

YANG model for LISP

Vina Ermagan (vermagan@cisco.com)

Alberto Rodriguez-Natal (arnatal@ac.upc.edu)

Florin Coras (fcoras@ac.upc.edu)

Albert Cabellos (acabello@ac.upc.edu)

Fabio Maino (fmaino@cisco.com)

YANG model for LISP

- No model at the IETF so far
- YANG models at other WGs
- Note that “data model” is on the charter
 - MIB RFC exists
 - Eventually we’ll need a YANG RFC

First iteration (-00)

- Core components
- Basic features
- Augmentations possible

lisp-yang-00

- Two modules
- lisp.yang
 - Configuration model
- lisp-address-types.yang
 - Mostly LCAF

lisp-address-types.yang

```
grouping lisp-address {
  leaf afi {
    type enumeration {
      enum "ipv4" {
        value 1;
      }
      ...
    }
  }
  leaf instance-id {
    type instance-id-type;
  }
}
```

```
choice address {
  case ipv4 {
    when "afi = ipv4";
    leaf ipv4 {
      type inet:ipv4-address;
    }
  }
  case lcaf {
    when "afi = lcaf";
    container lcaf {
      uses lcaf-address;
    }
  }
}
```

lisp-address-types.yang

```
grouping lcaf-address {  
  leaf lcaf-type {  
    type enumeration {  
      enum "as-number";  
    }  
  }  
  choice address {  
    container as-number {  
      when "lcaf-type = as-number";  
      leaf as { type inet:as-number; }  
      leaf address { type simple-address; }  
    }  
  }  
  ... }  
}
```

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

lisp.yang

```
feature itr {  
  description  
    "ITR operation supported";  
}
```

```
feature etr {  
  description  
    "ETR operation supported";  
}
```

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

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

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

```
feature map-resolver {  
  description  
    "MR operation supported";  
}
```

lisp.yang

```
grouping locators {
  list rloc {
    choice address-type {
      case interface-address {
        leaf interface { type interface-name; }
      }
      case lisp-address {
        container locator-address {
          uses lcaf:lisp-address;
        }
      }
    }
    leaf priority { type uint8; }
    leaf weight { type uint8; }
  }
}
```

```
grouping mappings {
  list mapping {
    container eid { uses lcaf:lisp-address; }
    choice locator-list {
      case negative-mapping {
        leaf map-reply-action { type map-reply-action; }
      }
      case positive-mapping {
        container rlocs { uses locators; }
      }
      default "positive-mapping";
    }
  }
}
```


lisp.yang

```
container itr-cfg {
  if-feature itr;
  presence "LISP ITR operation enabled";

  container local-eids {
    list local-eid {
      min-elements 1;
      container eid-address {
        uses lcaf:lisp-address;
      }
    }
  }
}
```

```
container proxy-etr {
  list proxy-etr {
    container eid-address { uses lcaf:lisp-address; }
    leaf-list proxy-etr-address { type inet:ip-address; }
  }
}

container map-cache {
  uses mappings {
    augment "mapping" {
      leaf static {
        description "A configured mapping is a static mapping. If the
          mapping is learned, it is operational data and static is false.";
        type boolean; default "true";
      }
    }
  }
}
```

Running code

- NETCONF support at LISPmob
- Implements proposed YANG model
 - ... or part of it
 - ... for now
- Currently main ITR features supported
 - Ongoing work to support the rest of the model