-resolver* [id]
|     +--rw id                        lisp:eid-id
|     +--rw eid-address
|     |  +--rw afi                     lisp-address-family-ref
|     |  +--rw instance-id?            instance-id-type
|     |  +--rw (address)?
|     |     +--:(ipv4)
|     |     |  +--rw ipv4?                 inet:ipv4-address
|     |     +--:(ipv6)
|     |     |  +--rw ipv6?                 inet:ipv6-address
|     |     +--:(mac)
|     |     |  +--rw mac?                  yang:mac-address
|     |     +--:(distinguished-name)
|     |     |  +--rw distinguished-name?   distinguished-name
|     |     +--:(lcaf)
|     |        +--rw lcaf
|     |           +--rw lcaf-type               lcaf-ref
|     |           +--rw (address)?
|     |              +--:(null-address)
```



```
|     |              |  +--rw null-address
|     |              |     +--rw address?   empty
|     |              +--:(afi-list)
|     |              |  +--rw afi-list
|     |              |     +--rw address-list*   simple-address
|     |              +--:(instance-id)
|     |              |  +--rw instance-id
|     |              |     +--rw instance-id?   instance-id-type
|     |              |     +--rw mask-length?   uint8
|     |              |     +--rw address?       simple-address
|     |              +--:(as-number)
|     |              |  +--rw as-number
|     |              |     +--rw as?        inet:as-number
|     |              |     +--rw address?   simple-address
|     |              +--:(application-data)
|     |              |  +--rw application-data
|     |              |     +--rw address?            simple-address
|     |              |     +--rw protocol?           uint8
```

```
|     |              |     +--rw ip-tos?             int32
|     |              |     +--rw local-port-low?     inet:port-number
|     |              |     +--rw local-port-high?    inet:port-number
|     |              |     +--rw remote-port-low?    inet:port-number
|     |              |     +--rw remote-port-high?   inet:port-number
|     |              |  +--:(geo-coordinates)
|     |              |  |  +--rw geo-coordinates
|     |              |     +--rw latitude?            bits
|     |              |     +--rw latitude-degrees?    uint8
|     |              |     +--rw latitude-minutes?    uint8
|     |              |     +--rw latitude-seconds?    uint8
|     |              |     +--rw longitude?           bits
|     |              |     +--rw longitude-degrees?   uint16
|     |              |     +--rw longitude-minutes?   uint8
|     |              |     +--rw longitude-seconds?   uint8
|     |              |     +--rw altitude?            int32
|     |              |     +--rw address?             simple-address
|     |              |  +--:(nat-traversal)
|     |              |  |  +--rw nat-traversal
|     |              |     +--rw ms-udp-port?        uint16
|     |              |     +--rw etr-udp-port?       uint16
|     |              |     +--rw global-etr-rloc?    simple-address
|     |              |     +--rw ms-rloc?            simple-address
|     |              |     +--rw private-etr-rloc?   simple-address
|     |              |     +--rw rtr-rlocs*          simple-address
|     |              |  +--:(explicit-locator-path)
|     |              |  |  +--rw explicit-locator-path
|     |              |     +--rw hop* [address]
|     |              |        +--rw address     simple-address
|     |              |        +--rw lrs-bits?   bits
```



```
|     |              +--:(sourc-dest-key)
|     |                 +--rw sourc-dest-key
|     |                    +--rw source?   inet:ip-prefix
|     |                    +--rw dest?     inet:ip-prefix
|     +--rw map-resolver-address*   inet:ip-address
+--rw proxy-etrs
|  +--rw proxy-etr* [id]
|     +--rw id                    lisp:eid-id
|     +--rw eid-address
|     |  +--rw afi                    lisp-address-family-ref
|     |  +--rw instance-id?           instance-id-type
```

```
|      |  +--rw (address)?
|      |     +--:(ipv4)
|      |     |  +--rw ipv4?                inet:ipv4-address
|      |     +--:(ipv6)
|      |     |  +--rw ipv6?                inet:ipv6-address
|      |     +--:(mac)
|      |     |  +--rw mac?                 yang:mac-address
|      |     +--:(distinguished-name)
|      |     |  +--rw distinguished-name?  distinguished-name
|      |     +--:(lcaf)
|      |        +--rw lcaf
|      |           +--rw lcaf-type               lcaf-ref
|      |           +--rw (address)?
|      |              +--:(null-address)
|      |              |  +--rw null-address
|      |              |     +--rw address?   empty
|      |              +--:(afi-list)
|      |              |  +--rw afi-list
|      |              |     +--rw address-list*   simple-address
|      |              +--:(instance-id)
|      |              |  +--rw instance-id
|      |              |     +--rw instance-id?   instance-id-type
|      |              |     +--rw mask-length?   uint8
|      |              |     +--rw address?       simple-address
|      |              +--:(as-number)
|      |              |  +--rw as-number
|      |              |     +--rw as?        inet:as-number
|      |              |     +--rw address?   simple-address
|      |              +--:(application-data)
|      |              |  +--rw application-data
|      |              |     +--rw address?            simple-address
|      |              |     +--rw protocol?           uint8
|      |              |     +--rw ip-tos?             int32
|      |              |     +--rw local-port-low?     inet:port-number
|      |              |     +--rw local-port-high?    inet:port-number
|      |              |     +--rw remote-port-low?    inet:port-number
|      |              |     +--rw remote-port-high?   inet:port-number
```



```
|      |              +--:(geo-coordinates)
|      |              |  +--rw geo-coordinates
|      |              |     +--rw latitude?            bits
|      |              |     +--rw latitude-degrees?    uint8
```

```
|      |              |     +--rw latitude-minutes?    uint8
|      |              |     +--rw latitude-seconds?    uint8
|      |              |     +--rw longitude?           bits
|      |              |     +--rw longitude-degrees?   uint16
|      |              |     +--rw longitude-minutes?   uint8
|      |              |     +--rw longitude-seconds?   uint8
|      |              |     +--rw altitude?            int32
|      |              |     +--rw address?             simple-address
|      |              +--:(nat-traversal)
|      |              |  +--rw nat-traversal
|      |              |     +--rw ms-udp-port?        uint16
|      |              |     +--rw etr-udp-port?       uint16
|      |              |     +--rw global-etr-rloc?    simple-address
|      |              |     +--rw ms-rloc?            simple-address
|      |              |     +--rw private-etr-rloc?   simple-address
|      |              |     +--rw rtr-rlocs*          simple-address
|      |              +--:(explicit-locator-path)
|      |              |  +--rw explicit-locator-path
|      |              |     +--rw hop* [address]
|      |              |        +--rw address     simple-address
|      |              |        +--rw lrs-bits?   bits
|      |              +--:(sourc-dest-key)
|      |                 +--rw sourc-dest-key
|      |                    +--rw source?   inet:ip-prefix
|      |                    +--rw dest?     inet:ip-prefix
|      +--rw proxy-etr-address*   inet:ip-address
+--rw static-mappings
   +--rw mapping* [id]
      +--rw id                  eid-id
      +--rw eid
      |  +--rw afi                   lisp-address-family-ref
      |  +--rw instance-id?          instance-id-type
      |  +--rw (address)?
      |     +--:(ipv4)
      |     |  +--rw ipv4?                inet:ipv4-address
      |     +--:(ipv6)
      |     |  +--rw ipv6?                inet:ipv6-address
      |     +--:(mac)
      |     |  +--rw mac?                 yang:mac-address
      |     +--:(distinguished-name)
      |     |  +--rw distinguished-name?   distinguished-name
      |     +--:(lcaf)
      |        +--rw lcaf
      |           +--rw lcaf-type               lcaf-ref
```

```
|            +--rw (address)?
|               +--:(null-address)
|               |  +--rw null-address
|               |     +--rw address?   empty
|               +--:(afi-list)
|               |  +--rw afi-list
|               |     +--rw address-list*   simple-address
|               +--:(instance-id)
|               |  +--rw instance-id
|               |     +--rw instance-id?   instance-id-type
|               |     +--rw mask-length?   uint8
|               |     +--rw address?       simple-address
|               +--:(as-number)
|               |  +--rw as-number
|               |     +--rw as?        inet:as-number
|               |     +--rw address?   simple-address
|               +--:(application-data)
|               |  +--rw application-data
|               |     +--rw address?            simple-address
|               |     +--rw protocol?           uint8
|               |     +--rw ip-tos?             int32
|               |     +--rw local-port-low?     inet:port-number
|               |     +--rw local-port-high?    inet:port-number
|               |     +--rw remote-port-low?    inet:port-number
|               |     +--rw remote-port-high?   inet:port-number
|               +--:(geo-coordinates)
|               |  +--rw geo-coordinates
|               |     +--rw latitude?            bits
|               |     +--rw latitude-degrees?    uint8
|               |     +--rw latitude-minutes?    uint8
|               |     +--rw latitude-seconds?    uint8
|               |     +--rw longitude?           bits
|               |     +--rw longitude-degrees?   uint16
|               |     +--rw longitude-minutes?   uint8
|               |     +--rw longitude-seconds?   uint8
|               |     +--rw altitude?            int32
|               |     +--rw address?             simple-address
|               +--:(nat-traversal)
|               |  +--rw nat-traversal
|               |     +--rw ms-udp-port?        uint16
|               |     +--rw etr-udp-port?       uint16
|               |     +--rw global-etr-rloc?    simple-address
|               |     +--rw ms-rloc?            simple-address
|               |     +--rw private-etr-rloc?   simple-address
|               |     +--rw rtr-rlocs*          simple-address
|               +--:(explicit-locator-path)
|               |  +--rw explicit-locator-path
```

```
|              |     +--rw hop* [address]
```



```
|              |        +--rw address     simple-address
|              |        +--rw lrs-bits?   bits
|              +--:(sourc-dest-key)
|                 +--rw sourc-dest-key
|                    +--rw source?   inet:ip-prefix
|                    +--rw dest?     inet:ip-prefix
+--rw ttl?                uint32
+--rw authoritative?      bits
+--rw (locator-list)?
   +--:(negative-mapping)
   |  +--rw map-reply-action?   map-reply-action
   +--:(positive-mapping)
      +--rw rlocs
         +--rw rloc* [id]
            +--rw id                      string
            +--rw (address-type)?
            |  +--:(interface-name)
            |  |  +--rw interface?            if:interface-state-re
            |  +--:(address)
            |     +--rw locator-address
            |        +--rw afi                    lisp-address-famil
            |        +--rw instance-id?           instance-id-type
            |        +--rw (address)?
            |           +--:(ipv4)
            |           |  +--rw ipv4?                inet:ipv4-ad
            |           +--:(ipv6)
            |           |  +--rw ipv6?                inet:ipv6-ad
            |           +--:(mac)
            |           |  +--rw mac?                 yang:mac-add
            |           +--:(distinguished-name)
            |           |  +--rw distinguished-name?   distinguishe
            |           +--:(lcaf)
            |              +--rw lcaf
            |                 +--rw lcaf-type               lcaf-r
            |                 +--rw (address)?
            |                    +--:(null-address)
            |                    |  +--rw null-address
            |                    |     +--rw address?   empty
            |                    +--:(afi-list)
            |                    |  +--rw afi-list
```

```
|                      |     +--rw address-list*   simple
|                      +--:(instance-id)
|                      |  +--rw instance-id
|                      |     +--rw instance-id?   instanc
|                      |     +--rw mask-length?   uint8
|                      |     +--rw address?       simple-
|                      +--:(as-number)
|                      |  +--rw as-number
```



```
|                      |     +--rw as?        inet:as-num
|                      |     +--rw address?   simple-addr
|                      +--:(application-data)
|                      |  +--rw application-data
|                      |     +--rw address?            si
|                      |     +--rw protocol?           ui
|                      |     +--rw ip-tos?             in
|                      |     +--rw local-port-low?     in
|                      |     +--rw local-port-high?    in
|                      |     +--rw remote-port-low?    in
|                      |     +--rw remote-port-high?   in
|                      +--:(geo-coordinates)
|                      |  +--rw geo-coordinates
|                      |     +--rw latitude?            b
|                      |     +--rw latitude-degrees?    u
|                      |     +--rw latitude-minutes?    u
|                      |     +--rw latitude-seconds?    u
|                      |     +--rw longitude?           b
|                      |     +--rw longitude-degrees?   u
|                      |     +--rw longitude-minutes?   u
|                      |     +--rw longitude-seconds?   u
|                      |     +--rw altitude?            i
|                      |     +--rw address?             s
|                      +--:(nat-traversal)
|                      |  +--rw nat-traversal
|                      |     +--rw ms-udp-port?        ui
|                      |     +--rw etr-udp-port?       ui
|                      |     +--rw global-etr-rloc?    si
|                      |     +--rw ms-rloc?            si
|                      |     +--rw private-etr-rloc?   si
|                      |     +--rw rtr-rlocs*          si
|                      +--:(explicit-locator-path)
|                      |  +--rw explicit-locator-path
```

```
                        |                     |     +--rw hop* [address]
                        |                     |        +--rw address     simple-
                        |                     |        +--rw lrs-bits?   bits
                        |                     +--:(sourc-dest-key)
                        |                        +--rw sourc-dest-key
                        |                           +--rw source?   inet:ip-pref
                        |                           +--rw dest?     inet:ip-pref
                        +--rw priority?             uint8
                        +--rw weight?               uint8
                        +--rw multicast-priority?   uint8
                        +--rw multicast-weight?     uint8
augment /lisp:lisp-state/lisp:devices/lisp:device:
   +--ro itr-state
      +--ro learned-mappings
         +--ro mapping* [id]
```



```
            +--ro id                  eid-id
            +--ro eid
            |  +--ro afi                   lisp-address-family-ref
            |  +--ro instance-id?          instance-id-type
            |  +--ro (address)?
            |     +--:(ipv4)
            |     |  +--ro ipv4?                inet:ipv4-address
            |     +--:(ipv6)
            |     |  +--ro ipv6?                inet:ipv6-address
            |     +--:(mac)
            |     |  +--ro mac?                 yang:mac-address
            |     +--:(distinguished-name)
            |     |  +--ro distinguished-name?  distinguished-name
            |     +--:(lcaf)
            |        +--ro lcaf
            |           +--ro lcaf-type               lcaf-ref
            |           +--ro (address)?
            |              +--:(null-address)
            |              |  +--ro null-address
            |              |     +--ro address?   empty
            |              +--:(afi-list)
            |              |  +--ro afi-list
            |              |     +--ro address-list*   simple-address
            |              +--:(instance-id)
            |              |  +--ro instance-id
            |              |     +--ro instance-id?   instance-id-type
```

```
|              |     +--ro mask-length?   uint8
|              |     +--ro address?       simple-address
|              +--:(as-number)
|              |  +--ro as-number
|              |     +--ro as?        inet:as-number
|              |     +--ro address?   simple-address
|              +--:(application-data)
|              |  +--ro application-data
|              |     +--ro address?            simple-address
|              |     +--ro protocol?           uint8
|              |     +--ro ip-tos?             int32
|              |     +--ro local-port-low?     inet:port-number
|              |     +--ro local-port-high?    inet:port-number
|              |     +--ro remote-port-low?    inet:port-number
|              |     +--ro remote-port-high?   inet:port-number
|              +--:(geo-coordinates)
|              |  +--ro geo-coordinates
|              |     +--ro latitude?            bits
|              |     +--ro latitude-degrees?    uint8
|              |     +--ro latitude-minutes?    uint8
|              |     +--ro latitude-seconds?    uint8
|              |     +--ro longitude?           bits
```



```
|              |     +--ro longitude-degrees?   uint16
|              |     +--ro longitude-minutes?   uint8
|              |     +--ro longitude-seconds?   uint8
|              |     +--ro altitude?            int32
|              |     +--ro address?             simple-address
|              +--:(nat-traversal)
|              |  +--ro nat-traversal
|              |     +--ro ms-udp-port?        uint16
|              |     +--ro etr-udp-port?       uint16
|              |     +--ro global-etr-rloc?    simple-address
|              |     +--ro ms-rloc?            simple-address
|              |     +--ro private-etr-rloc?   simple-address
|              |     +--ro rtr-rlocs*          simple-address
|              +--:(explicit-locator-path)
|              |  +--ro explicit-locator-path
|              |     +--ro hop* [address]
|              |        +--ro address     simple-address
|              |        +--ro lrs-bits?   bits
|              +--:(sourc-dest-key)
```

```
|                 +--ro sourc-dest-key
|                    +--ro source?   inet:ip-prefix
|                    +--ro dest?     inet:ip-prefix
+--ro ttl?               uint32
+--ro authoritative?     bits
+--ro (locator-list)?
   +--:(negative-mapping)
   |  +--ro map-reply-action?   map-reply-action
   +--:(positive-mapping)
      +--ro rlocs
         +--ro rloc* [id]
            +--ro id                     string
            +--ro (address-type)?
            |  +--:(interface-name)
            |  |  +--ro interface?            if:interface-state-re
            |  +--:(address)
            |     +--ro locator-address
            |        +--ro afi                    lisp-address-famil
            |        +--ro instance-id?           instance-id-type
            |        +--ro (address)?
            |           +--:(ipv4)
            |           |  +--ro ipv4?                inet:ipv4-ad
            |           +--:(ipv6)
            |           |  +--ro ipv6?                inet:ipv6-ad
            |           +--:(mac)
            |           |  +--ro mac?                 yang:mac-add
            |           +--:(distinguished-name)
            |           |  +--ro distinguished-name?   distinguishe
            |           +--:(lcaf)
```



```
                     |              +--ro lcaf
                     |                 +--ro lcaf-type               lcaf-r
                     |                 +--ro (address)?
                     |                    +--:(null-address)
                     |                    |  +--ro null-address
                     |                    |     +--ro address?   empty
                     |                    +--:(afi-list)
                     |                    |  +--ro afi-list
                     |                    |     +--ro address-list*   simple
                     |                    +--:(instance-id)
                     |                    |  +--ro instance-id
                     |                    |     +--ro instance-id?   instanc
```

```
|                      |     +--ro mask-length?   uint8
|                      |     +--ro address?       simple-
|                      +--:(as-number)
|                      |  +--ro as-number
|                      |     +--ro as?        inet:as-num
|                      |     +--ro address?   simple-addr
|                      +--:(application-data)
|                      |  +--ro application-data
|                      |     +--ro address?            si
|                      |     +--ro protocol?           ui
|                      |     +--ro ip-tos?             in
|                      |     +--ro local-port-low?     in
|                      |     +--ro local-port-high?    in
|                      |     +--ro remote-port-low?    in
|                      |     +--ro remote-port-high?   in
|                      +--:(geo-coordinates)
|                      |  +--ro geo-coordinates
|                      |     +--ro latitude?            b
|                      |     +--ro latitude-degrees?    u
|                      |     +--ro latitude-minutes?    u
|                      |     +--ro latitude-seconds?    u
|                      |     +--ro longitude?           b
|                      |     +--ro longitude-degrees?   u
|                      |     +--ro longitude-minutes?   u
|                      |     +--ro longitude-seconds?   u
|                      |     +--ro altitude?            i
|                      |     +--ro address?             s
|                      +--:(nat-traversal)
|                      |  +--ro nat-traversal
|                      |     +--ro ms-udp-port?        ui
|                      |     +--ro etr-udp-port?       ui
|                      |     +--ro global-etr-rloc?    si
|                      |     +--ro ms-rloc?            si
|                      |     +--ro private-etr-rloc?   si
|                      |     +--ro rtr-rlocs*          si
|                      +--:(explicit-locator-path)
```



```
|                      |  +--ro explicit-locator-path
|                      |     +--ro hop* [address]
|                      |        +--ro address     simple-
|                      |        +--ro lrs-bits?   bits
|                      +--:(sourc-dest-key)
```

```
                        |                        +--ro sourc-dest-key
                        |                           +--ro source?   inet:ip-pref
                        |                           +--ro dest?     inet:ip-pref
                        +--ro priority?              uint8
                        +--ro weight?                uint8
                        +--ro multicast-priority?    uint8
                        +--ro multicast-weight?      uint8
```

3.2. Module Definition

```
<CODE BEGINS> file "ietf-lisp-itr@2015-07-02.yang"
module ietf-lisp-itr {
  namespace "urn:ietf:params:xml:ns:yang:ietf-lisp-itr";
  prefix lisp-itr;
  import ietf-lisp {
    prefix lisp;
  }
  import ietf-lisp-address-types {
   prefix lcaf;
  }
  import ietf-inet-types {
    prefix inet;
  }
  organization
   "IETF LISP (Locator/ID Separation Protocol) Working Group";
  contact
    "lisp@ietf.org";
  description
    "This YANG module defines the generic configuration
     data for a LISP ITR. The module can be extended by vendors
     to define vendor-specific configuration parameters and
     policies.

     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 2015-07-02 {
  description
    "Initial revision.";
  reference
    "https://tools.ietf.org/html/rfc6830";
}
identity itr {
  base lisp:lisp-device;
  description
    "LISP ITR.";
}
augment "/lisp:lisp/lisp:devices/lisp:device" {
  when "lisp:type = lisp-itr:itr" {
    description
      "Augment is valid when LISP device type is ITR.";
  }
  description
    "This augments LISP devices list with ITR specific
     parameters.";
  container itr-cfg {
    presence "LISP ITR operation enabled";
    description
      "ITR configuration";
    container rloc-probing {
      presence "RLOC probing active";
      description
        "RLOC-probing parameters";
      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";
            description
              "ITR's list of RLOCs.";
            leaf id {
              type string;
              description
                "Unique RLOC id.";
            }
            container address {
              uses lcaf:lisp-address;
              description
                "RLOC address in generic LISP address format.";
            }
          }
        }
        container local-eids {
          description
            "Container for an ITR's local list of EIDs";
          list local-eid {
            key "id";
            min-elements 1;
            description
              "List of EIDs from which the ITR forwards traffic.";
            leaf id {
              type lisp:eid-id;
              description
                "Unique EID ID";
            }
            container eid-address {
              uses lcaf:lisp-address;
              description
                "Addres in generic LISP address format";
            }
          }
        }
        container map-resolvers {
          description
            "The Map-Resolvers configured for the ITR.";
```

```
            list map-resolver {
              key "id";
              min-elements 1;
              description
                "List of Map-Resolvers configured for the ITR.";
              leaf id {
```



```
                type lisp:eid-id;
                description
                  "Id of the EID for which the Map-Server is
                   configured.";
              }
              container eid-address {
                uses lcaf:lisp-address;
                description
                  "EID for which the Map-Resolver is configured.";
              }
              leaf-list map-resolver-address {
                type inet:ip-address;
                description
                  "Map-Resolver RLOC address.";
              }
            }
          }
          container proxy-etrs {
            description
              "Proxy ETRs configured for the ITR.";
            list proxy-etr {
              key "id";
              description
                "List of PETR configured for the ITR.";
              leaf id {
                type lisp:eid-id;
                description
                  "Id of the EID for which the PETR is configured.";
              }
              container eid-address {
                uses lcaf:lisp-address;
                description
                  "EID for which the PETR is configured.";
              }
              leaf-list proxy-etr-address{
```

```
              type inet:ip-address;
              description
                "PETR RLOC address.";
            }
          }
        }
        container static-mappings {
          uses lisp:mappings;
          description
            "EID to RLOCs mappings cache.";
        }
      }
    }
```



```
    augment "/lisp:lisp-state/lisp:devices/lisp:device" {
      when "lisp:type = lisp-itr:itr" {
        description
          "Augment is valid when LISP device type is ITR.";
      }
      description
        "This augments LISP devices list state with ITR specific
         parameters.";
      container itr-state {
        config false;
        description
          "ITR state.";
        container learned-mappings {
          uses lisp:mappings;
          description
            "EID to RLOCs mappings cache.";
        }
      }
    }
  }
  <CODE ENDS>
```

4. LISP ETR Module

This module captures the configuration data model of a LISP ETR. The model also captures some operational data elements.

4.1. Module Structure

```
module: ietf-lisp-etr
augment /lisp:lisp/lisp:devices/lisp:device:
   +--rw etr-cfg!
      +--rw local-eids
         +--rw local-eid* [id]
            +--rw id                          lisp:eid-id
            +--rw eid-address
            |  +--rw afi                      lisp-address-family-ref
            |  +--rw instance-id?             instance-id-type
            |  +--rw (address)?
            |     +--:(ipv4)
            |     |  +--rw ipv4?                    inet:ipv4-address
            |     +--:(ipv6)
            |     |  +--rw ipv6?                    inet:ipv6-address
            |     +--:(mac)
            |     |  +--rw mac?                     yang:mac-address
            |     +--:(distinguished-name)
            |     |  +--rw distinguished-name?      distinguished-name
            |     +--:(lcaf)
```



```
            |        +--rw lcaf
            |           +--rw lcaf-type                lcaf-ref
            |           +--rw (address)?
            |              +--:(null-address)
            |              |  +--rw null-address
            |              |     +--rw address?   empty
            |              +--:(afi-list)
            |              |  +--rw afi-list
            |              |     +--rw address-list*   simple-address
            |              +--:(instance-id)
            |              |  +--rw instance-id
            |              |     +--rw instance-id?   instance-id-type
            |              |     +--rw mask-length?   uint8
            |              |     +--rw address?       simple-address
            |              +--:(as-number)
            |              |  +--rw as-number
            |              |     +--rw as?        inet:as-number
            |              |     +--rw address?   simple-address
            |              +--:(application-data)
            |              |  +--rw application-data
            |              |     +--rw address?            simple-address
```

```
|              |     +--rw protocol?           uint8
|              |     +--rw ip-tos?             int32
|              |     +--rw local-port-low?     inet:port-number
|              |     +--rw local-port-high?    inet:port-number
|              |     +--rw remote-port-low?    inet:port-number
|              |     +--rw remote-port-high?   inet:port-number
|              |  +--:(geo-coordinates)
|              |  |  +--rw geo-coordinates
|              |     +--rw latitude?            bits
|              |     +--rw latitude-degrees?    uint8
|              |     +--rw latitude-minutes?    uint8
|              |     +--rw latitude-seconds?    uint8
|              |     +--rw longitude?           bits
|              |     +--rw longitude-degrees?   uint16
|              |     +--rw longitude-minutes?   uint8
|              |     +--rw longitude-seconds?   uint8
|              |     +--rw altitude?            int32
|              |     +--rw address?             simple-address
|              |  +--:(nat-traversal)
|              |  |  +--rw nat-traversal
|              |     +--rw ms-udp-port?        uint16
|              |     +--rw etr-udp-port?       uint16
|              |     +--rw global-etr-rloc?    simple-address
|              |     +--rw ms-rloc?            simple-address
|              |     +--rw private-etr-rloc?   simple-address
|              |     +--rw rtr-rlocs*          simple-address
|              |  +--:(explicit-locator-path)
```



```
|              |  +--rw explicit-locator-path
|              |     +--rw hop* [address]
|              |        +--rw address     simple-address
|              |        +--rw lrs-bits?   bits
|              |  +--:(sourc-dest-key)
|              |     +--rw sourc-dest-key
|              |        +--rw source?   inet:ip-prefix
|              |        +--rw dest?     inet:ip-prefix
|           +--rw map-servers
|           |  +--rw map-server* [address]
|           |     +--rw address          inet:ip-address
|           |     +--rw auth-key?        string
|           |     +--rw auth-key-type?   lisp:auth-key-type
|           +--rw rlocs
```

```
|  +--rw rloc* [id]
|     +--rw id                      string
|     +--rw (address-type)?
|     |  +--:(interface-name)
|     |  |  +--rw interface?            if:interface-state-ref
|     |  +--:(address)
|     |     +--rw locator-address
|     |        +--rw afi                    lisp-address-family-ref
|     |        +--rw instance-id?           instance-id-type
|     |        +--rw (address)?
|     |           +--:(ipv4)
|     |           |  +--rw ipv4?                 inet:ipv4-address
|     |           +--:(ipv6)
|     |           |  +--rw ipv6?                 inet:ipv6-address
|     |           +--:(mac)
|     |           |  +--rw mac?                  yang:mac-address
|     |           +--:(distinguished-name)
|     |           |  +--rw distinguished-name?   distinguished-name
|     |           +--:(lcaf)
|     |              +--rw lcaf
|     |                 +--rw lcaf-type               lcaf-ref
|     |                 +--rw (address)?
|     |                    +--:(null-address)
|     |                    |  +--rw null-address
|     |                    |     +--rw address?   empty
|     |                    +--:(afi-list)
|     |                    |  +--rw afi-list
|     |                    |     +--rw address-list*   simple-addre
|     |                    +--:(instance-id)
|     |                    |  +--rw instance-id
|     |                    |     +--rw instance-id?   instance-id-t
|     |                    |     +--rw mask-length?   uint8
|     |                    |     +--rw address?       simple-addres
|     |                    +--:(as-number)
```



```
|     |                    |  +--rw as-number
|     |                    |     +--rw as?        inet:as-number
|     |                    |     +--rw address?   simple-address
|     |                    +--:(application-data)
|     |                    |  +--rw application-data
|     |                    |     +--rw address?            simple-a
|     |                    |     +--rw protocol?           uint8
```

```
|      |                        |     +--rw ip-tos?              int32
|      |                        |     +--rw local-port-low?      inet:por
|      |                        |     +--rw local-port-high?     inet:por
|      |                        |     +--rw remote-port-low?     inet:por
|      |                        |     +--rw remote-port-high?    inet:por
|      |                        +--:(geo-coordinates)
|      |                        |  +--rw geo-coordinates
|      |                        |     +--rw latitude?             bits
|      |                        |     +--rw latitude-degrees?     uint8
|      |                        |     +--rw latitude-minutes?     uint8
|      |                        |     +--rw latitude-seconds?     uint8
|      |                        |     +--rw longitude?            bits
|      |                        |     +--rw longitude-degrees?    uint16
|      |                        |     +--rw longitude-minutes?    uint8
|      |                        |     +--rw longitude-seconds?    uint8
|      |                        |     +--rw altitude?             int32
|      |                        |     +--rw address?              simple-
|      |                        +--:(nat-traversal)
|      |                        |  +--rw nat-traversal
|      |                        |     +--rw ms-udp-port?         uint16
|      |                        |     +--rw etr-udp-port?        uint16
|      |                        |     +--rw global-etr-rloc?     simple-a
|      |                        |     +--rw ms-rloc?             simple-a
|      |                        |     +--rw private-etr-rloc?    simple-a
|      |                        |     +--rw rtr-rlocs*           simple-a
|      |                        +--:(explicit-locator-path)
|      |                        |  +--rw explicit-locator-path
|      |                        |     +--rw hop* [address]
|      |                        |        +--rw address     simple-addres
|      |                        |        +--rw lrs-bits?   bits
|      |                        +--:(sourc-dest-key)
|      |                           +--rw sourc-dest-key
|      |                              +--rw source?   inet:ip-prefix
|      |                              +--rw dest?     inet:ip-prefix
|      +--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
```

4.2. Module Definition

```
<CODE BEGINS> file "ietf-lisp-etr@2015-07-02.yang"
module ietf-lisp-etr {
  namespace "urn:ietf:params:xml:ns:yang:ietf-lisp-etr";
  prefix lisp-etr;
  import ietf-lisp {
    prefix lisp;
  }
  import ietf-lisp-address-types {
   prefix lcaf;
  }
  import ietf-inet-types {
    prefix inet;
  }
  organization
   "IETF LISP (Locator/ID Separation Protocol) Working Group";
  contact
    "lisp@ietf.org";
  description
    "This YANG module defines the generic configuration
     data for a LISP ETR. The module can be extended by vendors
     to define vendor-specific configuration parameters and
     policies.

     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 2015-07-02 {
    description
      "Initial revision.";
    reference
      "https://tools.ietf.org/html/rfc6830";
  }
  identity etr {
    base lisp:lisp-device;
```

```
      description
        "LISP ETR.";
    }
    augment "/lisp:lisp/lisp:devices/lisp:device" {
      when "lisp:type = lisp-etr:etr" {
        description
          "Augment is valid when LISP device type is ETR.";
      }
      description
        "This augments LISP devices list with ETR specific
         parameters.";
      container etr-cfg {
        presence "LISP ETR operation enabled";
        description
          "ETR configuration parameters.";
        container local-eids {
          description
            "EIDs served by the ETR.";
          list local-eid {
            key "id";
            min-elements 1;
            description
              "List of local EIDs.";
            leaf id {
              type lisp:eid-id;
              description
                "Unique id of local EID.";
            }
            container eid-address {
              uses lcaf:lisp-address;
              description
                "EID address in generic LISP address format.";
            }
            container map-servers {
              description
                "Map-Servers configured for the ETR.";
              list map-server {
                key "address";
                description
                  "List of Map-Servers configured for the ETR.";
                leaf address {
                  type inet:ip-address;
                  description
```

```
                    "Map-Server address.";
                }
                leaf auth-key {
                  type string;
                  description
```



```
                    "Map-Server authentication key.";
                }
                leaf auth-key-type {
                  type lisp:auth-key-type;
                  description
                    "Map-Server authentication type.";
                }
              }
            }
            container rlocs {
              uses lisp:locators;
              description
                "Locators mapped to local EID.";
            }
            leaf record-ttl {
              type uint32;
              description
                "Validity period of the EID to RLOCs mapping provided
                 in Map-Replies.";
            }
            leaf want-map-notify {
              type boolean;
              description
                "Flag which if set in a Map-Register requests that a
                 Map-Notify be sent in response.";
            }
            leaf proxy-reply {
              type boolean;
              description
                "Flag which if set in a Map-Register requests that the
                 Map-Server proxy Map-Replies for the ETR.";
            }
            leaf registration-interval {
              type uint16;
              units "seconds";
              default "60";
```

```
                description
                  "Interval between consecutive Map-Register messages.";
              }
            }
          }
        }
      }
    }
    <CODE ENDS>
```


5. LISP Map Server Module

This module captures the configuration data model of a LISP Map Server [RFC6833]. The model also captures some operational data elements.

5.1. Module Structure

```
module: ietf-lisp-mapserver
augment /lisp:lisp/lisp:devices/lisp:device:
   +--rw map-server-cfg!
      +--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?    lisp:auth-key-type
      |           +--rw eids
      |              +--rw eid* [id]
      |                 +--rw id                              lisp:eid-id
      |                 +--rw eid-address
      |                 |  +--rw afi                     lisp-address-family-ref
      |                 |  +--rw instance-id?            instance-id-type
      |                 |  +--rw (address)?
      |                 |     +--:(ipv4)
      |                 |     |  +--rw ipv4?                  inet:ipv4-address
```

```
|                  |     +--:(ipv6)
|                  |     |  +--rw ipv6?                 inet:ipv6-address
|                  |     +--:(mac)
|                  |     |  +--rw mac?                  yang:mac-address
|                  |     +--:(distinguished-name)
|                  |     |  +--rw distinguished-name?   distinguished-name
|                  |     +--:(lcaf)
|                  |        +--rw lcaf
|                  |           +--rw lcaf-type               lcaf-ref
|                  |           +--rw (address)?
|                  |              +--:(null-address)
|                  |              |  +--rw null-address
|                  |              |     +--rw address?   empty
|                  |              +--:(afi-list)
|                  |              |  +--rw afi-list
|                  |              |     +--rw address-list*   simple-addre
|                  |              +--:(instance-id)
|                  |              |  +--rw instance-id
|                  |              |     +--rw instance-id?   instance-id-t
```



```
|                  |              |     +--rw mask-length?   uint8
|                  |              |     +--rw address?       simple-addres
|                  |              +--:(as-number)
|                  |              |  +--rw as-number
|                  |              |     +--rw as?        inet:as-number
|                  |              |     +--rw address?   simple-address
|                  |              +--:(application-data)
|                  |              |  +--rw application-data
|                  |              |     +--rw address?            simple-a
|                  |              |     +--rw protocol?           uint8
|                  |              |     +--rw ip-tos?             int32
|                  |              |     +--rw local-port-low?     inet:por
|                  |              |     +--rw local-port-high?    inet:por
|                  |              |     +--rw remote-port-low?    inet:por
|                  |              |     +--rw remote-port-high?   inet:por
|                  |              +--:(geo-coordinates)
|                  |              |  +--rw geo-coordinates
|                  |              |     +--rw latitude?            bits
|                  |              |     +--rw latitude-degrees?    uint8
|                  |              |     +--rw latitude-minutes?    uint8
|                  |              |     +--rw latitude-seconds?    uint8
|                  |              |     +--rw longitude?           bits
```

```
|                  |                |     +--rw longitude-degrees?   uint16
|                  |                |     +--rw longitude-minutes?   uint8
|                  |                |     +--rw longitude-seconds?   uint8
|                  |                |     +--rw altitude?            int32
|                  |                |     +--rw address?             simple-
|                  |                +--:(nat-traversal)
|                  |                |  +--rw nat-traversal
|                  |                |     +--rw ms-udp-port?        uint16
|                  |                |     +--rw etr-udp-port?       uint16
|                  |                |     +--rw global-etr-rloc?    simple-a
|                  |                |     +--rw ms-rloc?            simple-a
|                  |                |     +--rw private-etr-rloc?   simple-a
|                  |                |     +--rw rtr-rlocs*          simple-a
|                  |                +--:(explicit-locator-path)
|                  |                |  +--rw explicit-locator-path
|                  |                |     +--rw hop* [address]
|                  |                |        +--rw address     simple-addres
|                  |                |        +--rw lrs-bits?   bits
|                  |                +--:(sourc-dest-key)
|                  |                   +--rw sourc-dest-key
|                  |                      +--rw source?   inet:ip-prefix
|                  |                      +--rw dest?     inet:ip-prefix
|                  +--rw more-specifics-accepted?      boolean
|                  +--rw mapping-expiration-timeout?   int16
+--rw static-mappings
|  +--rw mapping* [id]
```



```
|     +--rw id                  eid-id
|     +--rw eid
|     |  +--rw afi                   lisp-address-family-ref
|     |  +--rw instance-id?          instance-id-type
|     |  +--rw (address)?
|     |     +--:(ipv4)
|     |     |  +--rw ipv4?                inet:ipv4-address
|     |     +--:(ipv6)
|     |     |  +--rw ipv6?                inet:ipv6-address
|     |     +--:(mac)
|     |     |  +--rw mac?                 yang:mac-address
|     |     +--:(distinguished-name)
|     |     |  +--rw distinguished-name?   distinguished-name
|     |     +--:(lcaf)
|     |        +--rw lcaf
```

```
|     |           +--rw lcaf-type                lcaf-ref
|     |           +--rw (address)?
|     |              +--:(null-address)
|     |              |  +--rw null-address
|     |              |     +--rw address?   empty
|     |              +--:(afi-list)
|     |              |  +--rw afi-list
|     |              |     +--rw address-list*   simple-address
|     |              +--:(instance-id)
|     |              |  +--rw instance-id
|     |              |     +--rw instance-id?   instance-id-type
|     |              |     +--rw mask-length?   uint8
|     |              |     +--rw address?       simple-address
|     |              +--:(as-number)
|     |              |  +--rw as-number
|     |              |     +--rw as?        inet:as-number
|     |              |     +--rw address?   simple-address
|     |              +--:(application-data)
|     |              |  +--rw application-data
|     |              |     +--rw address?            simple-address
|     |              |     +--rw protocol?           uint8
|     |              |     +--rw ip-tos?             int32
|     |              |     +--rw local-port-low?     inet:port-number
|     |              |     +--rw local-port-high?    inet:port-number
|     |              |     +--rw remote-port-low?    inet:port-number
|     |              |     +--rw remote-port-high?   inet:port-number
|     |              +--:(geo-coordinates)
|     |              |  +--rw geo-coordinates
|     |              |     +--rw latitude?            bits
|     |              |     +--rw latitude-degrees?    uint8
|     |              |     +--rw latitude-minutes?    uint8
|     |              |     +--rw latitude-seconds?    uint8
|     |              |     +--rw longitude?           bits
```



```
|     |              |     +--rw longitude-degrees?   uint16
|     |              |     +--rw longitude-minutes?   uint8
|     |              |     +--rw longitude-seconds?   uint8
|     |              |     +--rw altitude?            int32
|     |              |     +--rw address?             simple-address
|     |              +--:(nat-traversal)
|     |              |  +--rw nat-traversal
|     |              |     +--rw ms-udp-port?        uint16
```

```
|     |              |     +--rw etr-udp-port?       uint16
|     |              |     +--rw global-etr-rloc?    simple-address
|     |              |     +--rw ms-rloc?            simple-address
|     |              |     +--rw private-etr-rloc?   simple-address
|     |              |     +--rw rtr-rlocs*          simple-address
|     |              +--:(explicit-locator-path)
|     |              |  +--rw explicit-locator-path
|     |              |     +--rw hop* [address]
|     |              |        +--rw address     simple-address
|     |              |        +--rw lrs-bits?   bits
|     |              +--:(sourc-dest-key)
|     |                 +--rw sourc-dest-key
|     |                    +--rw source?   inet:ip-prefix
|     |                    +--rw dest?     inet:ip-prefix
|     +--rw ttl?                uint32
|     +--rw authoritative?      bits
|     +--rw (locator-list)?
|        +--:(negative-mapping)
|        |  +--rw map-reply-action?   map-reply-action
|        +--:(positive-mapping)
|           +--rw rlocs
|              +--rw rloc* [id]
|                 +--rw id                     string
|                 +--rw (address-type)?
|                 |  +--:(interface-name)
|                 |  |  +--rw interface?            if:interface-state-re
|                 |  +--:(address)
|                 |     +--rw locator-address
|                 |        +--rw afi                    lisp-address-famil
|                 |        +--rw instance-id?           instance-id-type
|                 |        +--rw (address)?
|                 |           +--:(ipv4)
|                 |           |  +--rw ipv4?                inet:ipv4-ad
|                 |           +--:(ipv6)
|                 |           |  +--rw ipv6?                inet:ipv6-ad
|                 |           +--:(mac)
|                 |           |  +--rw mac?                 yang:mac-add
|                 |           +--:(distinguished-name)
|                 |           |  +--rw distinguished-name?   distinguishe
|                 |           +--:(lcaf)
```



```
|                 |              +--rw lcaf
```

```
|                  |                  +--rw lcaf-type                lcaf-r
|                  |                  +--rw (address)?
|                  |                     +--:(null-address)
|                  |                     |  +--rw null-address
|                  |                     |     +--rw address?   empty
|                  |                     +--:(afi-list)
|                  |                     |  +--rw afi-list
|                  |                     |     +--rw address-list*   simple
|                  |                     +--:(instance-id)
|                  |                     |  +--rw instance-id
|                  |                     |     +--rw instance-id?   instanc
|                  |                     |     +--rw mask-length?   uint8
|                  |                     |     +--rw address?       simple-
|                  |                     +--:(as-number)
|                  |                     |  +--rw as-number
|                  |                     |     +--rw as?        inet:as-num
|                  |                     |     +--rw address?   simple-addr
|                  |                     +--:(application-data)
|                  |                     |  +--rw application-data
|                  |                     |     +--rw address?            si
|                  |                     |     +--rw protocol?           ui
|                  |                     |     +--rw ip-tos?             in
|                  |                     |     +--rw local-port-low?     in
|                  |                     |     +--rw local-port-high?    in
|                  |                     |     +--rw remote-port-low?    in
|                  |                     |     +--rw remote-port-high?   in
|                  |                     +--:(geo-coordinates)
|                  |                     |  +--rw geo-coordinates
|                  |                     |     +--rw latitude?            b
|                  |                     |     +--rw latitude-degrees?    u
|                  |                     |     +--rw latitude-minutes?    u
|                  |                     |     +--rw latitude-seconds?    u
|                  |                     |     +--rw longitude?           b
|                  |                     |     +--rw longitude-degrees?   u
|                  |                     |     +--rw longitude-minutes?   u
|                  |                     |     +--rw longitude-seconds?   u
|                  |                     |     +--rw altitude?            i
|                  |                     |     +--rw address?             s
|                  |                     +--:(nat-traversal)
|                  |                     |  +--rw nat-traversal
|                  |                     |     +--rw ms-udp-port?        ui
|                  |                     |     +--rw etr-udp-port?       ui
|                  |                     |     +--rw global-etr-rloc?    si
|                  |                     |     +--rw ms-rloc?            si
|                  |                     |     +--rw private-etr-rloc?   si
|                  |                     |     +--rw rtr-rlocs*          si
|                  |                     +--:(explicit-locator-path)
```

```
      |                  |                    |  +--rw explicit-locator-path
      |                  |                    |     +--rw hop* [address]
      |                  |                    |        +--rw address     simple-
      |                  |                    |        +--rw lrs-bits?   bits
      |                  |                   +--:(sourc-dest-key)
      |                  |                      +--rw sourc-dest-key
      |                  |                         +--rw source?   inet:ip-pref
      |                  |                         +--rw dest?     inet:ip-pref
      |                  +--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             lisp:eid-id
            |           +--rw eid-address
            |              +--rw afi                    lisp-address-family-ref
            |              +--rw instance-id?           instance-id-type
            |              +--rw (address)?
            |                 +--:(ipv4)
            |                 |  +--rw ipv4?                inet:ipv4-address
            |                 +--:(ipv6)
            |                 |  +--rw ipv6?                inet:ipv6-address
            |                 +--:(mac)
            |                 |  +--rw mac?                 yang:mac-address
            |                 +--:(distinguished-name)
            |                 |  +--rw distinguished-name?   distinguished-name
            |                 +--:(lcaf)
            |                    +--rw lcaf
            |                       +--rw lcaf-type               lcaf-ref
            |                       +--rw (address)?
            |                          +--:(null-address)
            |                          |  +--rw null-address
            |                          |     +--rw address?   empty
            |                          +--:(afi-list)
            |                          |  +--rw afi-list
            |                          |     +--rw address-list*   simple-address
            |                          +--:(instance-id)
            |                          |  +--rw instance-id
            |                          |     +--rw instance-id?   instance-id-type
            |                          |     +--rw mask-length?   uint8
            |                          |     +--rw address?       simple-address
```

```
|                          +--:(as-number)
|                          |  +--rw as-number
```



```
|                          |     +--rw as?        inet:as-number
|                          |     +--rw address?   simple-address
|                          +--:(application-data)
|                          |  +--rw application-data
|                          |     +--rw address?            simple-addr
|                          |     +--rw protocol?           uint8
|                          |     +--rw ip-tos?             int32
|                          |     +--rw local-port-low?     inet:port-n
|                          |     +--rw local-port-high?    inet:port-n
|                          |     +--rw remote-port-low?    inet:port-n
|                          |     +--rw remote-port-high?   inet:port-n
|                          +--:(geo-coordinates)
|                          |  +--rw geo-coordinates
|                          |     +--rw latitude?            bits
|                          |     +--rw latitude-degrees?    uint8
|                          |     +--rw latitude-minutes?    uint8
|                          |     +--rw latitude-seconds?    uint8
|                          |     +--rw longitude?           bits
|                          |     +--rw longitude-degrees?   uint16
|                          |     +--rw longitude-minutes?   uint8
|                          |     +--rw longitude-seconds?   uint8
|                          |     +--rw altitude?            int32
|                          |     +--rw address?             simple-add
|                          +--:(nat-traversal)
|                          |  +--rw nat-traversal
|                          |     +--rw ms-udp-port?        uint16
|                          |     +--rw etr-udp-port?       uint16
|                          |     +--rw global-etr-rloc?    simple-addr
|                          |     +--rw ms-rloc?            simple-addr
|                          |     +--rw private-etr-rloc?   simple-addr
|                          |     +--rw rtr-rlocs*          simple-addr
|                          +--:(explicit-locator-path)
|                          |  +--rw explicit-locator-path
|                          |     +--rw hop* [address]
|                          |        +--rw address     simple-address
|                          |        +--rw lrs-bits?   bits
|                          +--:(sourc-dest-key)
|                             +--rw sourc-dest-key
|                                +--rw source?   inet:ip-prefix
```

```
         |                                  +--rw dest?     inet:ip-prefix
         +--:(alt-mapping-system)
            +--rw alt-mapping-system!
augment /lisp:lisp-state/lisp:devices/lisp:device:
   +--ro ms-state
      +--ro registered-mappings
         +--ro mapping* [id]
            +--ro id                   eid-id
            +--ro eid
```



```
            |  +--ro afi                    lisp-address-family-ref
            |  +--ro instance-id?           instance-id-type
            |  +--ro (address)?
            |     +--:(ipv4)
            |     |  +--ro ipv4?                 inet:ipv4-address
            |     +--:(ipv6)
            |     |  +--ro ipv6?                 inet:ipv6-address
            |     +--:(mac)
            |     |  +--ro mac?                  yang:mac-address
            |     +--:(distinguished-name)
            |     |  +--ro distinguished-name?   distinguished-name
            |     +--:(lcaf)
            |        +--ro lcaf
            |           +--ro lcaf-type               lcaf-ref
            |           +--ro (address)?
            |              +--:(null-address)
            |              |  +--ro null-address
            |              |     +--ro address?   empty
            |              +--:(afi-list)
            |              |  +--ro afi-list
            |              |     +--ro address-list*   simple-address
            |              +--:(instance-id)
            |              |  +--ro instance-id
            |              |     +--ro instance-id?   instance-id-type
            |              |     +--ro mask-length?   uint8
            |              |     +--ro address?       simple-address
            |              +--:(as-number)
            |              |  +--ro as-number
            |              |     +--ro as?        inet:as-number
            |              |     +--ro address?   simple-address
            |              +--:(application-data)
            |              |  +--ro application-data
```

```
|              |     +--ro address?            simple-address
|              |     +--ro protocol?           uint8
|              |     +--ro ip-tos?             int32
|              |     +--ro local-port-low?     inet:port-number
|              |     +--ro local-port-high?    inet:port-number
|              |     +--ro remote-port-low?    inet:port-number
|              |     +--ro remote-port-high?   inet:port-number
|              +--:(geo-coordinates)
|              |  +--ro geo-coordinates
|              |     +--ro latitude?            bits
|              |     +--ro latitude-degrees?    uint8
|              |     +--ro latitude-minutes?    uint8
|              |     +--ro latitude-seconds?    uint8
|              |     +--ro longitude?           bits
|              |     +--ro longitude-degrees?   uint16
|              |     +--ro longitude-minutes?   uint8
```



```
|              |     +--ro longitude-seconds?   uint8
|              |     +--ro altitude?            int32
|              |     +--ro address?             simple-address
|              +--:(nat-traversal)
|              |  +--ro nat-traversal
|              |     +--ro ms-udp-port?        uint16
|              |     +--ro etr-udp-port?       uint16
|              |     +--ro global-etr-rloc?    simple-address
|              |     +--ro ms-rloc?            simple-address
|              |     +--ro private-etr-rloc?   simple-address
|              |     +--ro rtr-rlocs*          simple-address
|              +--:(explicit-locator-path)
|              |  +--ro explicit-locator-path
|              |     +--ro hop* [address]
|              |        +--ro address     simple-address
|              |        +--ro lrs-bits?   bits
|              +--:(sourc-dest-key)
|                 +--ro sourc-dest-key
|                    +--ro source?   inet:ip-prefix
|                    +--ro dest?     inet:ip-prefix
|           +--ro ttl?                uint32
|           +--ro authoritative?      bits
|           +--ro (locator-list)?
|              +--:(negative-mapping)
|              |  +--ro map-reply-action?   map-reply-action
```

```
          +--:(positive-mapping)
             +--ro rlocs
                +--ro rloc* [id]
                   +--ro id                    string
                   +--ro (address-type)?
                   |  +--:(interface-name)
                   |  |  +--ro interface?           if:interface-state-re
                   |  +--:(address)
                   |     +--ro locator-address
                   |        +--ro afi                   lisp-address-famil
                   |        +--ro instance-id?          instance-id-type
                   |        +--ro (address)?
                   |           +--:(ipv4)
                   |           |  +--ro ipv4?                inet:ipv4-ad
                   |           +--:(ipv6)
                   |           |  +--ro ipv6?                inet:ipv6-ad
                   |           +--:(mac)
                   |           |  +--ro mac?                 yang:mac-add
                   |           +--:(distinguished-name)
                   |           |  +--ro distinguished-name?  distinguishe
                   |           +--:(lcaf)
                   |              +--ro lcaf
                   |                 +--ro lcaf-type               lcaf-r
```



```
                   |                 +--ro (address)?
                   |                    +--:(null-address)
                   |                    |  +--ro null-address
                   |                    |     +--ro address?   empty
                   |                    +--:(afi-list)
                   |                    |  +--ro afi-list
                   |                    |     +--ro address-list*   simple
                   |                    +--:(instance-id)
                   |                    |  +--ro instance-id
                   |                    |     +--ro instance-id?   instanc
                   |                    |     +--ro mask-length?   uint8
                   |                    |     +--ro address?       simple-
                   |                    +--:(as-number)
                   |                    |  +--ro as-number
                   |                    |     +--ro as?        inet:as-num
                   |                    |     +--ro address?   simple-addr
                   |                    +--:(application-data)
                   |                    |  +--ro application-data
```

```
|                      |     +--ro address?            si
|                      |     +--ro protocol?           ui
|                      |     +--ro ip-tos?             in
|                      |     +--ro local-port-low?     in
|                      |     +--ro local-port-high?    in
|                      |     +--ro remote-port-low?    in
|                      |     +--ro remote-port-high?   in
|                      |   +--:(geo-coordinates)
|                      |   |  +--ro geo-coordinates
|                      |     +--ro latitude?            b
|                      |     +--ro latitude-degrees?    u
|                      |     +--ro latitude-minutes?    u
|                      |     +--ro latitude-seconds?    u
|                      |     +--ro longitude?           b
|                      |     +--ro longitude-degrees?   u
|                      |     +--ro longitude-minutes?   u
|                      |     +--ro longitude-seconds?   u
|                      |     +--ro altitude?            i
|                      |     +--ro address?             s
|                      |   +--:(nat-traversal)
|                      |   |  +--ro nat-traversal
|                      |     +--ro ms-udp-port?        ui
|                      |     +--ro etr-udp-port?       ui
|                      |     +--ro global-etr-rloc?    si
|                      |     +--ro ms-rloc?            si
|                      |     +--ro private-etr-rloc?   si
|                      |     +--ro rtr-rlocs*          si
|                      |   +--:(explicit-locator-path)
|                      |   |  +--ro explicit-locator-path
|                      |     +--ro hop* [address]
```



```
|                      |        +--ro address     simple-
|                      |        +--ro lrs-bits?   bits
|                      |   +--:(sourc-dest-key)
|                      |      +--ro sourc-dest-key
|                      |         +--ro source?   inet:ip-pref
|                      |         +--ro dest?     inet:ip-pref
|                    +--ro priority?              uint8
|                    +--ro weight?                uint8
|                    +--ro multicast-priority?    uint8
|                    +--ro multicast-weight?      uint8
```

5.2. Module Definition

```
<CODE BEGINS> file "ietf-lisp-mapserver@2015-07-02.yang"
module ietf-lisp-mapserver {
  namespace "urn:ietf:params:xml:ns:yang:ietf-lisp-mapserver";
  prefix lisp-ms;
  import ietf-lisp {
    prefix lisp;
  }
  import ietf-lisp-address-types {
   prefix lcaf;
  }
  import ietf-inet-types {
    prefix inet;
  }
  organization
   "IETF LISP (Locator/ID Separation Protocol) Working Group";
  contact
    "lisp@ietf.org";
  description
    "This YANG module defines the generic configuration
     data for a LISP Map-Server. The module can be extended by
     vendors to define vendor-specific configuration parameters
     and policies.

     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 2015-07-02 {
    description
```

```
      "Initial revision.";
    reference
      "https://tools.ietf.org/html/rfc6833";
  }
  identity ms {
    base lisp:lisp-device;
    description
      "LISP Map-Server.";
  }
  augment "/lisp:lisp/lisp:devices/lisp:device" {
    when "lisp:type = lisp-ms:ms" {
      description
        "Augment is valid when LISP device type is Map-Server.";
    }
    description
      "This augments LISP devices list with Map-Server specific
       parameters.";
    container map-server-cfg {
      presence "LISP Map-Server operation enabled";
      description
        "Map-Server configuration parameters.";
      container sites {
        description
          "Sites for which the Map-Server accepts registrations.";
        list site {
          key "site-id";
          description
            "List of sites for which the Map-Server acceptes
             registrations.";
          leaf site-id {
            type uint64;
            description
              "Site identifier.";
          }
    //Can be augmented to have below for per site auth-key
    //      leaf auth-key {
    //        description "clear text authentication key";
    //        type string;
    //      }
          container devices {
            description
              "Site devices registered with the Map-Server.";
            list device {
              key "device-id";
```

```
                description
                  "List of site devices registered with the
                   Map-Server.";
                leaf device-id {
                  type uint64;
                  description
                    "Device identifier.";
                }
                container auth-key {
                  description
                    "Device authentication key.";
                  leaf auth-key-value {
                    type string;
                    description
                      "Clear text authentication key";
                  }
                  leaf auth-key-type {
                    type lisp:auth-key-type;
                    description
                      "Authentication key type.";
                  }
                }

                container eids {
                  description
                    "EIDs registered by device.";
                  list eid {
                    key "id";
                    description
                      "List of EIDs registered by device.";
                    leaf id {
                      type lisp:eid-id;
                      description
                        "Id of the EID registered.";
                    }
                    container eid-address {
                      uses lcaf:lisp-address;
                      description
                        "EID in generic LISP address format registered
                         with the Map-Server.";
                    }
                    leaf more-specifics-accepted {
                        type boolean;
                        description
                          "Flag indicating if more specific prefixes
                           can be registered.";
                    }
```

```
                    leaf mapping-expiration-timeout {
```



```
                      type int16;
                      units "seconds";
                      default "180"; //3 times the mapregister int
                      description
                        "Time before mapping is expired if no new
                         registrations are received.";
                    }
          // Can be augmented to have below for per-eid auth-key
          //          leaf auth-key {
          //            description "clear text authentication key";
          //            type string;
          //          }
                  }
                }
              }
            }
          }
          container static-mappings {
            uses lisp:mappings;
            description
              "Datastore of statically registred mappings.";
          }
          choice mapping-system {
            mandatory true;
            description
              "Interface to Mapping System.";
            container ddt-mapping-system {
              presence "DDT Mapping System in use";
              description
                "DDT Mapping System interface.";
              leaf-list ddt-root {
                type inet:ip-address;
                description
                  "List of DDT root address.";
              }
              container authoritative-eids {
                description
                  "Authoritative EIDs";
                list authoritative-eid {
```

```
                key "id";
                description
                  "List of authoritative EIDs.";
                leaf id {
                  type lisp:eid-id;
                  description
                    "EID id.";
                }
```



```
                container eid-address {
                  uses lcaf:lisp-address;
                  description
                    "EID in generic LISP address format.";
                }
              }
            }
          }
          container alt-mapping-system {
            presence "ALT Mapping System in use";
            description
              "ALT Mapping System interface.";
          // TBD
          }
        }
      }
    }
    augment "/lisp:lisp-state/lisp:devices/lisp:device" {
      when "lisp:type = lisp-ms:ms" {
        description
          "Augment is valid when LISP device type is Map-Server.";
      }
      description
        "This augments LISP devices list state with Map-Server
         specific parameters.";
      container ms-state {
        config false;
        description
          "Map-Server state.";
        container registered-mappings {
          uses lisp:mappings;
          description
            "Datastore of registred mappings.";
```

```
        }
      }
    }
  }
  <CODE ENDS>
```

6. LISP Map Resolver Module

This module captures the configuration data model of a LISP Map Resolver [RFC6833]. The model also captures some operational data elements.

6.1. Module Structure

```
module: ietf-lisp-mapresolver
augment /lisp:lisp/lisp:devices/lisp:device:
   +--rw map-resolver-cfg!
      +--rw (mapping-system)
         +--:(ddt-mapping-system)
         |  +--rw ddt-mapping-system!
         |     +--rw ddt-root*   inet:ip-address
         +--:(alt-mapping-system)
            +--rw alt-mapping-system!
```

6.2. Module Definition

```
<CODE BEGINS> file "ietf-lisp-mapresolver@2015-07-02.yang"
module ietf-lisp-mapresolver {
  namespace "urn:ietf:params:xml:ns:yang:ietf-lisp-mapresolver";
  prefix lisp-mr;
  import ietf-lisp {
    prefix lisp;
  }
  import ietf-inet-types {
    prefix inet;
  }
  organization
   "IETF LISP (Locator/ID Separation Protocol) Working Group";
```

```
    contact
      "lisp@ietf.org";
    description
      "This YANG module defines the generic configuration
       data for a LISP Map-Resolver. The module can be extended by
       vendors to define vendor-specific configuration parameters
       and policies.

       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 2015-07-02 {
```



```
      description
        "Initial revision.";
      reference
        "https://tools.ietf.org/html/rfc6833";
    }
    identity mr {
      base lisp:lisp-device;
      description
        "LISP Map-Resolver.";
    }
    augment "/lisp:lisp/lisp:devices/lisp:device" {
      when "lisp:type = lisp-mr:mr" {
        description
          "Augment is valid when LISP device type is Map-Resolver.";
      }
      description
        "This augments LISP devices list with Map-Resolver specific
         parameters.";
      container map-resolver-cfg {
```

```
        presence "LISP Map-Resolver operation enabled";
        description
          "Map-Resolver configuration parameters.";
        choice mapping-system {
          mandatory true;
          description
            "Mapping system interface.";
          container ddt-mapping-system {
            presence "DDT Mapping System in use";
            description
              "DDT mapping system interface.";
            leaf-list ddt-root {
              type inet:ip-address;
              description
                "List of DDT root addresses.";
            }
          }
          container alt-mapping-system{
            presence "ALT Mapping System in use";
            description
              "ALT mapping system interface.";
           // TBD
          }
        }
      }
    }
    <CODE ENDS>
```


7. LISP Proxy ITR Module

This module captures the configuration data model of a LISP Proxy ITR [RFC6832]. The model may also capture some operational data elements.

7.1. Module Structure

```
module: ietf-lisp-pitr
augment /lisp:lisp/lisp:devices/lisp:device:
   +--rw proxy-itr-cfg!
      +--rw servicing-eids
```

```
|  +--rw eid* [id]
|     +--rw id             lisp:eid-id
|     +--rw eid-address
|        +--rw afi                    lisp-address-family-ref
|        +--rw instance-id?           instance-id-type
|        +--rw (address)?
|           +--:(ipv4)
|           |  +--rw ipv4?                 inet:ipv4-address
|           +--:(ipv6)
|           |  +--rw ipv6?                 inet:ipv6-address
|           +--:(mac)
|           |  +--rw mac?                  yang:mac-address
|           +--:(distinguished-name)
|           |  +--rw distinguished-name?   distinguished-name
|           +--:(lcaf)
|              +--rw lcaf
|                 +--rw lcaf-type                lcaf-ref
|                 +--rw (address)?
|                    +--:(null-address)
|                    |  +--rw null-address
|                    |     +--rw address?   empty
|                    +--:(afi-list)
|                    |  +--rw afi-list
|                    |     +--rw address-list*   simple-address
|                    +--:(instance-id)
|                    |  +--rw instance-id
|                    |     +--rw instance-id?   instance-id-type
|                    |     +--rw mask-length?   uint8
|                    |     +--rw address?       simple-address
|                    +--:(as-number)
|                    |  +--rw as-number
|                    |     +--rw as?        inet:as-number
|                    |     +--rw address?   simple-address
|                    +--:(application-data)
|                    |  +--rw application-data
|                    |     +--rw address?            simple-address
```



```
|                    |     +--rw protocol?          uint8
|                    |     +--rw ip-tos?            int32
|                    |     +--rw local-port-low?    inet:port-number
|                    |     +--rw local-port-high?   inet:port-number
|                    |     +--rw remote-port-low?   inet:port-number
```

```
|                      |     +--rw remote-port-high?   inet:port-number
|                      +--:(geo-coordinates)
|                      |  +--rw geo-coordinates
|                      |     +--rw latitude?            bits
|                      |     +--rw latitude-degrees?    uint8
|                      |     +--rw latitude-minutes?    uint8
|                      |     +--rw latitude-seconds?    uint8
|                      |     +--rw longitude?           bits
|                      |     +--rw longitude-degrees?   uint16
|                      |     +--rw longitude-minutes?   uint8
|                      |     +--rw longitude-seconds?   uint8
|                      |     +--rw altitude?            int32
|                      |     +--rw address?             simple-address
|                      +--:(nat-traversal)
|                      |  +--rw nat-traversal
|                      |     +--rw ms-udp-port?        uint16
|                      |     +--rw etr-udp-port?       uint16
|                      |     +--rw global-etr-rloc?    simple-address
|                      |     +--rw ms-rloc?            simple-address
|                      |     +--rw private-etr-rloc?   simple-address
|                      |     +--rw rtr-rlocs*          simple-address
|                      +--:(explicit-locator-path)
|                      |  +--rw explicit-locator-path
|                      |     +--rw hop* [address]
|                      |        +--rw address     simple-address
|                      |        +--rw lrs-bits?   bits
|                      +--:(sourc-dest-key)
|                         +--rw sourc-dest-key
|                            +--rw source?   inet:ip-prefix
|                            +--rw dest?     inet:ip-prefix
|
+--rw map-resolvers
|  +--rw map-resolver* [id]
|     +--rw id         lisp:eid-id
|     +--rw eid
|     |  +--rw afi                   lisp-address-family-ref
|     |  +--rw instance-id?          instance-id-type
|     |  +--rw (address)?
|     |     +--:(ipv4)
|     |     |  +--rw ipv4?                inet:ipv4-address
|     |     +--:(ipv6)
|     |     |  +--rw ipv6?                inet:ipv6-address
|     |     +--:(mac)
|     |     |  +--rw mac?                 yang:mac-address
```

```
|      |      +--:(distinguished-name)
|      |      |  +--rw distinguished-name?   distinguished-name
|      |      +--:(lcaf)
|      |         +--rw lcaf
|      |            +--rw lcaf-type                lcaf-ref
|      |            +--rw (address)?
|      |               +--:(null-address)
|      |               |  +--rw null-address
|      |               |     +--rw address?   empty
|      |               +--:(afi-list)
|      |               |  +--rw afi-list
|      |               |     +--rw address-list*   simple-address
|      |               +--:(instance-id)
|      |               |  +--rw instance-id
|      |               |     +--rw instance-id?   instance-id-type
|      |               |     +--rw mask-length?   uint8
|      |               |     +--rw address?       simple-address
|      |               +--:(as-number)
|      |               |  +--rw as-number
|      |               |     +--rw as?        inet:as-number
|      |               |     +--rw address?   simple-address
|      |               +--:(application-data)
|      |               |  +--rw application-data
|      |               |     +--rw address?            simple-address
|      |               |     +--rw protocol?           uint8
|      |               |     +--rw ip-tos?             int32
|      |               |     +--rw local-port-low?     inet:port-number
|      |               |     +--rw local-port-high?    inet:port-number
|      |               |     +--rw remote-port-low?    inet:port-number
|      |               |     +--rw remote-port-high?   inet:port-number
|      |               +--:(geo-coordinates)
|      |               |  +--rw geo-coordinates
|      |               |     +--rw latitude?            bits
|      |               |     +--rw latitude-degrees?    uint8
|      |               |     +--rw latitude-minutes?    uint8
|      |               |     +--rw latitude-seconds?    uint8
|      |               |     +--rw longitude?           bits
|      |               |     +--rw longitude-degrees?   uint16
|      |               |     +--rw longitude-minutes?   uint8
|      |               |     +--rw longitude-seconds?   uint8
|      |               |     +--rw altitude?            int32
|      |               |     +--rw address?             simple-address
|      |               +--:(nat-traversal)
|      |               |  +--rw nat-traversal
|      |               |     +--rw ms-udp-port?        uint16
|      |               |     +--rw etr-udp-port?       uint16
|      |               |     +--rw global-etr-rloc?    simple-address
|      |               |     +--rw ms-rloc?            simple-address
```

```
      |     |              |     +--rw private-etr-rloc?   simple-address
      |     |              |     +--rw rtr-rlocs*          simple-address
      |     |              +--:(explicit-locator-path)
      |     |              |  +--rw explicit-locator-path
      |     |              |     +--rw hop* [address]
      |     |              |        +--rw address     simple-address
      |     |              |        +--rw lrs-bits?   bits
      |     |              +--:(sourc-dest-key)
      |     |                 +--rw sourc-dest-key
      |     |                    +--rw source?   inet:ip-prefix
      |     |                    +--rw dest?     inet:ip-prefix
      |     +--rw address*   inet:ip-address
      +--rw static-mappings
         +--rw mapping* [id]
            +--rw id                   eid-id
            +--rw eid
            |  +--rw afi                   lisp-address-family-ref
            |  +--rw instance-id?          instance-id-type
            |  +--rw (address)?
            |     +--:(ipv4)
            |     |  +--rw ipv4?                inet:ipv4-address
            |     +--:(ipv6)
            |     |  +--rw ipv6?                inet:ipv6-address
            |     +--:(mac)
            |     |  +--rw mac?                 yang:mac-address
            |     +--:(distinguished-name)
            |     |  +--rw distinguished-name?   distinguished-name
            |     +--:(lcaf)
            |        +--rw lcaf
            |           +--rw lcaf-type               lcaf-ref
            |           +--rw (address)?
            |              +--:(null-address)
            |              |  +--rw null-address
            |              |     +--rw address?   empty
            |              +--:(afi-list)
            |              |  +--rw afi-list
            |              |     +--rw address-list*   simple-address
            |              +--:(instance-id)
            |              |  +--rw instance-id
            |              |     +--rw instance-id?   instance-id-type
            |              |     +--rw mask-length?   uint8
            |              |     +--rw address?       simple-address
```

```
|              +--:(as-number)
|              |  +--rw as-number
|              |     +--rw as?        inet:as-number
|              |     +--rw address?   simple-address
|              +--:(application-data)
|              |  +--rw application-data
```



```
|              |     +--rw address?           simple-address
|              |     +--rw protocol?          uint8
|              |     +--rw ip-tos?            int32
|              |     +--rw local-port-low?    inet:port-number
|              |     +--rw local-port-high?   inet:port-number
|              |     +--rw remote-port-low?   inet:port-number
|              |     +--rw remote-port-high?  inet:port-number
|              +--:(geo-coordinates)
|              |  +--rw geo-coordinates
|              |     +--rw latitude?           bits
|              |     +--rw latitude-degrees?   uint8
|              |     +--rw latitude-minutes?   uint8
|              |     +--rw latitude-seconds?   uint8
|              |     +--rw longitude?          bits
|              |     +--rw longitude-degrees?  uint16
|              |     +--rw longitude-minutes?  uint8
|              |     +--rw longitude-seconds?  uint8
|              |     +--rw altitude?           int32
|              |     +--rw address?            simple-address
|              +--:(nat-traversal)
|              |  +--rw nat-traversal
|              |     +--rw ms-udp-port?       uint16
|              |     +--rw etr-udp-port?      uint16
|              |     +--rw global-etr-rloc?   simple-address
|              |     +--rw ms-rloc?           simple-address
|              |     +--rw private-etr-rloc?  simple-address
|              |     +--rw rtr-rlocs*         simple-address
|              +--:(explicit-locator-path)
|              |  +--rw explicit-locator-path
|              |     +--rw hop* [address]
|              |        +--rw address     simple-address
|              |        +--rw lrs-bits?   bits
|              +--:(sourc-dest-key)
|                 +--rw sourc-dest-key
|                    +--rw source?   inet:ip-prefix
```

```
|                     +--rw dest?     inet:ip-prefix
+--rw ttl?                uint32
+--rw authoritative?      bits
+--rw (locator-list)?
   +--:(negative-mapping)
   |  +--rw map-reply-action?   map-reply-action
   +--:(positive-mapping)
      +--rw rlocs
         +--rw rloc* [id]
            +--rw id                    string
            +--rw (address-type)?
            |  +--:(interface-name)
            |  |  +--rw interface?            if:interface-state-re
```



```
                     |  +--:(address)
                     |     +--rw locator-address
                     |        +--rw afi                    lisp-address-famil
                     |        +--rw instance-id?           instance-id-type
                     |        +--rw (address)?
                     |           +--:(ipv4)
                     |           |  +--rw ipv4?                 inet:ipv4-ad
                     |           +--:(ipv6)
                     |           |  +--rw ipv6?                 inet:ipv6-ad
                     |           +--:(mac)
                     |           |  +--rw mac?                  yang:mac-add
                     |           +--:(distinguished-name)
                     |           |  +--rw distinguished-name?   distinguishe
                     |           +--:(lcaf)
                     |              +--rw lcaf
                     |                 +--rw lcaf-type               lcaf-r
                     |                 +--rw (address)?
                     |                    +--:(null-address)
                     |                    |  +--rw null-address
                     |                    |     +--rw address?   empty
                     |                    +--:(afi-list)
                     |                    |  +--rw afi-list
                     |                    |     +--rw address-list*   simple
                     |                    +--:(instance-id)
                     |                    |  +--rw instance-id
                     |                    |     +--rw instance-id?   instanc
                     |                    |     +--rw mask-length?   uint8
                     |                    |     +--rw address?       simple-
```

```
|                      +--:(as-number)
|                      |  +--rw as-number
|                      |     +--rw as?        inet:as-num
|                      |     +--rw address?   simple-addr
|                      +--:(application-data)
|                      |  +--rw application-data
|                      |     +--rw address?            si
|                      |     +--rw protocol?           ui
|                      |     +--rw ip-tos?             in
|                      |     +--rw local-port-low?     in
|                      |     +--rw local-port-high?    in
|                      |     +--rw remote-port-low?    in
|                      |     +--rw remote-port-high?   in
|                      +--:(geo-coordinates)
|                      |  +--rw geo-coordinates
|                      |     +--rw latitude?            b
|                      |     +--rw latitude-degrees?    u
|                      |     +--rw latitude-minutes?    u
|                      |     +--rw latitude-seconds?    u
|                      |     +--rw longitude?           b
```



```
|                      |     +--rw longitude-degrees?   u
|                      |     +--rw longitude-minutes?   u
|                      |     +--rw longitude-seconds?   u
|                      |     +--rw altitude?            i
|                      |     +--rw address?             s
|                      +--:(nat-traversal)
|                      |  +--rw nat-traversal
|                      |     +--rw ms-udp-port?        ui
|                      |     +--rw etr-udp-port?       ui
|                      |     +--rw global-etr-rloc?    si
|                      |     +--rw ms-rloc?            si
|                      |     +--rw private-etr-rloc?   si
|                      |     +--rw rtr-rlocs*          si
|                      +--:(explicit-locator-path)
|                      |  +--rw explicit-locator-path
|                      |     +--rw hop* [address]
|                      |        +--rw address     simple-
|                      |        +--rw lrs-bits?   bits
|                      +--:(sourc-dest-key)
|                         +--rw sourc-dest-key
|                            +--rw source?   inet:ip-pref
```

```
                        |                              +--rw dest?     inet:ip-pref
                        +--rw priority?                 uint8
                        +--rw weight?                   uint8
                        +--rw multicast-priority?       uint8
                        +--rw multicast-weight?         uint8
augment /lisp:lisp-state/lisp:devices/lisp:device:
   +--ro pitr-state
      +--ro learned-mappings
         +--ro mapping* [id]
            +--ro id                    eid-id
            +--ro eid
            |  +--ro afi                     lisp-address-family-ref
            |  +--ro instance-id?            instance-id-type
            |  +--ro (address)?
            |     +--:(ipv4)
            |     |  +--ro ipv4?                  inet:ipv4-address
            |     +--:(ipv6)
            |     |  +--ro ipv6?                  inet:ipv6-address
            |     +--:(mac)
            |     |  +--ro mac?                   yang:mac-address
            |     +--:(distinguished-name)
            |     |  +--ro distinguished-name?    distinguished-name
            |     +--:(lcaf)
            |        +--ro lcaf
            |           +--ro lcaf-type                lcaf-ref
            |           +--ro (address)?
            |              +--:(null-address)
```



```
            |              |  +--ro null-address
            |              |     +--ro address?   empty
            |              +--:(afi-list)
            |              |  +--ro afi-list
            |              |     +--ro address-list*   simple-address
            |              +--:(instance-id)
            |              |  +--ro instance-id
            |              |     +--ro instance-id?   instance-id-type
            |              |     +--ro mask-length?   uint8
            |              |     +--ro address?       simple-address
            |              +--:(as-number)
            |              |  +--ro as-number
            |              |     +--ro as?         inet:as-number
            |              |     +--ro address?    simple-address
```

```
|              +--:(application-data)
|              |  +--ro application-data
|              |     +--ro address?            simple-address
|              |     +--ro protocol?           uint8
|              |     +--ro ip-tos?             int32
|              |     +--ro local-port-low?     inet:port-number
|              |     +--ro local-port-high?    inet:port-number
|              |     +--ro remote-port-low?    inet:port-number
|              |     +--ro remote-port-high?   inet:port-number
|              +--:(geo-coordinates)
|              |  +--ro geo-coordinates
|              |     +--ro latitude?            bits
|              |     +--ro latitude-degrees?    uint8
|              |     +--ro latitude-minutes?    uint8
|              |     +--ro latitude-seconds?    uint8
|              |     +--ro longitude?           bits
|              |     +--ro longitude-degrees?   uint16
|              |     +--ro longitude-minutes?   uint8
|              |     +--ro longitude-seconds?   uint8
|              |     +--ro altitude?            int32
|              |     +--ro address?             simple-address
|              +--:(nat-traversal)
|              |  +--ro nat-traversal
|              |     +--ro ms-udp-port?        uint16
|              |     +--ro etr-udp-port?       uint16
|              |     +--ro global-etr-rloc?    simple-address
|              |     +--ro ms-rloc?            simple-address
|              |     +--ro private-etr-rloc?   simple-address
|              |     +--ro rtr-rlocs*          simple-address
|              +--:(explicit-locator-path)
|              |  +--ro explicit-locator-path
|              |     +--ro hop* [address]
|              |        +--ro address     simple-address
|              |        +--ro lrs-bits?   bits
```



```
|              +--:(sourc-dest-key)
|                 +--ro sourc-dest-key
|                    +--ro source?   inet:ip-prefix
|                    +--ro dest?     inet:ip-prefix
|           +--ro ttl?                uint32
|           +--ro authoritative?      bits
|           +--ro (locator-list)?
```

```
                +--:(negative-mapping)
                |  +--ro map-reply-action?   map-reply-action
                +--:(positive-mapping)
                   +--ro rlocs
                      +--ro rloc* [id]
                         +--ro id                      string
                         +--ro (address-type)?
                         |  +--:(interface-name)
                         |  |  +--ro interface?            if:interface-state-re
                         |  +--:(address)
                         |     +--ro locator-address
                         |        +--ro afi                    lisp-address-famil
                         |        +--ro instance-id?           instance-id-type
                         |        +--ro (address)?
                         |           +--:(ipv4)
                         |           |  +--ro ipv4?                inet:ipv4-ad
                         |           +--:(ipv6)
                         |           |  +--ro ipv6?                inet:ipv6-ad
                         |           +--:(mac)
                         |           |  +--ro mac?                 yang:mac-add
                         |           +--:(distinguished-name)
                         |           |  +--ro distinguished-name?   distinguishe
                         |           +--:(lcaf)
                         |              +--ro lcaf
                         |                 +--ro lcaf-type               lcaf-r
                         |                 +--ro (address)?
                         |                    +--:(null-address)
                         |                    |  +--ro null-address
                         |                    |     +--ro address?   empty
                         |                    +--:(afi-list)
                         |                    |  +--ro afi-list
                         |                    |     +--ro address-list*   simple
                         |                    +--:(instance-id)
                         |                    |  +--ro instance-id
                         |                    |     +--ro instance-id?   instanc
                         |                    |     +--ro mask-length?   uint8
                         |                    |     +--ro address?       simple-
                         |                    +--:(as-number)
                         |                    |  +--ro as-number
                         |                    |     +--ro as?        inet:as-num
                         |                    |     +--ro address?   simple-addr
```

```
|                      +--:(application-data)
|                      |  +--ro application-data
|                      |     +--ro address?            si
|                      |     +--ro protocol?           ui
|                      |     +--ro ip-tos?             in
|                      |     +--ro local-port-low?     in
|                      |     +--ro local-port-high?    in
|                      |     +--ro remote-port-low?    in
|                      |     +--ro remote-port-high?   in
|                      +--:(geo-coordinates)
|                      |  +--ro geo-coordinates
|                      |     +--ro latitude?            b
|                      |     +--ro latitude-degrees?    u
|                      |     +--ro latitude-minutes?    u
|                      |     +--ro latitude-seconds?    u
|                      |     +--ro longitude?           b
|                      |     +--ro longitude-degrees?   u
|                      |     +--ro longitude-minutes?   u
|                      |     +--ro longitude-seconds?   u
|                      |     +--ro altitude?            i
|                      |     +--ro address?             s
|                      +--:(nat-traversal)
|                      |  +--ro nat-traversal
|                      |     +--ro ms-udp-port?        ui
|                      |     +--ro etr-udp-port?       ui
|                      |     +--ro global-etr-rloc?    si
|                      |     +--ro ms-rloc?            si
|                      |     +--ro private-etr-rloc?   si
|                      |     +--ro rtr-rlocs*          si
|                      +--:(explicit-locator-path)
|                      |  +--ro explicit-locator-path
|                      |     +--ro hop* [address]
|                      |        +--ro address     simple-
|                      |        +--ro lrs-bits?   bits
|                      +--:(sourc-dest-key)
|                         +--ro sourc-dest-key
|                            +--ro source?   inet:ip-pref
|                            +--ro dest?     inet:ip-pref
|
+--ro priority?              uint8
+--ro weight?                uint8
+--ro multicast-priority?    uint8
+--ro multicast-weight?      uint8
```

7.2. Module Definition

```
<CODE BEGINS> file "ietf-lisp-pitr@2015-07-02.yang"
module ietf-lisp-pitr {
  namespace "urn:ietf:params:xml:ns:yang:ietf-lisp-pitr";
```

```
    prefix lisp-pitr;
    import ietf-lisp {
      prefix lisp;
    }
    import ietf-lisp-address-types {
     prefix lcaf;
    }
    import ietf-inet-types {
      prefix inet;
    }
    organization
     "IETF LISP (Locator/ID Separation Protocol) Working Group";
    contact
      "lisp@ietf.org";
    description
      "This YANG module defines the generic configuration
       data for a LISP PITR. The module can be extended by vendors
       to define vendor-specific configuration parameters and
       policies.

       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 2015-07-02 {
      description
        "Initial revision.";
      reference
        "https://tools.ietf.org/html/rfc6832";
    }
    identity pitr {
      base lisp:lisp-device;
      description
        "LISP PITR.";
```

```
    }
    augment "/lisp:lisp/lisp:devices/lisp:device" {
      when "lisp:type = lisp-pitr:pitr" {
        description
          "Augment is valid when LISP device type is PITR.";
```



```
      }
      description
        "This augments LISP devices list with PITR specific
         parameters.";
      container proxy-itr-cfg {
        presence "LISP PITR operation enabled";
        description
          "Proxy-ITR configuration parameters.";
        container servicing-eids {
          description
            "EIDs serviced by the PITR.";
          list eid {
            key "id";
            description
              "List of EIDs serviced by the PITR.";
            leaf id {
              type lisp:eid-id;
              description
                "Id of serviced EID.";
            }
            container eid-address {
              uses lcaf:lisp-address;
              description
                "Serviced EID address in generic LISP address format.";
            }
          }
        }

        container map-resolvers {
          description
            "Map-Resolvers configured for PITR.";
          list map-resolver {
            key "id";
            description
              "List of Map-Resolvers configured for PITR.";
            leaf id {
```

```
            type lisp:eid-id;
            description
              "Id of EID for which the Map-Resolver is used.";
          }
          container eid {
            uses lcaf:lisp-address;
            description
              "EID for which the Map-Resolver is used.";
          }
          leaf-list address {
            type inet:ip-address;
            min-elements 1;
```



```
            description
              "List of Map-Resolver's addresses.";
          }
        }
      }
      container static-mappings{
        uses lisp:mappings;
        description
          "EID to RLOCs mappings cache.";
      }
    }
  }
  augment "/lisp:lisp-state/lisp:devices/lisp:device" {
    when "lisp:type = lisp-pitr:pitr" {
      description
        "Augment is valid when LISP device type is PITR.";
    }
    description
      "This augments LISP devices list state with PITR specific
       parameters.";
    container pitr-state {
      config false;
      description
        "ITR state.";
      container learned-mappings {
        uses lisp:mappings;
        description
          "EID to RLOCs mappings cache.";
      }
```

```
      }
    }
  }
  <CODE ENDS>
```

8. LISP Proxy ETR Module

This module captures the configuration data model of a LISP Proxy ETR [RFC6832]. The model may also capture some operational data elements.

8.1. Module Structure

```
module: ietf-lisp-petr
augment /lisp:lisp/lisp:devices/lisp:device:
   +--rw proxy-etr-cfg!
      +--rw servicing-eids
         +--rw eid* [id]
            +--rw id              lisp:eid-id
```



```
            +--rw eid-address
               +--rw afi                    lisp-address-family-ref
               +--rw instance-id?           instance-id-type
               +--rw (address)?
                  +--:(ipv4)
                  |  +--rw ipv4?                 inet:ipv4-address
                  +--:(ipv6)
                  |  +--rw ipv6?                 inet:ipv6-address
                  +--:(mac)
                  |  +--rw mac?                  yang:mac-address
                  +--:(distinguished-name)
                  |  +--rw distinguished-name?   distinguished-name
                  +--:(lcaf)
                     +--rw lcaf
                        +--rw lcaf-type                lcaf-ref
                        +--rw (address)?
                           +--:(null-address)
                           |  +--rw null-address
                           |     +--rw address?   empty
                           +--:(afi-list)
                           |  +--rw afi-list
                           |     +--rw address-list*   simple-address
```

```
                    +--:(instance-id)
                    |  +--rw instance-id
                    |     +--rw instance-id?   instance-id-type
                    |     +--rw mask-length?   uint8
                    |     +--rw address?       simple-address
                    +--:(as-number)
                    |  +--rw as-number
                    |     +--rw as?        inet:as-number
                    |     +--rw address?   simple-address
                    +--:(application-data)
                    |  +--rw application-data
                    |     +--rw address?            simple-address
                    |     +--rw protocol?           uint8
                    |     +--rw ip-tos?             int32
                    |     +--rw local-port-low?     inet:port-number
                    |     +--rw local-port-high?    inet:port-number
                    |     +--rw remote-port-low?    inet:port-number
                    |     +--rw remote-port-high?   inet:port-number
                    +--:(geo-coordinates)
                    |  +--rw geo-coordinates
                    |     +--rw latitude?            bits
                    |     +--rw latitude-degrees?    uint8
                    |     +--rw latitude-minutes?    uint8
                    |     +--rw latitude-seconds?    uint8
                    |     +--rw longitude?           bits
                    |     +--rw longitude-degrees?   uint16
```



```
                    |     +--rw longitude-minutes?   uint8
                    |     +--rw longitude-seconds?   uint8
                    |     +--rw altitude?            int32
                    |     +--rw address?             simple-address
                    +--:(nat-traversal)
                    |  +--rw nat-traversal
                    |     +--rw ms-udp-port?        uint16
                    |     +--rw etr-udp-port?       uint16
                    |     +--rw global-etr-rloc?    simple-address
                    |     +--rw ms-rloc?            simple-address
                    |     +--rw private-etr-rloc?   simple-address
                    |     +--rw rtr-rlocs*          simple-address
                    +--:(explicit-locator-path)
                    |  +--rw explicit-locator-path
                    |     +--rw hop* [address]
```

```
                              |        +--rw address     simple-address
                              |        +--rw lrs-bits?   bits
                              +--:(sourc-dest-key)
                                 +--rw sourc-dest-key
                                    +--rw source?   inet:ip-prefix
                                    +--rw dest?     inet:ip-prefix
```

8.2. Module Definition

```
<CODE BEGINS> file "ietf-lisp-petr@2015-07-02.yang"
module ietf-lisp-petr {
  namespace "urn:ietf:params:xml:ns:yang:ietf-lisp-petr";
  prefix lisp-petr;
  import ietf-lisp {
    prefix lisp;
  }
  import ietf-lisp-address-types {
    prefix lcaf;
  }
  organization
   "IETF LISP (Locator/ID Separation Protocol) Working Group";
  contact
    "lisp@ietf.org";
  description
    "This YANG module defines the generic configuration
     data for a LISP PETR. The module can be extended by vendors to
     define vendor-specific configuration parameters and policies.

     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 2015-07-02 {
      description
        "Initial revision.";
      reference
        "https://tools.ietf.org/html/rfc6832";
    }
    identity petr {
      base lisp:lisp-device;
      description
        "LISP PETR.";
    }
    augment "/lisp:lisp/lisp:devices/lisp:device" {
      when "lisp:type = lisp-petr:petr" {
        description
          "Augment is valid when LISP device type is PETR.";
      }
      description
        "This augments LISP devices list with PETR specific
         parameters.";
      container proxy-etr-cfg {
        presence "LISP PETR operation enabled";
        description
          "Proxy ETR configuration parameters.";
        container servicing-eids {
          description
            "EIDs serviced by the PETR.";
          list eid {
            key "id";
            description
              "List of EIDs serviced by the PETR.";
            leaf id {
              type lisp:eid-id;
              description
                "Id of serviced EID.";
            }
            container eid-address {
              uses lcaf:lisp-address;
              description
                "Serviced EID in generic LISP address format.";
            }
```



```
          }
```

```
        }
      }
    }
  }
  <CODE ENDS>
```

9. LISP Address Types

This module captures the various LISP address types, and is an essential building block used in other LISP modules.

9.1. Module Definition

```
  <CODE BEGINS> file "ietf-lisp-address-types@2015-11-05.yang"
  module ietf-lisp-address-types {
    namespace "urn:ietf:params:xml:ns:yang:ietf-lisp-address-types";
    prefix laddr;
    import ietf-inet-types {
      prefix inet;
      //revision-date 2010-09-24;
    }
    import ietf-yang-types {
      prefix yang;
      //revision-date 2010-09-24;
    }
    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) 2014 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 2015-11-05 {
      description
        "Initial revision.";
      reference
        "http://tools.ietf.org/html/draft-ietf-lisp-lcaf-10";
    }
    identity lisp-address-family {
      description
        "Base identity from which identities describing LISP address
         families are derived.";
    }
    identity no-address-afi {
      base lisp-address-family;
      description
        "IANA Reserved.";
      }
    identity ipv4-afi {
      base lisp-address-family;
      description
        "IANA IPv4 address family.";
    }
    identity ipv4-prefix-afi {
      base lisp-address-family;
      description
        "IANA IPv4 address family prefix.";
    }
    identity ipv6-afi {
      base lisp-address-family;
      description
        "IANA IPv6 address family.";
    }
    identity ipv6-prefix-afi {
      base lisp-address-family;
      description
        "IANA IPv6 address family prefix.";
    }
    identity mac-afi {
      base lisp-address-family;
      description
        "IANA MAC address family.";
    }
    identity distinguished-name-afi {
      base lisp-address-family;
      description
        "IANA Distinguished Name address family.";
```

```
    }
    identity as-number-afi {
```



```
        base lisp-address-family;
        description
          "IANA AS Number address family.";
    }
    identity lcaf {
      base lisp-address-family;
      description
        "IANA LISP Canonical Address Format address family.";
    }
    identity null-address-lcaf {
      base lcaf;
      description
        "Null body LCAF type.";
    }
    identity afi-list-lcaf {
      base lcaf;
      description
        "AFI-List LCAF type.";
    }
    identity instance-id-lcaf {
      base lcaf;
      description
        "Instance-ID LCAF type.";
    }
    identity as-number-lcaf {
      base lcaf;
      description
        "AS Number LCAF type.";
    }
    identity application-data-lcaf {
      base lcaf;
      description
        "Application Data LCAF type.";
    }
    identity geo-coordinates-lcaf {
      base lcaf;
      description
        "Geo-coordinates LCAF type.";
    }
```

```
identity opaque-key-lcaf {
  base lcaf;
  description
    "Opaque Key LCAF type.";
}
identity nat-traversal-lcaf {
  base lcaf;
  description
    "NAT-Traversal LCAF type.";
```



```
}
identity nonce-locator-lcaf {
  base lcaf;
  description
    "Nonce-Locator LCAF type.";
}
identity multicast-info-lcaf {
  base lcaf;
  description
    "Multicast Info LCAF type.";
}
identity explicit-locator-path-lcaf {
  base lcaf;
  description
    "Explicit Locator Path LCAF type.";
}
identity security-key-lcaf {
  base lcaf;
  description
    "Security Key LCAF type.";
}
identity source-dest-key-lcaf {
  base lcaf;
  description
    "Source/Dest LCAF type.";
}
identity replication-list-lcaf {
  base lcaf;
  description
    "Replication-List LCAF type.";
}
identity json-data-model-lcaf {
```

```
  base lcaf;
  description
    "JSON Data Model LCAF type.";
}
identity key-value-address-lcaf {
  base lcaf;
  description
    "Key/Value Address LCAF type.";
}
identity encapsulation-format-lcaf {
  base lcaf;
  description
    "Encapsulation Format LCAF type.";
}
identity service-path-lcaf {
  base lcaf;
```



```
  description
    "Service Path LCAF type.";
}
typedef instance-id-type {
  type uint32 {
    range "0..16777215";
  }
  description
    "Defines the range of values for an Instance ID.";
}
typedef service-path-id-type {
  type uint32 {
    range "0..16777215";
  }
  description
    "Defines the range of values for a Service Path ID.";
}
 typedef distinguished-name-type {
  type string;
  description
    "Distinguished Name address.";
  reference
    "http://www.iana.org/assignments/address-family-numbers/
     address-family-numbers.xhtml";
}
```

```
  typedef simple-address {
    type union {
      type inet:ip-address;
      type inet:ip-prefix;
      type yang:mac-address;
      type distinguished-name-type;
      type inet:as-number;
    }
    description
      "Union of address types that can be part of LCAFs.";
  }

  typedef lisp-address-family-ref {
    type identityref {
      base lisp-address-family;
    }
    description
      "LISP address family reference.";
  }
  typedef lcaf-ref {
    type identityref {
      base lcaf;
    }
```



```
    description
      "LCAF types reference.";
  }

  grouping lisp-address {
    description
      "Generic LISP address.";
    leaf address-type {
      type lisp-address-family-ref;
      mandatory true;
      description
        "Type of the LISP address.";
    }
    leaf virtual-network-id {
      type instance-id-type;
      description
        "Virtual Network Identifier (instance-id) of the address.";
    }
```

```
      choice address {
        description
          "Various LISP address types, including IP, MAC, and LCAF.";

        leaf no-address {
          when "../../address-type = 'laddr:no-addr-afi'" {
            description
              "When AFI is 0.";
          }
          type empty;
          description
            "No address.";
        }
        leaf ipv4 {
          when "../../address-type = 'laddr:ipv4-afi'" {
            description
              "When AFI is IPv4.";
          }
          type inet:ipv4-address;
          description
            "IPv4 address.";
        }
        leaf ipv4-prefix {
          when "../../address-type = 'laddr:ipv4-prefix-afi'" {
            description
              "When AFI is IPv4.";
          }
          type inet:ipv4-prefix;
          description
            "IPv4 prefix.";
```



```
        }
        leaf ipv6 {
          when "../../address-type = 'laddr:ipv6-afi'" {
            description
              "When AFI is IPv6.";
          }
          type inet:ipv6-address;
          description
            "IPv6 address.";
        }
        leaf ipv6-prefix {
```

```
            when "../../address-type = 'laddr:ipv6-prefix-afi'" {
              description
                "When AFI is IPv6.";
              }
              type inet:ipv6-prefix;
              description
                "IPv6 address.";
            }
            leaf mac {
              when "../../address-type = 'laddr:mac-afi'" {
                description
                  "When AFI is MAC.";
              }
              type yang:mac-address;
              description
                "MAC address.";
            }
            leaf distinguished-name {
              when "../../address-type = 'laddr:distinguished-name-afi'" {
                description
                  "When AFI is distinguished-name.";
              }
              type distinguished-name-type;
              description
                "Distinguished Name address.";
            }
            leaf as-number {
              when "../../address-type = 'laddr:as-number-afi'" {
                description
                  "When AFI is as-number.";
              }
              type inet:as-number;
              description
                "AS Number.";
            }
            container null-address {
              when "../../address-type = 'null-address-lcaf'" {
```



```
                description
                  "When LCAF type is null.";
              }
              description
```

```
        "Null body LCAF type";
      leaf address {
        type empty;
        description
          "AFI address.";
      }
    }
    container afi-list {
      when "../../address-type = 'afi-list-lcaf'" {
        description
          "When LCAF type is AFI-List.";
      }
      description
        "AFI-List LCAF type.";
      reference
        "http://tools.ietf.org/html/draft-ietf-lisp-lcaf-10
         #section-4.16.1";
      leaf-list address-list {
        type simple-address;
        description
          "List of AFI addresses.";
      }
    }
    container instance-id {
      when "../../address-type = 'instance-id-lcaf'" {
        description
          "When LCAF type is Instance-ID";
      }
      description
        "Instance ID LCAF type.";
      reference
        "http://tools.ietf.org/html/draft-ietf-lisp-lcaf-10
         #section-4.2";
      leaf iid {
        type instance-id-type;
        description
          "Instance ID value.";
      }
      leaf mask-length {
          type uint8;
          description
            "Mask length.";
      }
      leaf address {
```

```
            type simple-address;
            description
              "AFI address.";
          }
        }
        container as-number-lcaf {
          when "../../address-type = 'as-number-lcaf'" {
            description
              "When LCAF type is AS-Number.";
          }
          description
            "AS Number LCAF type.";
          reference
            "http://tools.ietf.org/html/draft-ietf-lisp-lcaf-10
             #section-4.3";
          leaf as {
            type inet:as-number;
            description
              "AS number.";
          }
          leaf address {
            type simple-address;
            description
              "AFI address.";
          }
        }
        container application-data {
          when "../../address-type = 'application-data-lcaf'" {
            description
              "When LCAF type is Application Data.";
          }
          description
            "Application Data LCAF type.";
          reference
            "http://tools.ietf.org/html/draft-ietf-lisp-lcaf-10
             #section-4.4";
          leaf address {
              type simple-address;
              description
                "AFI address.";
          }
          leaf protocol {
              type uint8;
              description
                "Protocol number.";
          }
          leaf ip-tos {
```

```
            type int32;
```



```
            description
              "Type of service field.";
          }
          leaf local-port-low {
            type inet:port-number;
            description
              "Low end of local port range.";
          }
          leaf local-port-high {
            type inet:port-number;
            description
              "High end of local port range.";
          }
          leaf remote-port-low {
            type inet:port-number;
            description
              "Low end of remote port range.";
          }
          leaf remote-port-high {
            type inet:port-number;
            description
              "High end of remote port range.";
          }
        }
        container geo-coordinates {
          when "../../address-type = 'geo-coordinates-lcaf'" {
            description
              "When LCAF type is Geo-coordinates.";
          }
          description
            "Geo-coordinates LCAF type.";
          reference
            "http://tools.ietf.org/html/draft-ietf-lisp-lcaf-10
             #section-4.5";
          leaf latitude {
            type bits {
              bit N {
                description
                  "Latitude bit.";
              }
```

```
        }
        description
          "Bit that selects between North and South latitude.";
      }
      leaf latitude-degrees {
        type uint8 {
          range "0 .. 90";
        }
```



```
        description
          "Degrees of latitude.";
      }
      leaf latitude-minutes {
        type uint8 {
          range "0..59";
        }
        description
          "Minutes of latitude.";
      }
      leaf latitude-seconds {
        type uint8 {
          range "0..59";
        }
        description
          "Seconds of latitude.";
      }
      leaf longitude {
        type bits {
          bit E {
            description
              "Longitude bit.";
          }
        }
        description
          "Bit that selects between East and West longitude.";
      }
      leaf longitude-degrees {
        type uint16 {
          range "0 .. 180";
        }
        description
          "Degrees of longitude.";
```

```
          }
          leaf longitude-minutes {
            type uint8 {
              range "0..59";
            }
            description
              "Minutes of longitude.";
          }
          leaf longitude-seconds {
            type uint8 {
              range "0..59";
            }
            description
              "Seconds of longitude.";
          }
```



```
          leaf altitude {
            type int32;
            description
              "Height relative to sea level in meters.";
          }
          leaf address {
            type simple-address;
            description
              "AFI address.";
          }
        }
        container nat-traversal {
          when "../../address-type = 'nat-traversal-lcaf'" {
            description
              "When LCAF type is NAT-Traversal.";
          }
          description
            "NAT-Traversal LCAF type.";
          reference
            "http://tools.ietf.org/html/draft-ietf-lisp-lcaf-10
             #section-4.6";
          leaf ms-udp-port {
            type uint16;
            description
              "Map-Server UDP port (set to 4342).";
          }
```

```
            leaf etr-udp-port {
              type uint16;
              description
                "ETR UDP port.";
            }
            leaf global-etr-rloc {
              type simple-address;
              description
                "Global ETR RLOC address.";
            }
            leaf ms-rloc {
              type simple-address;
              description
                "Map-Server RLOC address.";
            }
            leaf private-etr-rloc {
              type simple-address;
              description
                "Private ETR RLOC address.";
            }
            leaf-list rtr-rlocs {
              type simple-address;
```



```
              description
                "List of RTR RLOC addresses.";
            }
          }
          container explicit-locator-path {
            when "../../address-type = 'explicit-locator-path-lcaf'" {
              description
                "When LCAF type type is Explicit Locator Path.";
            }
            description
              "Explicit Locator Path LCAF type.";
            reference
              "http://tools.ietf.org/html/draft-ietf-lisp-lcaf-10
               #section-4.9";
            list hop {
              key "hop-id";
              ordered-by user;
              description
                "List of locator hops forming the explicit path.";
```

```
            leaf hop-id {
              type string;
              description
                "Unique identifier for the hop.";
            }
            leaf address {
              type simple-address;
              description
                "AFI address.";
            }
            leaf lrs-bits {
              type bits{
                bit lookup {
                  description
                    "Lookup bit.";
                }
                bit rloc-probe {
                  description
                    "RLOC-probe bit.";
                }
                bit strict {
                  description
                    "Strict bit.";
                }
              }
              description
                "Flag bits per hop.";
            }
          }
```



```
        }
        container source-dest-key {
          when "../../address-type = 'source-dest-key-lcaf'" {
            description
              "When LCAF type type is Source/Dest.";
          }
          description
            "Source/Dest LCAF type.";
          reference
              "http://tools.ietf.org/html/draft-ietf-lisp-lcaf-10
              #section-4.11";
          leaf source {
```

```
            type simple-address;
            description
              "Source address.";
          }
          leaf dest {
            type simple-address;
            description
              "Destination address.";
          }
        }
        container key-value-address {
          when "../../address-type = 'key-value-address-lcaf'" {
            description
              "When LCAF type type is Key/Value Address.";
          }
          description
            "Key/Value Address LCAF type.";
          reference
            "http://tools.ietf.org/html/draft-ietf-lisp-lcaf-10
             #section-4.11";
          leaf key {
            type simple-address;
            description
              "Address as Key.";
          }
          leaf value {
            type simple-address;
            description
              "Address as Value.";
          }
        }
        container service-path {
          when "../../address-type = 'service-path-lcaf'" {
          description
            "When LCAF type service path identifier.";
              }
```



```
          description
            "Service Path LCAF type.";
          reference
            "http://tools.ietf.org/html/draft-ermagan-lisp-nsh-00";
          leaf service-path-id {
```

```
            type service-path-id-type;
            description
              "Service path identifier for the path for NSH header";
          }
          leaf service-index {
            type uint8;
            description
              "Service path index for NSH header";
          }
        }
      }
    }
  }
  <CODE ENDS>
```

10. Acknowledgments

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

11. IANA Considerations

This memo includes no request to IANA.

12. Security Considerations

Security Considerations TBD

13. Normative References

[I-D.ietf-lisp-ddt]
Fuller, V., Lewis, D., Ermagan, V., and A. Jain, "LISP Delegated Database Tree", draft-ietf-lisp-ddt-03 (work in progress), April 2015.

[I-D.ietf-lisp-lcaf]
Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format (LCAF)", draft-ietf-lisp-lcaf-11 (work in progress), September 2015.

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

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <http://www.rfc-editor.org/info/rfc6830>.

[RFC6832] Lewis, D., Meyer, D., Farinacci, D., and V. Fuller, "Interworking between Locator/ID Separation Protocol (LISP) and Non-LISP Sites", RFC 6832.

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

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

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

Carl Moberg
Cisco Systems
170 W Tasman Dr
San Jose, CA
USA

Email: camoberg@cisco.com

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