Network Working Group D. Farinacci
Internet-Draft lispers.net
Intended status: Experimental C. Cantrell
Expires: July 14, 2018 Nexus
January 10, 2018

A Decent LISP Mapping System (LISP-Decent)
draft-farinacci-lisp-decent-00

Abstract

This draft describes how the LISP mapping system designed to be distributed for scale can also be decentralized for management and trust.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The LISP architecture and protocols [RFC6830] introduces two new numbering spaces, Endpoint Identifiers (EIDs) and Routing Locators (RLOCs) which is intended to provide overlay network functionality. To map from EID to a set or RLOCs, a control-plane mapping system are used [RFC6836] [RFC8111]. These mapping systems are distributed in nature in their deployment for scalability but are centrally managed by a third- party entity, namely a Mapping System Provider (MSP). The entities that use the mapping system, such as data-plane xTRs, depend on and trust the MSP. They do not participate in the mapping system other than to register and retrieve information to/from the mapping system [RFC6833].

This document introduces a Decentralized Mapping System (DMS) so the xTRs can participate in the mapping system as well as use it. They can trust each other rather than rely on third-party infrastructure. The xTRs act as Map-Servers to maintain distributed state for scale and reducing attack surface.

2. Definition of Terms

Decentralized Mapping System (DMS): is a mapping system entity that is not third-party to the xTR nodes that use it. The xTRs themselves are part of the mapping system. The state of the mapping system is fully distributed, decentralized, and the trust relies on the xTRs that use and participate in their own mapping system.

Mapping System Provider (MSP): is an infrastructure service that deploys LISP Map-Resolvers and Map-Servers [RFC6833] and possibly ALT-nodes [RFC6836] or DDT-nodes [RFC8111]. The MSP can be managed by a separate organization other than the one that manages xTRs. This model provides a business separation between who manages and is responsible for the control-plane versus who manages the data-plane overlay service.

Peer-Group: is a set of Map-Servers which are joined to the same multicast group that send and receive Map-Register messages addressed to the multicast group. Map-Resolvers can use the peer-group to resolve mappings by sending Map-Requests to the multicast group or to any member of the peer-group. Map-Resolvers can do a mapping system lookup for the peer-group multicast address to obtain members of the peer-group.

Core Peer-Group: is a set of Map-Servers and Map-Resolvers who are joined to a multicast group to bootstrap a multi-layer decentralized mapping system.

Replication List Entry (RLE): is an RLOC-record format that contains a list of RLOCs that an ITR replicates multicast packets on a multicast overlay. The RLE format is specified in [RFC8060].

Group Address EID: is an EID-record format that contains IPv4 (0.0.0.0/0, G) or IPv6 (0::/0, G) state. This state is encoded as a Multicast Info Type LCAF specified in [RFC8060]. Members of a peer-group send Map-Registers for (0.0.0.0/0, G) or (0::/0, G) with an RLOC-record that RLE encodes its RLOC address. Details are specified in [I-D.ietf-lisp-signal-free-multicast].

3. Overview

The clients of the Decentralized Mapping System (DMS) are also the providers of mapping state. Clients are typically ETRs that Map-Register EID-to-RLOC mapping state to the mapping database system. ITRs are clients in that they send Map-Requests to the mapping database system to obtain EID-to-RLOC mappings that are cached for data-plane use. When xTRs participate in a DMS, they are also acting as Map-Resolvers and Map-Servers using the protocol machinery defined in LISP control-plane specifications [RFC6833], [I-D.ietf-lisp-sec], and [I-D.farinacci-lisp-ecdsa-auth]. The xTRs are not required to run the database mapping transport system protocols specified in [RFC6836] or [RFC8111].

The xTRs are organized in a peer-group. The peer-group is identified by an IPv4 or IPv6 multicast group address. The xTRs join the same multicast group and receive LISP control-plane messages addressed to the group. Messages sent to the multicast group are distributed when the underlay network supports IP multicast [RFC6831] or is achieved with the overlay multicast mechanism described in [I-D.ietf-lisp-signal-free-multicast]. When overlay multicast is used and LISP Map-Register messages are sent to a peer-group, they are LISP data encapsulated with a instance-ID set to 0xffffff in the LISP header. The inner header of the encapsulated packet has the destination address set to the peer-group multicast group address and the outer header that is prepended has the destination address set to the RLOC of peer-group member. The members of the peer-group are kept in the LISP data-plane map-cache so packets for the peer-group can be replicated to each member RLOC.

All xTRs in a peer-group will store the same registered mappings and maintain the state as Map-Servers normally do. The peer-group members are not only receivers of the multicast group but also send packets to the group.

4. Components of a LISP-Decent xTR

When an xTR is configured to be a LISP-Decent xTR (or PxTR [RFC6832]), it runs the ITR, ETR, Map-Resolver, and Map-Server LISP network functions.

The following diagram shows 3 LISP-Decent xTRs joined to peer-group 224.1.1.1. When the ETR function of xTR1 originates a Map-Register, it is sent to all xTRs (including itself) synchronizing all 3 Map-Servers in xTR1, xTR2, and xTR3. The ITR function can populate its map-cache by sending a Map-Request locally to its Map-Resolver so it can replicate packets to each RLOC for EID 224.1.1.1.

```
                        xTR1
 Map-Request    +--------------------+
(always local)  |  +-----+  +-----+  |
   +---------------| ITR |  | ETR |--------------+
   |            |  +-----+  +-----+  |          |
   |            |                    |          |   Map-Register to EID
   |            |      +-------+     |          |  224.1.1.1 encapsulated to
   +------------------>| MR/MS |<---------------+  RLOCs xTR1, xTR2, and xTR3
                |      +-------+     |          |
                +--------------------+          |
                                                |
                           +--------------------+------------+
                           |                                 |
                           |                                 |
                +----------v---------+        +----------v---------+
                |     +--------+     |        |     +--------+     |
                |     |  MR/MS |     |        |     |  MR/MS |     |
                |     +--------+     |        |     +--------+     |
                |  +-----+  +-----+  |        |  +-----+  +-----+  |
                |  | ITR |  | ETR |  |        |  | ITR |  | ETR |  |
                |  +-----+  +-----+  |        |  +-----+  +-----+  |
                +--------------------+        +--------------------+
                         xTR2                          xTR3
```

Note if any external xTR would like to use a Map-Resolver from the peer-group, it only needs to have one of the LISP-Decent Map-Resolvers configured. By doing a looking to this Map-Resolver for EID 224.1.1,1, the external xTR could get the complete list of members for the peer-group.

For future study, an external xTR could multicast the Map-Request to 224.1.1.1 and either one of the LISP-Decent Map-Resolvers would return a Map-Reply or the external xTR is prepared to receive multiple Map-Replies.

5. No LISP Protocol Changes

There are no LISP protocol changes required to support this LISP-Decent specification. However, an implementation that sends Map-Register messages to a multicast group versus a specific Map-Server unicast address must change to call the data-plane component so the ITR functionality in the node can encapsulate the Map-Register as a unicast packet to each member of the peer-group.

An ITR SHOULD lookup its peer-group address periodically to determine if the membership has changed. The ITR can also use the pubsub capability documented in [I-D.rodrigueznatal-lisp-pubsub] to be notified when a new member joins or leaves the peer-group.

6. Configuration and Authentication

When xTRs are joined to a multicast peer-group, they must have their site registration configuration consistent. Any policy or authentication key material must be configured correctly and consistently among all members. When [I-D.farinacci-lisp-ecdsa-auth] is used to sign Map-Register messages, public-keys can be registered to the peer-group using the site authentication key mentioned above or using a different authentication key from the one used for registering EID records.

7. Core Peer-Group

A core peer-group multicast address can be preconfigured to bootstrap the decentralized mapping system. The group address (or DNS name that maps to a group address) can be explicitly configured in a few xTRs to start building up the mappings. Then as other xTRs come online, they can add themselves to the core peer-group by joining the peer-group multicast group.

Alternatively or additionally, new xTRs can join a new peer-group multicast group to form another layer of a decentralized mapping system. The group address and members of this new layer peer-group would be registered to the core peer-group address and stored in the core peer-group mapping system. Note each mapping system layer could have a specific function or a specific circle of trust.

This multi-layer mapping system can be illustrated:

```
              ___________               ---------
             /   core    \  224.2.2.2  / layer-1 \
            | peer-group | --------> |     I     |
            |  224.1.1.1 |           |    / \    |
             \___________/           |   J---K   |
                  |                   \_________/
                  | 224.3.3.3
                  |
                  v
              ---------
             / layer-2 \
            |     X     |
            |    / \    |
            |   Y---Z   |
             \_________/
```

Configured in xTRs A, B, and C (they make up the core peer-group):
  224.1.1.1 -> RLE: A, B, C

core peer-group DMS, mapping state in A, B, and C:
  224.2.2.2 -> RLE: I, J, K
  224.3.3.3 -> RLE: X, Y, Z

layer-1 peer-group DMS (inter-continental), mapping state in I, J, K:
   EID1 -> RLOCs: i(1), j(2)
   ...
   EIDn -> RLOCs: i(n), j(n)

layer-2 peer-group DMS (intra-continental), mapping sate in X, Y, Z::
   EIDa -> RLOCs: x(1), y(2)
   ...
   EIDz -> RLOCs: x(n), y(n)

The core peer-group multicast address 224.1.1.1 is configured in xTRs A, B and C so when each of them send Map-Register messages, they would all be able to maintain synchronized mapping state. Any EID can be registered to this DMS but in this example, peer-group multicast group EIDs are being registered only to find other peer-groups.

For example, lets say that xTR I boots up and it wants to find its other peers in its peer-group 224.2.2.2. Group address 224.2.2.2 is configured so xTR I knows what group to join for its peer-group. But xTR I needs a mapping system to register to, so the core peer-group is used and available to receive Map-Registers. The other xTRs J and

K in the peer-group do the same so when any of I, J or K needs to register EIDs, they can now send their Map-Register messages to group 224.2.2.2. Examples of EIDs being register are EID1 through EIDn shown above.

When Map-Registers are sent to group 224.2.2.2, they are encapsulated by the LISP data-plane by looking up EID 224.2.2.2 in the core peer-group mapping system. For the map-cache entry to be populated for 224.2.2.2, the data-plane must send a Map-Request so the RLOCs I, J, and K are cached for replication. To use the core peer-group mapping system, the data-plane must know of at least one of the RLOCs A, B, and/or C.

8. Security Considerations

Refer to the Security Considerations section of [I-D.ietf-lisp-rfc6833bis] for a complete list of security mechanisms as well as pointers to threat analysis drafts.

9. IANA Considerations

At this time there are no specific requests for IANA.

10. References

10.1. Normative References

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

[RFC6831] Farinacci, D., Meyer, D., Zwiebel, J., and S. Venaas, "The Locator/ID Separation Protocol (LISP) for Multicast Environments", RFC 6831, DOI 10.17487/RFC6831, January 2013, <https://www.rfc-editor.org/info/rfc6831>.

[RFC6832] Lewis, D., Meyer, D., Farinacci, D., and V. Fuller, "Interworking between Locator/ID Separation Protocol (LISP) and Non-LISP Sites", RFC 6832, DOI 10.17487/RFC6832, January 2013, <https://www.rfc-editor.org/info/rfc6832>.

[RFC6833] Fuller, V. and D. Farinacci, "Locator/ID Separation Protocol (LISP) Map-Server Interface", RFC 6833, DOI 10.17487/RFC6833, January 2013, <https://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, <https://www.rfc-editor.org/info/rfc6836>.

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

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

10.2. Informative References

[I-D.farinacci-lisp-ecdsa-auth]
Farinacci, D. and E. Nordmark, "LISP Control-Plane ECDSA Authentication and Authorization", draft-farinacci-lisp-ecdsa-auth-01 (work in progress), October 2017.

[I-D.ietf-lisp-rfc6833bis]
Fuller, V., Farinacci, D., and A. Cabellos-Aparicio, "Locator/ID Separation Protocol (LISP) Control-Plane", draft-ietf-lisp-rfc6833bis-07 (work in progress), December 2017.

[I-D.ietf-lisp-sec]
Maino, F., Ermagan, V., Cabellos-Aparicio, A., and D. Saucez, "LISP-Security (LISP-SEC)", draft-ietf-lisp-sec-14 (work in progress), October 2017.

[I-D.ietf-lisp-signal-free-multicast]
Moreno, V. and D. Farinacci, "Signal-Free LISP Multicast", draft-ietf-lisp-signal-free-multicast-07 (work in progress), November 2017.

[I-D.rodrigueznatal-lisp-pubsub]
Rodriguez-Natal, A., Ermagan, V., Leong, J., Maino, F., Cabellos-Aparicio, A., Barkai, S., Farinacci, D., Boucadair, M., Jacquenet, C., and s. stefano.secci@lip6.fr, "Publish/Subscribe Functionality for LISP", draft-rodrigueznatal-lisp-pubsub-01 (work in progress), October 2017.

Appendix A. Acknowledgments

The authors would like to thank the LISP WG for their review and acceptance of this draft.

The authors would also like to give a special thanks to Roman Shaposhnik for several discussions that occured before the first draft was published.

Appendix B. Document Change Log

[RFC Editor: Please delete this section on publication as RFC.]

B.1. Changes to draft-farinacci-lisp-decent

- o Initial draft posted January 2018.

Authors' Addresses

Dino Farinacci
lispers.net
San Jose, CA
USA

Email: farinacci@gmail.com

Colin Cantrell
Nexus
Scottsdale, AZ
USA

Email: colin@nexus.io

Network Working Group D. Farinacci
Internet-Draft lispers.net
Intended status: Experimental P. Pillay-Esnault
Expires: March 22, 2018 U. Chunduri
Huawei Technologies
September 18, 2017

LISP for the Mobile Network
draft-farinacci-lisp-mobile-network-02

Abstract

This specification describes how the LISP architecture and protocols can be used in a LTE/5G mobile network to support session survivable EID mobility. A recommendation is provided to SDOs on how to integrate LISP into the mobile network.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The LISP architecture and protocols [RFC6830] introduces two new numbering spaces, Endpoint Identifiers (EIDs) and Routing Locators (RLOCs) which provide an architecture to build overlays on top of the underlying Internet. Mapping EIDs to RLOC-sets is accomplished with a Mapping Database System. By using a level of indirection for routing and addressing, separating an address identifier from its location can allow flexible and scalable mobility. By assigning EIDs to mobile devices and RLOCs to the network nodes that support such mobile devices, LISP can provide seamless mobility.

For a reading audience unfamiliar with LISP, a brief tutorial level document is available at [I-D.ietf-lisp-introduction].

This specification will describe how LISP can be used to provide layer-3 mobility within and across an LTE [LTE401-3GPP] [LTE402-3GPP] and 5G [ARCH5G-3GPP] [PROC5G-3GPP] mobile network.

The following are the design requirements:

1. Layer-3 address mobility is provided within a mobile network RAN supported by a pGW region (intra-pGW) as well as across pGW regions (inter-pGW).

2. UE nodes can get layer-3 address mobility when roaming off the mobile network to support Fixed Mobile Convergence [FMC].

3. Transport layer session survivability exists while roaming within, across, and off of the mobile network.

4. No address management is required when UEs roam. EID addresses are assigned to UEs at subscription time. EIDs can be reassigned when UE ownership changes.

5. The design will make efficient use of radio resources thereby not adding extra headers to packets that traverse the RAN.

6. The design can support IPv4 unicast and multicast packet delivery and will support IPv6 unicast and multicast packet delivery.

7. The design will allow use of both the GTP [GTPv1-3GPP] [GTPv2-3GPP] and LISP [I-D.ietf-lisp-rfc6830bis] data-planes while using the LISP control-plane and mapping system.

8. The design can be used for either 4G/LTE and 5G mobile networks and may be able to support interworking between the different mobile networks.

9. The LISP architecture provides a level of indirection for routing and addressing. From a mobile operator's perspective, these mechanisms provide advantages and efficiencies for the URLLC, FMC, and mMTC use cases. See Section 2 for definitions and references of these use cases.

The goal of this specification is take advantage of LISP's non-disruptive incremental deployment benefits. This can be achieved by changing the fewest number of components in the mobile network. The proposal suggests adding LISP functionality only to eNodeB and pGW nodes. There are no hardware or software changes to the UE devices or the RF-based RAN to realize this architecture. The LISP mapping database system is deployed as an addition to the mobile network and does not require any coordination with existing management and provisioning systems.

Similar ID Oriented Networking (ION) mechanisms for the 5G [ARCH5G-3GPP] [PROC5G-3GPP] mobile network are also being considered in other standards organizations such as ETSI [ETSI-NGP] and ITU

[ITU-IMT2020]. The NGMN Alliance describes Locator/ID separation an enabler to meet Key Performance Indicator Requirements [NGMN].

2. Definition of Terms

xTR: Is a LISP node in the network that runs the LISP control-plane and data-plane protocols according to [I-D.ietf-lisp-rfc6830bis] and [I-D.ietf-lisp-rfc6833bis]. A formal definition of an xTR can be found in [RFC6830]. In this specification, a LISP xTR is a node that runs the LISP control-plane with the GTP data-plane.

EID: Is an Endpoint Identifier. EIDs are assigned to UEs and other Internet nodes in LISP sites. A formal definition of an EID can be found in [RFC6830].

UE EID: A UE can be assigned an IPv4 and/or an IPv6 address either statically, or dynamically as is the procedure in the mobile network today. These IP addresses are known as LISP EIDs and are registered to the LISP mapping system. These EIDs are used as the source address in packets that the UE originates.

RLOC: Is an Routing Locator. RLOCs are assigned to eNodeBs and pGWs and other LISP xTRs in LISP sites. A formal definition of an RLOC can be found in [RFC6830].

Mapping System: Is the LISP mapping database system that stores EID-to-RLOC mappings. The mapping system is centralized for use and distributed to scale and secure deployment. LISP Map-Register messages are used to publish mappings and LISP Map-Requests messages are used to lookup mappings. LISP Map-Reply messages are used to return mappings. EID-records are used as lookup keys, and RLOC-records are returned as a result of the lookup. Details can be found in [RFC6833].

LISP Control-Plane: In this specification, a LISP xTR runs the LISP control-plane which originates, consumes, and processes Map-Request, Map-Register, Map-Reply, and Map-Notify messages.

RAN: Radio Access Network where UE nodes connect to eNodeB nodes via radios to get access to the Internet.

EPC: Evolved Packet Core [EPS-3GPP] system is the part of the mobile network that allows the RAN to connect to a data packet network. The EPC is a term used for the 4G/LTE mobile network.

NGC: Next Generation Core [EPS-3GPP] system is the part of the 5G mobile network that allows the RAN to connect to a data packet network.

GTP: GTP [GTPv1-3GPP] [GTPv2-3GPP] is the UDP tunneling mechanism used in the LTE/4G and 5G mobile network.

UE: User Equipment as defined by [GPRS-3GPP] which is typically a mobile phone. The UE is connected to the network across the RAN to eNodeB nodes.

eNodeB: Is the device defined by [GPRS-3GPP] which borders the RAN and connects UEs to the EPC in a 4G/LTE mobile network. The eNodeB nodes are termination point for a GTP tunnel and are LISP xTRs. The equivalent term in the 5G mobile network is "(R)AN" and "5G-NR", or simply "gNB". In this document, the two terms are used interchangeably.

pGW: Is the PDN-Gateway as defined by [GPRS-3GPP] connects the EPC in a 4G/LTE mobile network to the Internet. The pGW nodes are termination point for a GTP tunnel and is a LISP xTR. The equivalent user/data-plane term in the 5G mobile network is the "UPF", which also has the capability to chain network functions. In this document, the two terms are used interchangeably.

URLLC: Ultra-Reliable and Low-Latency provided by the 5G mobile network for the shortest path between UEs [NGMN].

FMC: Fixed Mobile Convergence [FMC] is a term used that allows a UE device to move to and from the mobile network. By assigning a fixed EID to a UE device, LISP supports transport layer continuity between the mobile network and a fixed infrastructure such as a WiFi network.

mMTC: Massive Machine-Type Services [mMTC] is a term used to refer to using the mobile network for large-scale deployment of Internet of Things (IoT) applications.

3. Design Overview

LISP will provide layer-3 address mobility based on the procedures in [I-D.ietf-lisp-eid-mobility] where the EID and RLOCs are not co-located. In this design, the EID is assigned to the UE device and the RLOC(s) are assigned to eNodeB nodes. So any packets going to a UE are always encapsulated to the eNodeB that associates with the UE. For data flow from the UE to any EIDs (or destinations to non-LISP sites) that are outside of the EPC, use the RLOCs of the pGW nodes so the pGW can send packets into the Internet core (unencapsulated).

The following procedures are used to incorporate LISP in the EPC:

- o UEs are assigned EIDs. They usually never change. They identify the mobile device and are used for transport connections. If privacy for EIDs is desired, refer to details in [I-D.ietf-lisp-eid-anonymity].

- o eNodeB nodes are LISP xTRs. They have GTP, and optionally LISP, tunnels to the pGW nodes. The eNodeB is the RLOC for all EIDs assigned to UE devices that are attached to the eNodeB.

- o pGW nodes are LISP xTRs. They have GTP, and optionally LISP, tunnels to the eNodeB nodes. The pGW is the RLOC for all traffic destined for the Internet.

- o The LISP mapping system runs in the EPC. It maps EIDs to RLOC-sets.

- o Traffic from a UE to UE within a pGW region can be encapsulated from eNodeB to another eNodeB or via the pGW, acting as an RTR [RFC6830], to provide data-plane policy.

- o Traffic from a UE to UE across a pGW region have these options for data flow:

    1. Encapsulation by a eNodeB in one region to a eNodeB in another region.

    2. Encapsulation by a eNodeB in one region to a pGW in the same region and then the pGW reencapsulates to a eNodeB in another region.

    3. Encapsulation by a eNodeB in one region to a pGW in another region and then the pGW reencapsulates to a eNodeB in its same region

- Note when encapsulation happens between a eNodeB and a pGW, GTP is used as the data-plane and when encapsulation between two eNodeBs occur, LISP can be used as the data-plane when there is no X2 interface [X2-3GPP] between the eNodeB nodes.

- The pGW nodes register their RLOCs for a default EID-prefix to the LISP mapping system. This is done so eNodeB nodes can find pGW nodes to encapsulate to.

- The eNodeB nodes register EIDs to the mapping system for the UE nodes. The registration occurs when eNodeB nodes discover the layer-3 addresses of the UEs that connect to them. The eNodeB nodes register multiple RLOCs associated with the EIDs to get multi-homing and path diversity benefits from the EPC network.

- When a UE moves off a eNodeB, the eNodeB node deregisters itself as an RLOC for the EID associated with the UE.

- Optionally, and for further study for future architectures, the eNodeB or pGW could encapsulate to an xTR that is outside of the EPC network. They could encapsulate to a LISP CPE router at a branch office, a LISP top-of-rack router in a data center, a LISP wifi access-point, LISP border routers at a hub site, and even a LISP router running in a VM or container on a server.

The following diagram illustrates the LTE mobile network topology and structure [LTE401-3GPP] [LTE402-3GPP]:

```
                (--------------------------------------------)
                (                                            )
                (                  Internet                  )
                (                                            )
                (--------------------------------------------)
                          |                         |
                          |                         |
                (---------|---------)     (---------|---------)
                (        pGW        )     (        pGW        )
                (                   )     (                   )
                (        EPC        )     (        EPC        )
                (                   )     (                   )
                (  eNodeB   eNodeB  )     (  eNodeB   eNodeB  )
                (---/--\-----/--\---)     (---/--\-----/--\---)
                   /    \   /    \           /    \   /    \
                  /      \ /      \         /      \ /      \
                 /                 \       /                 \
                /        RAN        \     /        RAN        \
               /                     \   /                     \
              (   UE      UE      UE  ) (  UE        UE      UE  )
```

LTE/5G Mobile Network Architecture

The following diagram illustrates how LISP is used on the mobile network:

(1) IPv6 EIDs are assigned to UEs.
(2) RLOCs assigned to eNodeB nodes are [a1,a2], [b1,b2], [c1,c2], [d1,d2] on their uplink interfaces.
(3) RLOCs assigned to pGW nodes are [p1,p2], [p3,p4].
(4) RLOCs can be IPv4 or IPv6 addresses or mixed RLOC-sets.

```
             (--------------------------------------------)
             (                                            )
             (                  Internet                  )
             (                                            )
             (--------------------------------------------)
                       |                         |
                       |                         |
             (---------|---------)     (---------|---------)
             (        pGW        )     (        pGW        )
             (       p1 p2       )     (       p3 p4       )
             (                   )     (                   )
             (        EPC        )     (        EPC        )
             (                   )     (                   )
             (  a1  a2   b1  b2  )     (  c1  c2   d1  d2  )
             (  eNodeB   eNodeB  )     (  eNodeB   eNodeB  )
             (---/--\-----/--\---)     (---/--\-----/--\---)
                /    \   /    \           /    \   /    \
               /      \ /      \         /      \ /      \
              /                 \       /                 \
             /        RAN        \     /        RAN        \
            /                     \   /                     \
           (   UE      UE      UE  ) (  UE       UE      UE  )
    EIDs:     a::1    b::1    c::1     x::1     y::1    z::1
```

Mobile Network with EID/RLOC Assignment

The following table lists the EID-to-RLOC entries that reside in the LISP Mapping System when the above UEs are are attached to the 4 eNodeBs:

EID-Record	RLOC-Record	Commentary	Footnote
0::/0	[p1,p2,p3 p4]	eNodeBs encap to p1-p4 for Internet destinations which are non-EIDs	(1)
a::1/128	[a1,a2]	pGWs load-split traffic to [a1,a2] for UE a::1 and it can move to [b1,b2]	(2)
b::1/128	[a1,a2]	eNodeB tracks both UEs a::1 and b::1, it can do local routing between the UEs	(3)
c::1/128	[b1,b2]	UE c::1 can roam to [c1,c2] or [d1,d2], may use pGW [p1,p2] after move	(4)
x::1/128	[c1,c2]	UE x::1 can talk directly to UE y::1, eNodeBs encap to each other	(5)
y::1/128	[d1,d2]	UE can talk to Internet when [d1,d2], encap to pGW [p3,p4] or use backup [p1,p2]	(6)
z::1/128	[d1,d2]	UE z::1 can talk to a::1 directly where [d1,d2] encaps to [a1,a2]	(7)

(1) For packets that flow from UE nodes to destinations that are not in LISP sites, the eNodeB node use one of the RLOCs p1, p2, p3, or p4 as the destination address in the outer encapsulated header. Encapsulated packets are then routed by the EPC core to the pGW nodes. In turn, the pGW nodes, then route packets into the Internet core.

(2) Packets that arrive to pGW nodes from the Internet destined to UE nodes are encapsulated to one of the eNodeB RLOCs a1, a2, b1, b2. When UE, with EID a::1 is attached to the leftmost eNodeB, the EID a::1 is registered to the mapping system with RLOCs a1 and a2. When UE with EID c::1 is attached to the rightmost eNodeB (in the left region), the EID c::1 is registered to the mapping system with RLOCs b1 and b2.

(3) If UE with EID a::1 and UE with EID b::1 are attached to the same eNodeB node, the eNodeB node tracks what radio interface to use to route packets from one UE to the other.

(4) If UE with EID c::1 roams away from eNodeB with RLOCs b1 and b2, to the eNodeB with RLOCs c1 and c2 (in the rightmost region), packets destined toward the Internet, can use any pGW. Any packets that flow back from the Internet can use any pGW. In either case, the pGW is

informed by the mapping system that the UE with EID c::1 has new RLOCs and should now encapsulate to either RLOC c1 or c2.

(5) When UE with EID x::1 is attached to eNodeB with RLOCs c1 and c2 and UE with EID y::1 is attached to eNodeB with RLOCs d1 and d2, they can talk directly, on the shortest path to each eNodeB, when each encapsulate packets to each other's RLOCs.

(6) When packets from UE with EID y::1 are destined for the Internet, the eNodeB with RLOCs d1 and d2 that the UE is attached to can use any exit pGWs RLOCs p1, p2, p3, or p4.

(7) UE with EID z::1 can talk directory to UE with EID a::1 by each eNodeB they are attached to encapsulsates to each other's RLOCs. In case (5), the two eNodeB's were in the same region. In this case, the eNodeBs are in different regions.

The following abbreviated diagram shows a topology that illustrates how a UE roams with LISP across pGW regions:

```
                (--------------------------------------------)
                (                                            )
                (                  Internet                  )
                (                                            )
                (--------------------------------------------)
                          |                         |
                          |                         |
                (---------|---------)     (---------|---------)
                (        pGW        )     (        pGW        )
                (       p1 p2       )     (       p3 p4       )
                (                   )     (                   )
                (        EPC        )     (        EPC        )
                (                   )     (                   )
                (  a1  a2   b1  b2  )     (  c1  c2   d1  d2  )
                (  eNodeB   eNodeB  )     (  eNodeB   eNodeB  )
                (---/--\-----/--\---)     (---/--\-----/--\---)
                   /    \   /    \           /    \   /    \
                  /      \ /      \         /      \ /      \
                 /                 \       /                 \
                /        RAN        \     /        RAN        \
               /                     \   /                     \
              (   UE    ------------------------------>  UE     )
                 a::1                                   a::1
```

UE EID Mobility

The contents of the LISP mapping database before UE moves:

EID-Record	RLOC-Record	Commentary
0::/0	[p1,p2,p3,p4]	eNodeB [a1,a2] encaps to p1-p4 for Internet destinations when a::1 on eNodeB [a1,a2]
a::1/128	[a1,a2]	Before UE moves to other pGW region

The contents of the LISP mapping database after UE moves:

EID-Record	RLOC-Record	Commentary
0::/0	[p1,p2,p3,p4]	eNodeB [d1,d2] encaps to p1-p4 for Internet destinations when a::1 moves to eNodeB [d1,d2]
a::1/128	[d1,d2]	After UE moves to new pGW region

4. Addressing and Routing

UE based EID addresses will be IPv6 addresses. It will be determined at a future time what length the IPv6 prefix will be to cover all UEs in a mobile network. This coarse IPv6 prefix is called an EID-prefix where more-specific EID-prefixes will be allocated out of it for each pGW node. Each pGW node is responsible for advertising the more-specific EID-prefix into the Internet routing system so they can attract packets from non-EIDs nodes to UE EIDs.

An RLOC address will either be an IPv4 or IPv6 address depending on the support for single or dual-stack address-family in the EPC network. An RLOC-set in the mapping system can have a mixed address-family locator set. There is no requirement for the EPC to change to support one address-family or the other. And there is no requirement for the EPC network to support IPv4 multicast or IPv6 multicast. The LISP overlay will support both.

The only requirement for RLOC addresses is that they are routable in the EPC and the Internet core network.

The requirements of the LISP and GTP data-plane overlay is to support a layer-3 overlay network only. There is no architectural requirement to support layer-2 overlays. However, operators may want to provide a layer-2 LAN service over their mobile network. Details about how LISP supports layer-2 overlays can be found in [I-D.ietf-lisp-eid-mobility].

5. eNodeB LISP Functionality

The eNodeB node runs as a LISP xTR for control-plane functionality and runs GTP for data-plane functionality. Optionally, the LISP data-plane can be used to establish dynamic tunnels from one eNodeB node to another eNodeB node.

The eNodeB LISP xTR will follow the procedures of [I-D.ietf-lisp-eid-mobility] to discover UE based EIDs, track them by monitoring liveness, registering them when appear, and deregistering them when they move away. Since the eNodeB node is an xTR, it is acting as a layer-3 router and the GTP tunnel from the eNodeB node to the pGW node is realizing a layer-3 overlay. This will provide scaling benefits since broadcast and link-local multicast packets won’t have to travel across the EPC to the pGW node.

A day in the life of a UE originated packet:

1. The UE node originates an IP packet over the RAN.

2. The eNodeB receives the packet, extracts the source address from the packet, learns the UE based EID, stores its RAN location locally and registers the EID to the mapping system.

3. The eNodeB extracts the destination address, looks up the address in the mapping system. The lookup returns the RLOC of a pGW node if the destination is not an EID or an RLOC eNodeB node if the destination is a UE based EID.

4. The eNodeB node encapsulates the packet to the RLOC using GTP or optionally the LISP data-plane.

It is important to note that in [I-D.ietf-lisp-eid-mobility], EID discovery occurs when a LISP xTR receives an IP or ARP/ND packet. However, if there are other methods to discover the EID of a device, like in UE call setup, the learning and registration referenced in Paragraph 2 can happen before any packet is sent.

6. pGW LISP Functionality

The pGW node runs as a LISP xTR for control-plane functionality and runs GTP for data-plane functionality. Optionally, the LISP data-plane can be used to establish dynamic tunnels from one pGW node to another pGW or eNodeB node.

The pGW LISP xTR does not follow the EID mobility procedures of [I-D.ietf-lisp-eid-mobility] since it is not responsible for discovering UE based EIDs. A pGW LISP xTR simply follows the procedures of a PxTR in [RFC6830] and for interworking to non-EID sites in [RFC6832].

A day in the life of a pGW received packet:

1. The pGW node receives a IP packet from the Internet core.

2. The pGW node extracts the destination address from the packet and looks it up in the LISP mapping system. The lookup returns an RLOC of a eNodeB node. Optionally, the RLOC could be another pGW node.

3. The pGW node encapsulates the packet to the RLOC using GTP or optionally the LISP data-plane.

7. Compatible Data-Plane using GTP

Since GTP is a UDP based encapsulating tunnel protocol, it has the same benefits as LISP encapsulation. At this time, there appears to be no urgent need to not continue to use GTP for tunnels between a eNodeB nodes and between a eNodeB node and a pGW node.

There are differences between GTP tunneling and LISP tunneling. GTP tunnels are setup at call initiation time. LISP tunnels are dynamically encapsulating, used on demand, and don’t need setup or teardown. The two tunneling mechanisms are a hard state versus soft state tradeoff.

This specification recommends for early phases of deployment, to use GTP as the data-plane so a transition for it to use the LISP control-plane can be achieved more easily. At later phases, the LISP data-plane may be considered so a more dynamic way of using tunnels can be achieved to support URLLC.

This specification recommends the use of procedures from [I-D.ietf-lisp-eid-mobility] and NOT the use of LISP-MN [I-D.ietf-lisp-mn]. Using LISP-MN states that a LISP xTR reside on the mobile UE. This is to be avoided so extra encapsulation header overhead is NOT sent on the RAN. The LISP data-plane or control-plane will not run on the UE.

8. Roaming and Packet Loss

Using LISP for the data-plane has some advantages in terms of providing near-zero packet loss. In the current mobile network, packets are queued on the eNodeB node the UE is roaming to or rerouted on the eNodeB node the UE has left. In the LISP architecture, packets can be sent to multiple "roamed-from" and "roamed-to" nodes while the UE is moving or is off the RAN. See mechanisms in [I-D.ietf-lisp-predictive-rlocs] for details.

9. Mobile Network LISP Mapping System

The LISP mapping system stores and maintains EID-to-RLOC mappings. There are two mapping database transport systems that are available for scale, LISP-ALT [RFC6836] and LISP-DDT [RFC8111]. The mapping system will store EIDs assigned to UE nodes and the associated RLOCs assigned to eNodeB nodes and pGW nodes. The RLOC addresses are routable addresses by the EPC network.

This specification recommends the use of LISP-DDT.

10. Multicast Considerations

Since the mobile network runs the LISP control-plane, and the mapping system is available to support EIDs for unicast packet flow, it can also support multicast packet flow. Support for multicast can be provided by the LISP/GTP overlay with no changes to the EPC network.

Multicast (S-EID,G) entries can be stored and maintained in the same mapping database that is used to store UE based EIDs. Both Internet connected nodes, as well as UE nodes, can source multicast packets. The protocol procedures from [I-D.ietf-lisp-signal-free-multicast] are followed to make multicast delivery available. Both multicast packet flow and UE mobility can occur at the same time.

A day in the life of a 1-to-many multicast packet:

1. A UE node joins an (S,G) multicast flow by using IGMPv2 or IGMPv3.

2. The eNodeB node records which UE on the RAN should get packets sourced by S and destined for group G.

3. The eNodeB node registers the (S,G) entry to the mapping system with its RLOC according to the receiver site procedures in [I-D.ietf-lisp-signal-free-multicast]. The eNodeB does this to show interest in joining the multicast flow.

4. When other UE nodes join the same (S,G), their associated eNodeB nodes will follow the procedures in steps 1 through 3.

5. The (S,G) entry stored in the mapping database has an RLOC-set which contains a replication list of all the eNodeB RLOCs that registered.

6. A multicast packet from source S to destination group G arrives at the pGW. The pGW node looks up (S,G), gets returned the replication list of all joined eNodeB nodes and replicates the multicast packet by encapsulating the packet to each of them.

7. Each eNodeB node decapsulates the packet and delivers the multicast packet to one or more IGMP-joined UEs on the RAN.

11. Security Considerations

For control-plane authentication and authorization procedures, this specification recommends the mechanisms in [I-D.ietf-lisp-rfc6833bis], LISP-SEC [I-D.ietf-lisp-sec] AND LISP-ECDSA [I-D.farinacci-lisp-ecdsa-auth].

For data-plane privacy procedures, this specification recommends the mechanisms in [RFC8061] When the LISP data-plane is used. otherwise, the EPC must provide data-plane encryption support.

12. IANA Considerations

There are no specific requests for IANA.

13. SDO Recommendations

The authors request other Standards Development Organizations to consider LISP as a technology for device mobility. It is recommended to start with this specification as a basis for design and develop more deployment details in the appropriate Standards Organizations. The authors are willing to facilitate this activity.

14. References

14.1. Normative References

[RFC1700] Reynolds, J. and J. Postel, "Assigned Numbers", RFC 1700, DOI 10.17487/RFC1700, October 1994, <https://www.rfc-editor.org/info/rfc1700>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <https://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, DOI 10.17487/RFC6832, January 2013, <https://www.rfc-editor.org/info/rfc6832>.

[RFC6833] Fuller, V. and D. Farinacci, "Locator/ID Separation Protocol (LISP) Map-Server Interface", RFC 6833, DOI 10.17487/RFC6833, January 2013, <https://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, <https://www.rfc-editor.org/info/rfc6836>.

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

[RFC8061] Farinacci, D. and B. Weis, "Locator/ID Separation Protocol (LISP) Data-Plane Confidentiality", RFC 8061, DOI 10.17487/RFC8061, February 2017, <https://www.rfc-editor.org/info/rfc8061>.

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

14.2. Informative References

[ARCH5G-3GPP] 3GPP, "System Architecture for the 5G System", TS.23.501 https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144, December 2016.

[EPS-3GPP] 3GPP, "Non-Access-Stratum (NAS) Protocol for Evolved Packet System (EPS); Stage 3", TS.23.501 https://portal.3gpp.org/desktopmodules/specifications/specificationdetails.aspx?specificationid=1072, January 2015.

[ETSI-NGP] ETSI-NGP, "NGP Evolved Architecture for mobility using Identity Oriented Networks", NGP-004, version 0.0.3 https://portal.etsi.org/webapp/WorkProgram/Report_WorkItem.asp?WKI_ID=50531, May 2017.

[FMC] ipv6.com, "FIXED MOBILE CONVERGENCE", https://www.ipv6.com/mobile/fixed-mobile-convergence/, November 2006.

[GPRS-3GPP] 3GPP, "General Packet Radio Service (GPRS) for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) Access", TS23.401 Release 8 https://portal.3gpp.org/desktopmodules/specifications/specificationdetails.aspx?specificationid=849, January 2015.

[GTPv1-3GPP]
3GPP, "General Packet Radio System (GPRS) Tunnelling Protocol User Plane (GTPv1-U)", TS.29.281 https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1699, January 2015.

[GTPv2-3GPP]
3GPP, "3GPP Evolved Packet System (EPS); Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C); Stage 3", TS.29.274 https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1692, January 2015.

[I-D.farinacci-lisp-ecdsa-auth]
Farinacci, D. and E. Nordmark, "LISP Control-Plane ECDSA Authentication and Authorization", draft-farinacci-lisp-ecdsa-auth-00 (work in progress), July 2017.

[I-D.ietf-lisp-eid-anonymity]
Farinacci, D., Pillay-Esnault, P., and W. Haddad, "LISP EID Anonymity", draft-ietf-lisp-eid-anonymity-00 (work in progress), August 2017.

[I-D.ietf-lisp-eid-mobility]
Portoles-Comeras, M., Ashtaputre, V., Moreno, V., Maino, F., and D. Farinacci, "LISP L2/L3 EID Mobility Using a Unified Control Plane", draft-ietf-lisp-eid-mobility-00 (work in progress), May 2017.

[I-D.ietf-lisp-introduction]
Cabellos-Aparicio, A. and D. Saucez, "An Architectural Introduction to the Locator/ID Separation Protocol (LISP)", draft-ietf-lisp-introduction-13 (work in progress), April 2015.

[I-D.ietf-lisp-mn]
Farinacci, D., Lewis, D., Meyer, D., and C. White, "LISP Mobile Node", draft-ietf-lisp-mn-00 (work in progress), April 2017.

[I-D.ietf-lisp-predictive-rlocs]
Farinacci, D. and P. Pillay-Esnault, "LISP Predictive RLOCs", draft-ietf-lisp-predictive-rlocs-00 (work in progress), June 2017.

[I-D.ietf-lisp-rfc6830bis]
Farinacci, D., Fuller, V., Meyer, D., Lewis, D., and A. Cabellos-Aparicio, "The Locator/ID Separation Protocol (LISP)", draft-ietf-lisp-rfc6830bis-05 (work in progress), August 2017.

[I-D.ietf-lisp-rfc6833bis]
Fuller, V., Farinacci, D., and A. Cabellos-Aparicio, "Locator/ID Separation Protocol (LISP) Control-Plane", draft-ietf-lisp-rfc6833bis-05 (work in progress), May 2017.

[I-D.ietf-lisp-sec]
Maino, F., Ermagan, V., Cabellos-Aparicio, A., and D. Saucez, "LISP-Security (LISP-SEC)", draft-ietf-lisp-sec-12 (work in progress), November 2016.

[I-D.ietf-lisp-signal-free-multicast]
Moreno, V. and D. Farinacci, "Signal-Free LISP Multicast", draft-ietf-lisp-signal-free-multicast-06 (work in progress), August 2017.

[ITU-IMT2020]
ITU-FG, "Focus Group on IMT-2020", https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.687-2-199702-I!!PDF-E.pdf.

[LTE401-3GPP]
3GPP, "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access", TS.23.401 https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=849, January 2015.

[LTE402-3GPP]
3GPP, "Architecture enhancements for non-3GPP accesses", TS.23.402 https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=850, January 2015.

[mMTC] NGMN Alliance, "NGMN KPIs and Deployment Scenarios for Consideration for IMT2020", https://www.ngmn.org/uploads/media/151204_NGMN_KPIs_and_Deployment_Scenarios_for_Consideration_for_IMT_2020_-_LS_Annex_V1_approved.pdf, December 2015.

[NGMN] NGMN Alliance, "5G End-to-End Architecture Framework", NGMN https://www.ngmn.org/uploads/media/170511_NGMN_E2EArchFramework_v0.6.5.pdf, March 2016.

[PROC5G-3GPP] 3GPP, "Procedures for the 5G System", TS.23.502 https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3145, December 2016.

[X2-3GPP] 3GPP, "Evolved Universal Terrestrial Radio Access Network (E-UTRAN); X2 Application Protocol (X2AP)", TS.36.423 https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2452, June 2017.

Appendix A. Acknowledgments

The authors would like to thank Gerry Foster and Peter Ashwood Smith for their expertise with 3GPP mobile networks and for their early review and contributions. The authors would also like to thank Fabio Maino, Malcolm Smith, and Marc Portoles for their expertise in both 5G and LISP as well as for their early review comments.

The authors would like to give a special thank you to Ryosuke Kurebayashi from NTT Docomo for his operational and practical commentary.

Appendix B. Document Change Log

B.1. Changes to draft-farinacci-lisp-mobile-network-02.txt

o Posted mid September 2017.

o Editorial fixes from draft -01.

B.2. Changes to draft-farinacci-lisp-mobile-network-01.txt

o Posted September 2017.

o Explain each EID case illustrated in the "Mobile Network with EID/RLOC Assignment" diagram.

o Make a reference to mMTC as a 3GPP use-case for 5G.

o Add to the requirements section how mobile operators believe that using Locator/ID separation mechanisms provide for more efficient mobile netwowks.

o Indicate that L2-overlays is not recommended by this specification as the LISP mobile network architeture but how operators may want to deploy a layer-2 overlay service.

B.3. Changes to draft-farinacci-lisp-mobile-network-00.txt

o Initial draft posted August 2017.

Authors' Addresses

Dino Farinacci
lispers.net
San Jose, CA
USA

Email: farinacci@gmail.com

Padma Pillay-Esnault
Huawei Technologies
Santa Clara, CA
USA

Email: padma@huawei.com

Uma Chunduri
Huawei Technologies
Santa Clara, CA
USA

Email: uma.chunduri@huawei.com

B. Haindl
M. Lindner
Frequentis
R. Rahman
M. Portoles
V. Moreno
F. Maino
Cisco Systems

December 5, 2017

Ground-Based LISP for the Aeronautical Telecommunications Network

draft-haindl-lisp-gb-atn-00

Abstract

This document describes the use of the LISP architecture and protocols to address the requirements of the worldwide Aeronautical Telecommunications Network with Internet Protocol Services, as articulated by the International Civil Aviation Organization.

The ground-based LISP overlay provides mobility and multi-homing services to the IPv6 networks hosted on commercial aircrafts, to support Air Traffic Management communications with Air Traffic Controllers and Air Operation Controllers. The proposed architecture doesn’t require support for LISP protocol in the airborne routers, and can be easily deployed over existing ground infrastructures.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Status of This Memo

This Internet-Draft will expire on June 8, 2018.

Copyright Notice

Table of Contents

1. Introduction

This document describes the use of the LISP [RFC6830] architecture and protocols to address the requirements of the worldwide Aeronautical Telecommunications Network with Internet Protocol Services (ATN/IPS), as articulated by the International Civil Aviation Organization (ICAO).

ICAO is proposing to replace the existing aeronautical communication services with an IPv6 based infrastructure that supports Air Traffic Management (ATM) between commercial aircrafts, Air Traffic Controllers (ATC) and Air Operation Controllers (AOC).

This document describes how a LISP overlay can be used to offer mobility and multi-homing services to the IPv6 networks hosted on commercial aircrafts without requiring LISP support in the airborne routers. Use of the LISP protocol is limited to the ground-based routers, hence the name "ground-based LISP". The material for this document is derived from [GBL].

2. Definition of Terms

AOC: Airline Operational Control

ATN/IPS: Aeronautical Telecommunications Network with Internet Protocol Services

AC-R: Access Ground Router

A/G-R: Air/Ground Router

G/G-R: Ground/Ground Router

A-R: Airborne Router

A-E: Airborne Endsystem

ATS-E: ATS Endsystem

For definitions of other terms, notably Map-Register, Map-Request, Map-Reply, Routing Locator (RLOC), Solicit-Map-Request (SMR), Ingress Tunnel Router (ITR), Egress Tunnel Router (ETR), xTR (ITR or ETR), Map-Server (MS), and Map-Resolver (MR) please consult the LISP specification [RFC6830].

3. Design Overview

In the ATN/IPS architecture the airborne endsystems hosted on an aircraft are part of an IPV6 network connected to the ground network by one or more Airborne Routers (A-R). A-Rs have multiple radio interfaces that connects them via various radios infrastructures (e.g. SATCOM, LDACS, AeroMACS) to a given radio region, also known as subnetwork, on the ground. Typically an A-R has a corresponding ground based Access Router (AC-R) that terminates the radio protocol with the A-R and provides access services to the ground based portion of the radio network infrastructure. Each radio region is interconnected with the ATN/IPS ground network via an Air-to-Ground router (AG-R).

Similarly, the Air Traffic Controllers and Air Operation Controllers Endsystems (ATS-E and AOC-E) are part of IPv6 networks reachable via one or more Ground-to-Ground Routers (G/G-Rs).

The ATN/IPS ground network infrastructure is the internetworking region located between the A/G routers and the G/G routers.

In the ground-based LISP architecture, a LISP overlay is laid over the ATN/IPS internetworking region (that is in the LISP RLOC space) and provides connectivity between endsystems (that are in the LISP EID space) hosted in the aircrafts and in the AOC/ATS regions. The A/G-Rs and the G/G-Rs assume the role of LISP xTRs supported by a LISP mapping system infrastructure.

```
                                                 ,------,
                                 ,---------.    : A-E1 :
                               ,'           './'------'
                              (    AIRCRAFT   )
                               '.  +-----+  ,'\ ,------,
                                 '-| A-R |-'   \: A-E2 :
                                   +-----+      '------'
                                 //       \\
                                //         \\
                         +---+--+           +-+--+--+
                    .--.-.| AC-R1|'.-.     .| AC-R2 |.-.
                   (      +---+--+    )   ( +-+--+--+   )
                  (                 __.  (             '.
                ..'SATCOM Region  )     .' LDACS Region  )
               (              .'-'     (              .'-'
                '  +-------+  )         '  +-------+   )
                 '-| A/G-R |-'           '-| A/G-R |-''
                   |       |               |       |
                   |  xTR1 |               | xTR2  |
                   +-------+               +-------+
                         \                       /
                          \    .--..--. .--. ../
                           \  (    '         '.--.
                           .-.'  Internetworking   ''   '-------'
                          (          region          )--: MS/MR :
                           (     (RLOC SPACE)    '-''   '-------'
                            ._.'--'._.'.-._.'.-._)
                              /                    \
                         +---+---+                +-+--+--+
                     -.-.| G/G-R |'.             .| G/G-R |.
                    (    |       |  )           ( |       | )
                   (     |  xTR3 |   )         (  | xTR4  |  )
                  (      +---+---+    )       (   +-+--+--+   )
                 (                 _._.      (                '.
                ..'  ATS Region   )         .'   AOC Region  )
               (              .'-'          (              .'-'
                '--'._.'.    )\              '--'._.'.    )\
                 /       '--'  \              /      '--'  \
              '--------'    '--------'   '--------'   '--------'
              : ATS-E1 :    : ATS-E2 :   : AOC-E1 :   : AOC-E2 :
              '--------'    '--------'   '--------'   '--------'
```

Figure 1: ATN/IPS and ground-based LISP overlay

Endsystems in the AOC/ATS regions are mapped in the LISP overlay by the G/G-Rs, that are responsible for the registration of the AOC/ATS endsystems to the LISP mapping system. Each G/G-R is basically an xTR which has direct connections only to the terrestrial regions, i.e. no direct connection to the radio regions.

Aircrafts will attach to a specific radio region, via the radio interfaces of the A-Rs. How the radio attachment works is specific to each particular radio infrastructure, and out of the scope of this document, see [GBL].

Typically at the end of the attachment phase, the access router (AC-R) corresponding to the A-R, will announce the reachability of the EID prefixes corresponding to the attached aircraft (the announcement is specific to each particular radio infrastructure, and is out of the scope of this document). A/G-Rs in that particular radio region are responsible to detect those announcements, and, since they act as xTRs, register to the LISP mapping systems the corresponding IPv6 EID prefixes on behalf of the A-R, but with the RLOC of the A/G-R.

The EID prefixes registered by the A/G-Rs are then reachable by any of the AOC/ATS Endsystems that are part of the ground based LISP overlay.

The LISP infrastructure is used to support seamless aircraft mobility from one radio network to another, as well as multi-homing attachment of an aircraft to multiple radio networks with use of LISP weight and priorities to load balance traffic directed toward the aircraft.

The rest of this document provides further details on how ground-based LISP is used to address the requirements of the ATN/IPS use cases. The main design goals are:

- minimize added complexity on the aircraft
    * airborne routers can assume that any ground system is reachable via any A/G router. Static routing policies can be used on board
    * no need for routing/mobility protocols on board. Routing/mobility is managed on the ground ATN/IPS network
    * on-board outgoing link selection can be done with simple static policy
- seamless support for aircraft mobility and multi-homing with minimal traffic overhead on the A/G datalink
- minimize complexity of ground deployment

* ground-based LISP can be easily deployed over existing ATN/IPS ground infrastructure

* it is based on COTS solutions

* can ease IPv4 to IPv6 transition issues

4. Basic Protocol Operation

Figure 1 provides the reference topology for a description of the basic operation. A more detailed description of the basic protocol operation is described in [GBL].

4.1. Endsystem Registration

The following are the steps via which airborne endsystem prefixes are registered with the LISP mapping system:

1. Each Airborne Endsystem (A-E) is assigned an IPv6 address that is the endsystem EID. Each EID includes a Network-ID prefix that comprises (1) an ICAO ID which uniquely identifies the aircraft, and possibly (2) an aircraft network identifier. Airborne devices are grouped in one (and possibly several) IPv6 EID prefixes. As an example an IPv6 EID prefix could be used for all ATC applications located in a safety critical domain of the aircraft network, another IPv6 EID prefix could be used for AOC applications located in a less safety critical domain.

2. After the Airborne Router (A-R) on an aircraft attaches to one radio region, the corresponding Access Router (AC-R) learns the IPv6 EID prefixes belonging to the aircraft. The AC-R also announces reachability of these prefixes in the radio region (subnetwork) e.g. by using an IGP protocol like OSPF. The attachment to a radio includes a preference parameter and a quality parameter, these parameters are used e.g. to calculate the IGP reachability advertisement metric.

3. The Air/Ground Router (A/G-R) in the subnetwork receives the radio region announcements which contain reachablity information for the IPv6 EID prefixes corresponding to the Airborne Endsystems. Since each A/G-R is also an xTR, the A/G-R registers the IPv6 EID prefixes with the LISP MS/MR on behalf of the A-R, but with the RLOC of the A/G-R. The included quality parameter (e.g. IGP metric) is converted to a LISP priority, so that a lower quality metric results in a lower LISP priority value.

Ground based endsystems are part of ground subnetworks where the Ground/Ground Router (G/G-R) is an xTR. Each G/G-R therefore registers the prefixes corresponding to the AOC endsystems and ATS endsystems with the LISP mapping system, as specified in [RFC6830].

4.2. Ground to Airborne Traffic Flow

Here is an example of how traffic flows from the ground to the airborne endsystems, when ATS endsystem 1 (ATS-E1) has traffic destined to airborne endsystem 1 (A-E1):

1. The default route in the ATS region takes the traffic to xTR3 which is also a Ground/Ground Router (G/G-R).

2. xTR3 sends a Map-Request message for the address of A-E1 to the LISP mapping system. xTR2 sends a Map-Reply to xTR3 with RLOC set to its address which is reachable from xTR3 via the internetworking region.

3. xTR3 encapsulates the traffic to xTR2 using the RLOC information in the Map-Reply message.

4. xTR2 decapsulates the traffic coming from xTR3. The destination address of the inner packet belongs to A-E1 which has been advertised by the AC-R in the same region. The traffic is therefore forwarded to AC-R2.

5. AC-R2 sends the traffic to the Airborne Router of the aircraft and the A-R sends it to the endsystem.

4.3. Airborne to Ground Traffic Flow

Here is an example of how traffic flows from the airborne endsystems to the ground when airborne endsystem 2 (A-E2) has traffic destined to ATS endsystem 2 (ATS-E2):

1. The default route in the aircraft points to the Airborne Router (A-R). The latter forwards the traffic over the radio link to AC-R2.

2. The default route on AC-R2 points to xTR2 (also an A/G-R), so the traffic is sent from AC-R2 to xTR2.

3. xTR2 sends a Map-Request message for the address of ATS-E2 to the LISP mapping system. xTR3 sends a Map-Reply to xTR2 with RLOC set to its address which is reachable from xTR2 via the internetworking region.

4. xTR2 encapsulates the traffic to xTR3 using the RLOC information in the Map-Reply message.

5. xTR3 decapsulates the traffic coming from xTR2, and forwards it to ATS-E2.

4.4. Default forwarding path

When an xTR is waiting for a Map-Reply for an EID, the xTR does not know how to forward the packets destined to that EID. This means that the first packets for ground-to-air traffic would get dropped until the Map-Reply is received and a map-cache entry is created. However if a device acting as RTR, see [I-D.ermagan-lisp-nat-traversal], has mappings for all EIDs, the xTR could use the RTR as default path for packets which have to be encapsulated. How the RTR gets all the mappings is outside the scope of this document but one example is the use of LISP pub-sub as specified in [I-D.rodrigueznatal-lisp-pubsub]. Note that the RTR does not have to be a new device, the device which has the MS/MR role can also act as RTR. It is only the RTR which needs to subscribe to all the aircraft EIDs, the XTRs (i.e. the A/G-Rs and G/G-Rs) do not need to subscribe.

4.5. Traffic symmetry

The requirements for traffic symmetry are still TBD.

5. Multi-Homing and Mobility

Multi-homing support builds on the procedures described in Section 4:

1. The Airborne Router (A-R) on an aircraft attaches to multiple radio regions. As an example, and referring to Figure 1, the A-R attaches to the LDACS and SATCOM regions, via AC-R2 and AC-R1 respectively.

2. Through the preference parameter sent to each region, the A-R has control over which path (i.e. radio region) ground to air traffic flows. For example, A-R would indicate preference of the LDACS region by choosing a better preference value for the LDACS region compared to the preference value sent to the SATCOM region.

3. Both xTR1 and xTR2 register the IPv6 EID prefixes with the LISP mapping system using merge semantic, as specified in section 4.6 of [I-D.ietf-lisp-rfc6833bis]. Since the priority used in the LISP registrations is derived from the preference and quality parameters, xTR2 would use a lower priority value than xTR1. In this way the LISP mapping system will favour xTR2 (A/G-R for the LDACS region) over xTR1 (A/G-R for the SATCOM region), as specified by the preference and quality parameters.

4. Upon registration the LISP MS/MR will send Map-Notify messages to both xTR1 and xTR2, to inform that they have reachability to the aircraft's IPv6 EID prefixes. Both xTRs are notified because they have both set the merge-request and want-map-notify bits in their respective Map-Register message.

5. Upstream and downstream traffic flows on the same path, i.e. both use the LDACS region.

With mobility, the aircraft could want to switch traffic from one radio link to another. For example while transiting from an area covered by LDACS to an area covered by SATCOM, the aircraft could desire to switch all traffic from LDACS to SATCOM. For air-to-ground traffic, the A-R has complete control over which radio link to use, and will simply select the SATCOM outgoing interface. For ground-to-air traffic:

1. The A-R sends a radio advertisement to AC-R1 indicating a better preference for the SATCOM link.

2. This leads to AC-R1 lowering its quality parameter (e.g. IGP metric) for the IPv6 EID prefixes.

3. Upon receiving the better preference value, xTR1 registers the IPv6 EID prefixes with the MS/MR, using a lower priority value than what xTR2 had used. Both xTR1 and xTR2 receives Map-Notify messages signaling to xTR2 that xTR1 is now the preferred path toward the aircraft.

4. xTR3 has a map-cache which still points to xTR2, therefore xTR3 still sends traffic via xTR2. xTR2 sends Solicit-Map-Request (SMR) to xTR3 who queries the LISP mapping system again. This results in updating the map-cache on xTR3 which now points to XTR1 so ground-to-air traffic now flows on the SATCOM radio link.

The procedure for mobility is derived from [I-D.ietf-lisp-eid-mobility].

6. Convergence

When traffic is flowing on a radio link and that link goes down, the network has to converge rapidly on the other link available for that aircraft.

For air-to-ground traffic, once the A-R detects the failure it can switch immediatly to the other radio link.

For ground-to-air traffic, when a radio link fails, the corresponding AC-R sends a reachability update that the IPv6 EID prefixes are not reachable anymore. This leads to the A/G-R (also an xTR) in that region to unregister the IPv6 EID prefixes with the MS/MR. This indicates that the xTR in question has no reachability to the EID prefixes. The notification of the failure should reach all relevant xTRs as soon as possible. For example, if the LDACS radio link fails, xTR3 and xTR4 need to learn about the failure so that they stop sending traffic via xTR2 and use xTR1 instead.

In the sub-sections below, we the use of RLOC-probing, Solicit-Map-Request, and LISP pub-sub as alternative mechanisms for link failure notification.

6.1. Use of RLOC-probing

RLOC-probing is described in section 6.3.2 of [RFC6830].

At regular intervals xTR3 sends Map-Request to xTR2 for the aircraft’s EID prefixes. When xTR3 detects via RLOC-probing that it can not use xTR2 anymore, it sends a Map-Request for the aircraft’s EID prefixes. The corresponding Map-Reply indicates that xTR1 should now be used. The map-cache on xTR3 is updated and air-to-ground traffic now goes through xTR1 to use the SATCOM radio link to the aircraft.

The disadvantage of RLOC-probing is that fast detection becomes more difficult when the number of EID prefixes is large.

6.2. Use of Solicit-Map-Request

Solicit-Map-Request is used as described in Section 5:

1. xTR3 is still sending traffic to xTR2 since its map-cache has not been updated yet.

2. Upon detecting that the link is down, and receiving data plane traffic from the ground network, xTR2 sends an SMR to xTR3 that sends a Map-Request to update its map-cache. The corresponding Map-Reply indicates that xTR1 should now be used.

The disadvantage of this approach is that the traffic is delayed pending control-plane resolution. This method also depends on data traffic being continuous, in many cases data traffic may be sporadic, leading to very slow convergence.

6.3. Use of LISP pub-sub

As specified in [I-D.rodrigueznatal-lisp-pubsub], ITRs can subscribe to changes in the LISP mapping system. So if all ITRs subscribe to the EID prefixes for which they have traffic, the ITRs will be notified when there is mapping change.

In the example where the LDACS radio link fails, when xTR2 unregisters the EID prefixes with the MS/MR, xTR3 would be notified via LISP pub-sub (assuming xTR3 has a map-cache entry for these EID prefixes).

This mechanism provides the fastest convergence at the cost of more state in the LISP mapping system.

7. Security Considerations

For LISP control-plane message security, please refer to [I-D.ietf-lisp-sec]. This addresses the control-plane threats that target EID-to-RLOC mappings, including manipulations of Map-Request and Map-Reply messages, and malicious ETR EID prefix overclaiming.

8. IANA Considerations

No IANA considerations.

9. Acknowledgements

The authors would like to thank Dino Farinacci for his review of the document.

10. References

10.1. Normative References

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

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

10.2. Informative References

[GBL] Frequentis, "Ground Based LISP for Multilink Operation, https://www.icao.int/safety/acp/ACPWGF/CP WG-I 19/WP06 Ground_Based_LISP 2016-01-14.pdf", January 2016.

[I-D.ermagan-lisp-nat-traversal]
Ermagan, V., Farinacci, D., Lewis, D., Skriver, J., Maino, F., and C. White, "NAT traversal for LISP", draft-ermagan-lisp-nat-traversal-13 (work in progress), September 2017.

[I-D.ietf-lisp-eid-mobility]
Portoles-Comeras, M., Ashtaputre, V., Moreno, V., Maino, F., and D. Farinacci, "LISP L2/L3 EID Mobility Using a Unified Control Plane", draft-ietf-lisp-eid-mobility-01 (work in progress), November 2017.

[I-D.ietf-lisp-rfc6833bis]
Fuller, V., Farinacci, D., and A. Cabellos-Aparicio, "Locator/ID Separation Protocol (LISP) Control-Plane", draft-ietf-lisp-rfc6833bis-06 (work in progress), October 2017.

[I-D.ietf-lisp-sec]
Maino, F., Ermagan, V., Cabellos-Aparicio, A., and D. Saucez, "LISP-Security (LISP-SEC)", draft-ietf-lisp-sec-14 (work in progress), October 2017.

[I-D.rodrigueznatal-lisp-pubsub]
Rodriguez-Natal, A., Ermagan, V., Leong, J., Maino, F., Cabellos-Aparicio, A., Barkai, S., Farinacci, D., Boucadair, M., Jacquenet, C., and s. stefano.secci@lip6.fr, "Publish/Subscribe Functionality for LISP", draft-rodrigueznatal-lisp-pubsub-01 (work in progress), October 2017.

Authors' Addresses

Bernhard Haindl
Frequentis

Email: bernhard.haindl@frequentis.com

Manfred Lindner
Frequentis

Email: manfred.lindner@frequentis.com

Reshad Rahman
Cisco Systems

Email: rrahman@cisco.com

Marc Portoles Comeras
Cisco Systems

Email: mportole@cisco.com

Victor Moreno
Cisco Systems

Email: vmoreno@cisco.com

Fabio Maino
Cisco Systems

Email: fmaino@cisco.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: September 6, 2018

D. Lewis
Cisco
J. Lemon
Broadcom
P. Agarwal
Innovium
L. Kreeger

P. Quinn
M. Smith
N. Yadav
F. Maino, Ed.
Cisco
March 05, 2018

LISP Generic Protocol Extension
draft-ietf-lisp-gpe-01

Abstract

This draft describes extending the Locator/ID Separation Protocol (LISP), via changes to the LISP header, to support multi-protocol encapsulation.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

LISP, as defined in [RFC6830] and extended in [I-D.ietf-lisp-rfc6830bis], defines an encapsulation format that carries IPv4 or IPv6 (henceforth referred to as IP) packets in a LISP header and outer UDP/IP transport.

The LISP header does not specify the protocol being encapsulated and therefore is currently limited to encapsulating only IP packet payloads. Other protocols, most notably VXLAN [RFC7348] (which defines a similar header format to LISP), are used to encapsulate L2 protocols such as Ethernet.

This document defines an extension for the LISP header, as defined in [I-D.ietf-lisp-rfc6830bis], to indicate the inner protocol, enabling the encapsulation of Ethernet, IP or any other desired protocol all the while ensuring compatibility with existing LISP deployments.

A flag in the LISP header, called the P-bit, is used to signal the presence of the 8-bit Next Protocol field. The Next Protocol field, when present, uses 8 bits of the field allocated to the echo-noncing and map-versioning features. The two features are still available, albeit with a reduced length of Nonce and Map-Version.

1.1. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1.2. Definition of Terms

This document uses terms already defined in [I-D.ietf-lisp-rfc6830bis].

2. LISP Header Without Protocol Extensions

As described in the introduction, the LISP header has no protocol identifier that indicates the type of payload being carried. Because of this, LISP is limited to carry IP payloads.

The LISP header [I-D.ietf-lisp-rfc6830bis] contains a series of flags (some defined, some reserved), a Nonce/Map-version field and an instance ID/Locator-status-bit field. The flags provide flexibility to define how the various fields are encoded. Notably, Flag bit 5 is the last reserved bit in the LISP header.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|N|L|E|V|I|R|K|K|            Nonce/Map-Version                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Instance ID/Locator-Status-Bits               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

LISP Header

3. Generic Protocol Extension for LISP (LISP-GPE)

This document defines the following changes to the LISP header in order to support multi-protocol encapsulation:

P Bit: Flag bit 5 is defined as the Next Protocol bit. The P bit MUST be set to 1 to indicate the presence of the 8 bit next protocol field.

P = 0 indicates that the payload MUST conform to LISP as defined in [I-D.ietf-lisp-rfc6830bis]. Flag bit 5 was chosen as the P bit because this flag bit is currently unallocated.

Next Protocol: The lower 8 bits of the first 32-bit word are used to carry a Next Protocol. This Next Protocol field contains the protocol of the encapsulated payload packet.

LISP uses the lower 24 bits of the first word for either a nonce, an echo-nonce, or to support map-versioning [RFC6834]. These are all optional capabilities that are indicated in the LISP header by setting the N, E, and the V bit respectively.

When the P-bit and the N-bit are set to 1, the Nonce field is the middle 16 bits.

When the P-bit and the V-bit are set to 1, the Version field is the middle 16 bits.

When the P-bit is set to 1 and the N-bit and the V-bit are both 0, the middle 16-bits are set to 0.

This draft defines the following Next Protocol values:

0x1 : IPv4

0x2 : IPv6

0x3 : Ethernet

0x4 : Network Service Header [RFC8300]

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|N|L|E|V|I|P|K|K|       Nonce/Map-Version      | Next Protocol |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Instance ID/Locator-Status-Bits               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

LISP-GPE Header

4. Backward Compatibility

LISP-GPE uses the same UDP destination port (4341) allocated to LISP.

A LISP-GPE router MUST not encapsulate non-IP packets to a LISP router. A method for determining the capabilities of a LISP router (GPE or "legacy") is out of the scope of this draft.

When encapsulating IP packets to a LISP "legacy" router the P bit MUST be set to 0.

4.1. Type of Service

When a LISP-GPE router performs Ethernet encapsulation, the inner 802.1Q [IEEE8021Q] priority code point (PCP) field MAY be mapped from the encapsulated frame to the Type of Service field in the outer IPv4 header, or in the case of IPv6 the 'Traffic Class' field.

4.2. VLAN Identifier (VID)

When a LISP-GPE router performs Ethernet encapsulation, the inner header 802.1Q [IEEE8021Q] VLAN Identifier (VID) MAY be mapped to, or used to determine the LISP Instance ID field.

5. IANA Considerations

IANA is requested to set up a registry of LISP-GPE "Next Protocol". These are 8-bit values. Next Protocol values in the table below are defined in this draft. New values are assigned via Standards Action [RFC5226].

Next Protocol	Description	Reference
0	Reserved	This Document
1	IPv4	This Document
2	IPv6	This Document
3	Ethernet	This Document
4	NSH	This Document
5..255	Unassigned	

6. Security Considerations

LISP-GPE security considerations are similar to the LISP security considerations documented at length in [I-D.ietf-lisp-rfc6830bis]. With LISP-GPE, issues such as dataplane spoofing, flooding, and traffic redirection may depend on the particular protocol payload encapsulated.

7. Acknowledgements

A special thank you goes to Dino Farinacci for his guidance and detailed review.

8. References

8.1. Normative References

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

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <https://www.rfc-editor.org/info/rfc5226>.

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

[RFC6834] Iannone, L., Saucez, D., and O. Bonaventure, "Locator/ID Separation Protocol (LISP) Map-Versioning", RFC 6834, DOI 10.17487/RFC6834, January 2013, <https://www.rfc-editor.org/info/rfc6834>.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <https://www.rfc-editor.org/info/rfc7348>.

[RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <https://www.rfc-editor.org/info/rfc8300>.

8.2. Informative References

[I-D.ietf-lisp-rfc6830bis]
Farinacci, D., Fuller, V., Meyer, D., Lewis, D., and A. Cabellos-Aparicio, "The Locator/ID Separation Protocol (LISP)", draft-ietf-lisp-rfc6830bis-10 (work in progress), March 2018.

Authors' Addresses

Darrel Lewis
Cisco Systems

Email: darlewis@cisco.com

John Lemon
Broadcom
3151 Zanker Road
San Jose, CA 95134
USA

Email: john.lemon@broadcom.com

Puneet Agarwal
Innovium
USA

Email: puneet@acm.org

Larry Kreeger
USA

Email: lkreeger@gmail.com

Paul Quinn
Cisco Systems

Email: paulq@cisco.com

Michael Smith
Cisco Systems

Email: michsmit@cisco.com

Navindra Yadav
Cisco Systems

Email: nyadav@cisco.com

Fabio Maino (editor)
Cisco Systems
San Jose, CA 95134
USA

Email: fmaino@cisco.com

LISP Working Group A. Rodriguez-Natal
Internet-Draft V. Ermagan
Intended status: Experimental A. Smirnov
Expires: August 20, 2018 V. Ashtaputre
Cisco Systems
D. Farinacci
lispers.net
2 16, 2018

Vendor Specific LCAF
draft-ietf-lisp-vendor-lcaf-01

Abstract

This document describes a new LCAF for LISP, the Vendor Specific LCAF. This LCAF enables organizations to have internal encodings for LCAF addresses.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The LISP Canonical Address Format [RFC8060] defines the format and encoding for different address types that can be used on LISP [RFC6830] deployments. However, certain deployments require specific format encodings that may not be applicable outside of the use-case for which they are defined. The Vendor Specific LCAF allows organizations to create LCAF addresses to be used only internally on particular LISP deployments.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]

3. Vendor Specific LCAF

The Vendor Specific LCAF relies on using the IEEE Organizationally Unique Identifier (OUI) [IEEE.802_2001] to prevent collisions across vendors or organizations using the LCAF. The format of the Vendor Specific LCAF is provided below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           AFI = 16387         |     Rsvd1     |     Flags     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type = 255  |     Rsvd2     |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Rsvd3    |    Organizationally Unique Identifier (OUI)   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Internal format...                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Vendor Specific LCAF

The Vendor Specific LCAF has the following fields.

Rsvd3: This 8-bit field is reserved for future use. It MUST be set to 0 on transmit and MUST be ignored on receipt.

Organizationally Unique Identifier (OUI): This is a 24-bit field that carries the IEEE OUI [IEEE.802_2001] of the organization.

Internal format: This is a variable length field that is left undefined on purpose. Each vendor or organization can define its own internal format(s) to use with the Vendor Specific LCAF.

The definition for the rest of the fields can be found in [RFC8060].

The Vendor Specific LCAF type SHOULD not be used in deployments where different organizations interoperate. If a LISP device receives a LISP message containing a Vendor Specific LCAF with an OUI that it does not understand, it SHOULD drop the message and a log action MUST be taken.

4. Security Considerations

This document enables organizations to define new LCAFs for their internal use. It is the responsibility of these organizations to properly assess the security implications of the formats they define.

5. Acknowledgments

The authors would like to thank Joel Halpern for his suggestions and comments regarding this document.

6. IANA Considerations

Following the guidelines of [RFC5226], this document requests IANA to update the "LISP Canonical Address Format (LCAF) Types" Registry defined in [RFC8060] to allocate the following assignment:

Value #	LISP LCAF Type Name	Reference
255	Vendor Specific	Section 3

Table 1: Vendor Specific LCAF assignment

7. Normative References

[IEEE.802_2001]
IEEE, "IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture", IEEE 802-2001, DOI 10.1109/ieeestd.2002.93395, July 2002, <http://ieeexplore.ieee.org/servlet/opac?punumber=7732>.

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

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <https://www.rfc-editor.org/info/rfc5226>.

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

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

Authors' Addresses

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

Email: natal@cisco.com

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

Email: vermagan@cisco.com

Anton Smirnov
Cisco Systems
170 Tasman Drive
San Jose, CA
USA

Email: asmirnov@cisco.com

Vrushali Ashtaputre
Cisco Systems
170 Tasman Drive
San Jose, CA
USA

Email: vrushali@cisco.com

Dino Farinacci
lispers.net
San Jose, CA
USA

Email: farinacci@gmail.com

LISP Working Group V. Ermagan
Internet-Draft A. Rodriguez-Natal
Intended status: Experimental F. Coras
Expires: September 6, 2018 C. Moberg
R. Rahman
Cisco Systems
A. Cabellos-Aparicio
Technical University of Catalonia
F. Maino
Cisco Systems
March 5, 2018

LISP YANG Model
draft-ietf-lisp-yang-07

Abstract

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

The YANG modules in this document conform to the Network Management Datastore Architecture (NMDA).

Status of This Memo

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 locator-sets
       |  +--rw locator-set* [locator-set-name]
       |     +--rw locator-set-name    string
       |     +--rw (locator-type)?
       |        +--:(local-interface)
       |        |  +--rw interface* [interface-ref]
       |        |     +--rw interface-ref         if:interface-ref
       |        |     +--rw priority?             uint8
       |        |     +--rw weight?               uint8
       |        |     +--rw multicast-priority?   uint8
       |        |     +--rw multicast-weight?     uint8
       |        +--:(general-locator)
       |           +--rw locator* [id]
       |              +--rw id                    string
       |              +--rw locator-address
       |              |  +--rw address-type             lisp-address-family-ref
       |              |  +--rw virtual-network-id?      instance-id-type
       |              |  +--rw (address)?
       |              |     +--:(no-address)
       |              |     |  +--rw no-address?              empty
       |              |     +--:(ipv4)
       |              |     |  +--rw ipv4?                    inet:ipv4-address
       |              |     +--:(ipv4-prefix)
       |              |     |  +--rw ipv4-prefix?             inet:ipv4-prefix
       |              |     +--:(ipv6)
       |              |     |  +--rw ipv6?                    inet:ipv6-address
       |              |     +--:(ipv6-prefix)
       |              |     |  +--rw ipv6-prefix?             inet:ipv6-prefix
       |              |     +--:(mac)
       |              |     |  +--rw mac?                     yang:mac-address
       |              |     +--:(distinguished-name)
       |              |     |  +--rw distinguished-name?      distinguished-name
-type
       |              |     +--:(as-number)
       |              |     |  +--rw as-number?               inet:as-number
       |              |     +--:(null-address)
       |              |     |  +--rw null-address
       |              |     |     +--rw address?   empty
       |              |     +--:(afi-list)
       |              |     |  +--rw afi-list
       |              |     |     +--rw address-list*   simple-address
       |              |     +--:(instance-id)
       |              |     |  +--rw instance-id
       |              |     |     +--rw iid?           instance-id-type
       |              |     |     +--rw mask-length?   uint8
```

```
       |              |     |     +--rw address?       simple-address
       |              |     +--:(as-number-lcaf)
       |              |     |  +--rw as-number-lcaf
       |              |     |     +--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* [hop-id]
       |              |     |        +--rw hop-id      string
       |              |     |        +--rw address?    simple-address
       |              |     |        +--rw lrs-bits?   bits
       |              |     +--:(source-dest-key)
       |              |     |  +--rw source-dest-key
       |              |     |     +--rw source?   simple-address
       |              |     |     +--rw dest?     simple-address
       |              |     +--:(key-value-address)
       |              |     |  +--rw key-value-address
       |              |     |     +--rw key?     simple-address
       |              |     |     +--rw value?   simple-address
```

```
       |              |     +--:(service-path)
       |              |        +--rw service-path
       |              |           +--rw service-path-id?   service-path-id-type
       |              |           +--rw service-index?     uint8
       |              +--rw priority?            uint8
       |              +--rw weight?              uint8
       |              +--rw multicast-priority?  uint8
       |              +--rw multicast-weight?    uint8
       +--rw lisp-router-instances
          +--rw lisp-router-instance* [lisp-router-instance-id]
             +--rw lisp-router-instance-id    int32
             +--rw lisp-role* [lisp-role-type]
             |  +--rw lisp-role-type    lisp-role-ref
             +--rw lisp-router-id
                +--rw site-id?   uint64
                +--rw xtr-id?    lisp:xtr-id-type
```

2.2. Module Definition

```
   <CODE BEGINS> file "ietf-lisp@2018-03-05.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;
     }
     import ietf-yang-types {
       prefix yang;
     }
     organization
       "IETF LISP (Locator/ID Separation Protocol) Working Group";
     contact
       "WG Web:   <http://tools.ietf.org/wg/lisp/>
        WG List:  <mailto:lisp@ietf.org>

        Editor:   Vina Ermagan
                  <mailto:vermagan@cisco.com>

        Editor:   Alberto Rodriguez-Natal
                  <mailto:natal@cisco.com>

        Editor:   Reshad Rahman
                  <mailto:rrahman@cisco.com>";
     description
       "This YANG module defines the generic parameters for LISP.
```

```
        The module can be extended by vendors to define vendor-specific
        LISP parameters and policies.

        Copyright (c) 2018 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 XXXX; see
        the RFC itself for full legal notices.
      ";
    // RFC Ed.: replace XXXX with actual RFC number and remove
    // this note
    reference "RFC XXXX";

    revision 2018-03-05 {
      description
        "Initial revision.";
      reference
        "https://tools.ietf.org/html/rfc6830";
    }
    identity lisp-role {
      description
        "LISP router role.";
    }
    identity itr {
      base lisp-role;
      description
        "LISP ITR.";
    }
    identity pitr {
      base lisp-role;
      description
        "LISP PITR.";
    }
    identity etr {
      base lisp-role;
      description
        "LISP ETR.";
    }
    identity petr {
      base lisp-role;
      description
```

```
           "LISP PETR.";
       }
       identity mapping-system {
         description
           "Mapping System interface";
       }
       identity single-node-mapping-system {
         base mapping-system;
         description
           "logically singular Map Server";
       }
       typedef mapping-system-ref {
         type identityref {
           base mapping-system;
         }
         description
           "Mapping System reference";
       }

       typedef lisp-role-ref {
         type identityref {
           base lisp-role;
         }
         description
           "LISP role 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 mechanisms supported by
          LISP.";
       reference
         "https://tools.ietf.org/html/rfc6830#section-6.1.6";
     }
     typedef xtr-id-type {
       type binary {
         length "16";
       }
       description
       "128 bit xTR identifier.";
     }

     grouping locator-properties {
       description
         "Properties of a RLOC";
       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 locators-grouping {
        description
          "Group that defines a list of LISP locators.";
          list locator {
            key "id";
            description
              "List of routing locators";
            leaf id {
              type string {
                length "1..64";
              }
              description
                "Locator id";
            }
            container locator-address {
              uses lcaf:lisp-address;
              description
                "The locator address provided in LISP canonincal
                 address format.";
            }
            uses locator-properties;
          }

      }

      grouping local-locators-grouping {
        description
```

```
        "Group that defines a list of LISP locators.";
      list interface {
        key "interface-ref";
        description
          "The address type of the locator";
        leaf interface-ref {
          type if:interface-ref;
          description
            "The name of the interface supporting the locator.";
        }
        uses locator-properties;
      }
    }

    grouping mapping {
      description
        "Group that defines a LISP mapping.";
      container eid {
        uses lcaf:lisp-address;
        description
          "End-host Identifier (EID) to be mapped to a list of
           locators";
      }
      leaf time-to-live {
        type uint32;
        units minutes;
        description
          "Mapping validity period in minutes.";
      }
      leaf creation-time {
        type yang:date-and-time;
        description
          "Time when the mapping was created.";
      }
      leaf authoritative {
        type bits {
          bit A {
            description
              "Authoritative bit.";
          }
        }
        description
          "Bit that indicates if mapping comes from an
           authoritative source.";
      }
      leaf static {
        type boolean;
        default "false";
```

```
          description
            "This leaf should be true if the mapping is static.";
        }
        choice locator-list {
          description
            "list of locartors are either negative, or positive.";
          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-grouping;
              description
                "List of locators for a positive mapping.";
            }
          }
        }
      }

      grouping mappings {
        description
          "Group that defines a list of LISP mappings.";
        list virtual-network {
          key "vni";
          description
            "Virtual network to which the mappings belong.";
          leaf vni {
            type lcaf:instance-id-type;
            description
              "Virtual network identifier.";
          }
          container mappings {
            description
              "Mappings within the virtual network.";
            list mapping {
              key "id";
              description
                "List of EID to RLOCs mappings.";
              leaf id {
                type eid-id;
                description
                  "Id that uniquely identifies a mapping.";
              }
              uses mapping;
```

```
          }
        }
      }
    }

    container lisp {
      description
        "Parameters for the LISP subsystem.";

      container locator-sets {
        description
          "Container that defines a named locator set which can be
          referenced elsewhere.";
        list locator-set {
          key "locator-set-name";
          description
            "Multiple locator sets can be defined.";
          leaf locator-set-name {
            type string {
              length "1..64";
            }
            description
              "Locator set name";
          }
          choice locator-type {
            description
              "Locator sets can be based on local interfaces, or
               general locators.";
            case local-interface {
              uses local-locators-grouping;
              description
                "List of locators in this set based on local
                interfaces.";
            }
            case general-locator {
              uses locators-grouping;
              description
                "List of locators in this set based on lisp-address.";
            }
          }
        }
      }

      container lisp-router-instances {
        description
          "Different LISP routers instantiated in the device";
        list lisp-router-instance {
          key "lisp-router-instance-id";
```

```
          description
            "Each entry contains parameters for a LISP router.";
          leaf lisp-router-instance-id {
            type int32;
            description
              "Arbitrary lisp-router id.";
          }
          list lisp-role {
            key lisp-role-type;
            description
              "List of lisp device roles such as MS, MR, ITR,
              PITR, ETR or PETR.";
            leaf lisp-role-type {
              type lisp-role-ref;
              description
                "The type of LISP device - identity derived from the
                 'lisp-device' base identity.";
            }
          }
          container lisp-router-id {
            when "../lisp-role/lisp-role-type = 'itr' or
                  ../lisp-role/lisp-role-type = 'pitr' or
                  ../lisp-role/lisp-role-type = 'etr' or
                  ../lisp-role/lisp-role-type = 'petr'" {
              description "Only when ITR, PITR, ETR or PETR.";
            }
            description
              "Site-ID and xTR-ID of the device.";
            leaf site-id {
              type uint64;
              description "Site ID";
            }
            leaf xtr-id {
              type lisp:xtr-id-type;
              description "xTR ID";
            }
          }
        }
      }
    }
  }
  <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:lisp-router-instances/lisp:lisp-router-instance:
    +--rw itr!
       +--rw rloc-probing!
       |  +--rw interval?           uint16
       |  +--rw retries?            uint8
       |  +--rw retries-interval?   uint16
       +--rw itr-rlocs?       -> /lisp:lisp/locator-sets/locator-set/locator-set
-name
       +--rw map-resolvers
       |  +--rw map-resolver*   inet:ip-address
       +--rw proxy-etrs
       |  +--rw proxy-etr-address*   inet:ip-address
       +--rw map-cache
          +--rw virtual-network* [vni]
             +--rw vni         lcaf:instance-id-type
             +--rw mappings
                +--rw mapping* [id]
                   +--rw id                  eid-id
                   +--rw eid
                   |  +--rw address-type             lisp-address-family-ref
                   |  +--rw virtual-network-id?      instance-id-type
                   |  +--rw (address)?
                   |     +--:(no-address)
                   |     |  +--rw no-address?              empty
                   |     +--:(ipv4)
                   |     |  +--rw ipv4?                    inet:ipv4-address
                   |     +--:(ipv4-prefix)
                   |     |  +--rw ipv4-prefix?             inet:ipv4-prefix
                   |     +--:(ipv6)
                   |     |  +--rw ipv6?                    inet:ipv6-address
                   |     +--:(ipv6-prefix)
                   |     |  +--rw ipv6-prefix?             inet:ipv6-prefix
                   |     +--:(mac)
                   |     |  +--rw mac?                     yang:mac-address
                   |     +--:(distinguished-name)
                   |     |  +--rw distinguished-name?      distinguished-name-ty
pe
                   |     +--:(as-number)
                   |     |  +--rw as-number?               inet:as-number
                   |     +--:(null-address)
                   |     |  +--rw null-address
                   |     |     +--rw address?   empty
                   |     +--:(afi-list)
                   |     |  +--rw afi-list
                   |     |     +--rw address-list*   simple-address
                   |     +--:(instance-id)
                   |     |  +--rw instance-id
```

```
           |     |     +--rw iid?           instance-id-type
           |     |     +--rw mask-length?   uint8
           |     |     +--rw address?       simple-address
           |     +--:(as-number-lcaf)
           |     |  +--rw as-number-lcaf
           |     |     +--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* [hop-id]
           |     |        +--rw hop-id      string
           |     |        +--rw address?    simple-address
           |     |        +--rw lrs-bits?   bits
           |     +--:(source-dest-key)
           |     |  +--rw source-dest-key
           |     |     +--rw source?   simple-address
           |     |     +--rw dest?     simple-address
           |     +--:(key-value-address)
           |     |  +--rw key-value-address
```

```
                   |     |     +--rw key?     simple-address
                   |     |     +--rw value?   simple-address
                   |     +--:(service-path)
                   |        +--rw service-path
                   |           +--rw service-path-id?   service-path-id-type
                   |           +--rw service-index?     uint8
                   +--rw time-to-live?       uint32
                   +--rw creation-time?      yang:date-and-time
                   +--rw authoritative?      bits
                   +--rw static?             boolean
                   +--rw (locator-list)?
                      +--:(negative-mapping)
                      |  +--rw map-reply-action?   map-reply-action
                      +--:(positive-mapping)
                         +--rw rlocs
                            +--rw locator* [id]
                               +--rw id                     string
                               +--rw locator-address
                               |  +--rw address-type             lisp-address-fa
mily-ref
                               |  +--rw virtual-network-id?      instance-id-typ
e
                               |  +--rw (address)?
                               |     +--:(no-address)
                               |     |  +--rw no-address?              empty
                               |     +--:(ipv4)
                               |     |  +--rw ipv4?                    inet:ipv4
-address
                               |     +--:(ipv4-prefix)
                               |     |  +--rw ipv4-prefix?             inet:ipv4
-prefix
                               |     +--:(ipv6)
                               |     |  +--rw ipv6?                    inet:ipv6
-address
                               |     +--:(ipv6-prefix)
                               |     |  +--rw ipv6-prefix?             inet:ipv6
-prefix
                               |     +--:(mac)
                               |     |  +--rw mac?                     yang:mac-
address
                               |     +--:(distinguished-name)
                               |     |  +--rw distinguished-name?      distingui
shed-name-type
                               |     +--:(as-number)
                               |     |  +--rw as-number?               inet:as-n
umber
                               |     +--:(null-address)
                               |     |  +--rw null-address
                               |     |     +--rw address?   empty
                               |     +--:(afi-list)
                               |     |  +--rw afi-list
                               |     |     +--rw address-list*   simple-address
                               |     +--:(instance-id)
                               |     |  +--rw instance-id
                               |     |     +--rw iid?           instance-id-type
                               |     |     +--rw mask-length?   uint8
                               |     |     +--rw address?       simple-address
```

```
                                |     +--:(as-number-lcaf)
                                |     |  +--rw as-number-lcaf
                                |     |     +--rw as?        inet:as-number
                                |     |     +--rw address?   simple-address
                                |     +--:(application-data)
                                |     |  +--rw application-data
                                |     |     +--rw address?            simple-addr
ess
                                |     |     +--rw protocol?           uint8
                                |     |     +--rw ip-tos?             int32
                                |     |     +--rw local-port-low?     inet:port-n
umber
                                |     |     +--rw local-port-high?    inet:port-n
umber
                                |     |     +--rw remote-port-low?    inet:port-n
umber
                                |     |     +--rw remote-port-high?   inet:port-n
umber
                                |     +--:(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
ress
                                |     +--:(nat-traversal)
                                |     |  +--rw nat-traversal
                                |     |     +--rw ms-udp-port?        uint16
                                |     |     +--rw etr-udp-port?       uint16
                                |     |     +--rw global-etr-rloc?    simple-addr
ess
                                |     |     +--rw ms-rloc?            simple-addr
ess
                                |     |     +--rw private-etr-rloc?   simple-addr
ess
                                |     |     +--rw rtr-rlocs*          simple-addr
ess
                                |     +--:(explicit-locator-path)
                                |     |  +--rw explicit-locator-path
                                |     |     +--rw hop* [hop-id]
                                |     |        +--rw hop-id      string
                                |     |        +--rw address?    simple-address
                                |     |        +--rw lrs-bits?   bits
                                |     +--:(source-dest-key)
                                |     |  +--rw source-dest-key
                                |     |     +--rw source?   simple-address
                                |     |     +--rw dest?     simple-address
                                |     +--:(key-value-address)
                                |     |  +--rw key-value-address
                                |     |     +--rw key?     simple-address
                                |     |     +--rw value?   simple-address
                                |     +--:(service-path)
```

```
                               |        +--rw service-path
                               |           +--rw service-path-id?   service-path
-id-type
                               |           +--rw service-index?     uint8
                               +--rw priority?            uint8
                               +--rw weight?              uint8
                               +--rw multicast-priority?  uint8
                               +--rw multicast-weight?    uint8
```

3.2. Module Definition

```
<CODE BEGINS> file "ietf-lisp-itr@2018-03-05.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-inet-types {
    prefix inet;
  }
  organization
    "IETF LISP (Locator/ID Separation Protocol) Working Group";
  contact
    "WG Web:   <http://tools.ietf.org/wg/lisp/>
     WG List:  <mailto:lisp@ietf.org>

     Editor:   Vina Ermagan
               <mailto:vermagan@cisco.com>

     Editor:   Alberto Rodriguez-Natal
              <mailto:natal@cisco.com>

     Editor:   Reshad Rahman
               <mailto:rrahman@cisco.com>";
  description
    "This YANG module defines the generic parameters for a LISP
     ITR. The module can be extended by vendors to define
     vendor-specific parameters and policies.

     Copyright (c) 2018 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 XXXX; see
       the RFC itself for full legal notices.
      ";
    // RFC Ed.: replace XXXX with actual RFC number and remove
    // this note
    reference "RFC XXXX";

    revision 2018-03-05 {
      description
        "Initial revision.";
      reference
        "https://tools.ietf.org/html/rfc6830";
    }
    augment "/lisp:lisp/lisp:lisp-router-instances/lisp:lisp-router-instance" {
      when "lisp:lisp-role/lisp:lisp-role-type = 'lisp:itr' or
            lisp:lisp-role/lisp:lisp-role-type = 'lisp:pitr'" {
        description
          "Augment is valid when LISP role type is ITR or PITR.";
      }
      description
        "This augments LISP devices list with (P)ITR specific
         parameters.";
      container itr {
        presence "LISP (P)ITR operation enabled";
        description
          "ITR parameters";
        container rloc-probing {
          presence "RLOC probing active";
          description
            "RLOC-probing parameters";
          leaf interval {
            type uint16;
            units "seconds";
            description
              "Interval in seconds for resending the probes";
          }
          leaf retries {
            type uint8;
            description
              "Number of retries for sending the probes";
          }
          leaf retries-interval {
            type uint16;
            units "seconds";
            description
              "Interval in seconds between retries when sending probes.
               The action taken if all retries fail to receive is
               impementation specific.";
```

```
          }
        }
        leaf itr-rlocs {
          type leafref {
            path "/lisp:lisp/lisp:locator-sets/lisp:locator-set/"
            + "lisp:locator-set-name";
          }
          description
            "Reference to a locator set that the (P)ITR includes in
             Map-Requests";
        }
        container map-resolvers {
          description
            "Map-Resolvers that the (P)ITR uses.";
          leaf-list map-resolver {
            type inet:ip-address;
            min-elements 1;
            description
              "Each Map-Resolver within the list of Map-Resolvers.";
          }
        }
        container proxy-etrs {
          when "../../lisp:lisp-role/lisp:lisp-role-type = 'lisp:itr'" {
            description
              "Container exists only when LISP role type is ITR";
          }
          description
            "Proxy ETRs that the ITR uses.";
          leaf-list proxy-etr-address{
            type inet:ip-address;
            description
              "Proxy ETR RLOC address.";
          }
        }
        container map-cache{
          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:lisp-router-instances/lisp:lisp-router-instance:
    +--rw etr!
       +--rw map-servers
       |  +--rw map-server* [ms-address]
       |     +--rw ms-address       inet:ip-address
       |     +--rw auth-key?        string
       |     +--rw auth-key-type?   lisp:auth-key-type
       +--rw local-eids
          +--rw virtual-network* [vni]
             +--rw vni     lcaf:instance-id-type
             +--rw eids
                +--rw local-eid* [id]
                   +--rw id                      lisp:eid-id
                   +--rw eid-address
                   |  +--rw address-type            lisp-address-family-ref
                   |  +--rw virtual-network-id?     instance-id-type
                   |  +--rw (address)?
                   |     +--:(no-address)
                   |     |  +--rw no-address?              empty
                   |     +--:(ipv4)
                   |     |  +--rw ipv4?                    inet:ipv4-address
                   |     +--:(ipv4-prefix)
                   |     |  +--rw ipv4-prefix?             inet:ipv4-prefix
                   |     +--:(ipv6)
                   |     |  +--rw ipv6?                    inet:ipv6-address
                   |     +--:(ipv6-prefix)
                   |     |  +--rw ipv6-prefix?             inet:ipv6-prefix
                   |     +--:(mac)
                   |     |  +--rw mac?                     yang:mac-address
                   |     +--:(distinguished-name)
                   |     |  +--rw distinguished-name?      distinguished-name-ty
pe
                   |     +--:(as-number)
                   |     |  +--rw as-number?               inet:as-number
                   |     +--:(null-address)
                   |     |  +--rw null-address
                   |     |     +--rw address?   empty
                   |     +--:(afi-list)
                   |     |  +--rw afi-list
                   |     |     +--rw address-list*   simple-address
                   |     +--:(instance-id)
```

```
   |     |  +--rw instance-id
   |     |     +--rw iid?           instance-id-type
   |     |     +--rw mask-length?   uint8
   |     |     +--rw address?       simple-address
   |     +--:(as-number-lcaf)
   |     |  +--rw as-number-lcaf
   |     |     +--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* [hop-id]
   |     |        +--rw hop-id      string
   |     |        +--rw address?    simple-address
   |     |        +--rw lrs-bits?   bits
   |     +--:(source-dest-key)
   |     |  +--rw source-dest-key
   |     |     +--rw source?   simple-address
   |     |     +--rw dest?     simple-address
   |     +--:(key-value-address)
```

```
                   |     |  +--rw key-value-address
                   |     |     +--rw key?     simple-address
                   |     |     +--rw value?   simple-address
                   |     +--:(service-path)
                   |        +--rw service-path
                   |           +--rw service-path-id?   service-path-id-type
                   |           +--rw service-index?     uint8
                   +--rw rlocs?                   -> /lisp:lisp/locator-sets/loc
ator-set/locator-set-name
                   +--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@2018-03-05.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
    "WG Web:   <http://tools.ietf.org/wg/lisp/>
     WG List:  <mailto:lisp@ietf.org>

     Editor:   Vina Ermagan
               <mailto:vermagan@cisco.com>

     Editor:   Alberto Rodriguez-Natal
               <mailto:natal@cisco.com>

     Editor:   Reshad Rahman
               <mailto:rrahman@cisco.com>";
  description
    "This YANG module defines the generic parameters for a LISP
     ETR. The module can be extended by vendors to define
     vendor-specific parameters and policies.

     Copyright (c) 2018 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 XXXX; see
     the RFC itself for full legal notices.
    ";
  // RFC Ed.: replace XXXX with actual RFC number and remove
  // this note
  reference "RFC XXXX";

  revision 2018-03-05 {
    description
      "Initial revision.";
    reference
      "https://tools.ietf.org/html/rfc6830";
  }
  augment "/lisp:lisp/lisp:lisp-router-instances/lisp:lisp-router-instance" {
    when "lisp:lisp-role/lisp:lisp-role-type = 'lisp:etr' or
          lisp:lisp-role/lisp:lisp-role-type = 'lisp:petr'" {
      description
        "Augment is valid when LISP device type is (P)ETR.";
    }
    description
      "This augments LISP devices list with (P)ETR specific
       parameters.";
    container etr {
      presence "LISP (P)ETR operation enabled";
      description
        "(P)ETR parameters.";

      container map-servers {
        when "../../lisp:lisp-role/lisp:lisp-role-type = 'lisp:etr'" {
          description
            "Container exists only when LISP device type is ETR.";
        }
        description
          "Map-Servers that the ETR uses.";
        list map-server {
          key "ms-address";
          description
            "Each Map-Server within the list of Map-Servers.";
          leaf ms-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 local-eids {
      when "../../lisp:lisp-role/lisp:lisp-role-type = 'lisp:etr'" {
        description
          "Container exists only when LISP device type is ETR.";
      }
      description
        "Virtual networks served by the ETR.";
      list virtual-network {
        key "vni";
        description
          "Virtual network for local-EIDs.";
        leaf vni {
          type lcaf:instance-id-type;
          description
            "Virtual network identifier.";
        }
        container 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.";
              }
              leaf rlocs {
                type leafref {
                  path "/lisp:lisp/lisp:locator-sets/lisp:locator-set/"
                  + "lisp:locator-set-name";
                }
                description
                  "Locator set mapped to this local EID.";
              }
              leaf record-ttl {
                type uint32;
                units minutes;
                description
                  "Validity period of the EID to RLOCs mapping provided
                  in Map-Replies.";
              }
              leaf want-map-notify {
                type boolean;
                default "true";
                description
                  "Flag which if set in a Map-Register requests that a
                  Map-Notify be sent in response.";
              }
              leaf proxy-reply {
                type boolean;
                default "false";
                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:lisp-router-instances/lisp:lisp-router-instance:
    +--rw map-server!
       +--rw sites
       |  +--rw site* [site-id]
       |     +--rw site-id     uint64
       |     +--rw auth-key
       |        +--rw auth-key-value?   string
       |        +--rw auth-key-type*    lisp:auth-key-type
       +--rw virtual-network-ids
       |  +--rw virtual-network-identifier* [vni]
       |     +--rw vni          lcaf:instance-id-type
       |     +--rw mappings
       |     |  +--rw mapping* [eid-id]
       |     |     +--rw eid-id                        lisp:eid-id
       |     |     +--rw eid-address
       |     |     |  +--rw address-type            lisp-address-family-ref
       |     |     |  +--rw virtual-network-id?      instance-id-type
       |     |     |  +--rw (address)?
       |     |     |     +--:(no-address)
       |     |     |     |  +--rw no-address?             empty
       |     |     |     +--:(ipv4)
       |     |     |     |  +--rw ipv4?                   inet:ipv4-address
       |     |     |     +--:(ipv4-prefix)
       |     |     |     |  +--rw ipv4-prefix?            inet:ipv4-prefix
       |     |     |     +--:(ipv6)
       |     |     |     |  +--rw ipv6?                   inet:ipv6-address
       |     |     |     +--:(ipv6-prefix)
       |     |     |     |  +--rw ipv6-prefix?            inet:ipv6-prefix
       |     |     |     +--:(mac)
       |     |     |     |  +--rw mac?                    yang:mac-address
       |     |     |     +--:(distinguished-name)
       |     |     |     |  +--rw distinguished-name?      distinguished-name-ty
pe
       |     |     |     +--:(as-number)
       |     |     |     |  +--rw as-number?              inet:as-number
       |     |     |     +--:(null-address)
       |     |     |     |  +--rw null-address
       |     |     |     |     +--rw address?   empty
       |     |     |     +--:(afi-list)
       |     |     |     |  +--rw afi-list
```

```
        |     |     |     |     +--rw address-list*   simple-address
        |     |     |     +--:(instance-id)
        |     |     |     |  +--rw instance-id
        |     |     |     |     +--rw iid?           instance-id-type
        |     |     |     |     +--rw mask-length?   uint8
        |     |     |     |     +--rw address?       simple-address
        |     |     |     +--:(as-number-lcaf)
        |     |     |     |  +--rw as-number-lcaf
        |     |     |     |     +--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* [hop-id]
        |     |     |     |        +--rw hop-id      string
        |     |     |     |        +--rw address?    simple-address
        |     |     |     |        +--rw lrs-bits?   bits
        |     |     |     +--:(source-dest-key)
        |     |     |     |  +--rw source-dest-key
        |     |     |     |     +--rw source?   simple-address
```

```
       |     |     |     |     +--rw dest?     simple-address
       |     |     |     +--:(key-value-address)
       |     |     |     |  +--rw key-value-address
       |     |     |     |     +--rw key?     simple-address
       |     |     |     |     +--rw value?   simple-address
       |     |     |     +--:(service-path)
       |     |     |        +--rw service-path
       |     |     |           +--rw service-path-id?   service-path-id-type
       |     |     |           +--rw service-index?     uint8
       |     |     +--rw site-id*                        uint64
       |     |     +--rw more-specifics-accepted?      boolean
       |     |     +--rw mapping-expiration-timeout?   int16
       |     |     +--rw mapping-records
       |     |        +--rw mapping-record* [xtr-id]
       |     |           +--rw xtr-id              lisp:xtr-id-type
       |     |           +--rw site-id?            uint64
       |     |           +--rw eid
       |     |           |  +--rw address-type             lisp-address-family-r
ef
       |     |           |  +--rw virtual-network-id?      instance-id-type
       |     |           |  +--rw (address)?
       |     |           |     +--:(no-address)
       |     |           |     |  +--rw no-address?              empty
       |     |           |     +--:(ipv4)
       |     |           |     |  +--rw ipv4?                    inet:ipv4-addre
ss
       |     |           |     +--:(ipv4-prefix)
       |     |           |     |  +--rw ipv4-prefix?             inet:ipv4-prefi
x
       |     |           |     +--:(ipv6)
       |     |           |     |  +--rw ipv6?                    inet:ipv6-addre
ss
       |     |           |     +--:(ipv6-prefix)
       |     |           |     |  +--rw ipv6-prefix?             inet:ipv6-prefi
x
       |     |           |     +--:(mac)
       |     |           |     |  +--rw mac?                     yang:mac-addres
s
       |     |           |     +--:(distinguished-name)
       |     |           |     |  +--rw distinguished-name?      distinguished-n
ame-type
       |     |           |     +--:(as-number)
       |     |           |     |  +--rw as-number?               inet:as-number
       |     |           |     +--:(null-address)
       |     |           |     |  +--rw null-address
       |     |           |     |     +--rw address?   empty
       |     |           |     +--:(afi-list)
       |     |           |     |  +--rw afi-list
       |     |           |     |     +--rw address-list*   simple-address
       |     |           |     +--:(instance-id)
       |     |           |     |  +--rw instance-id
       |     |           |     |     +--rw iid?           instance-id-type
       |     |           |     |     +--rw mask-length?   uint8
       |     |           |     |     +--rw address?       simple-address
       |     |           |     +--:(as-number-lcaf)
```

```
       |     |           |     |  +--rw as-number-lcaf
       |     |           |     |     +--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* [hop-id]
       |     |           |     |        +--rw hop-id      string
       |     |           |     |        +--rw address?    simple-address
       |     |           |     |        +--rw lrs-bits?   bits
       |     |           |     +--:(source-dest-key)
       |     |           |     |  +--rw source-dest-key
       |     |           |     |     +--rw source?   simple-address
       |     |           |     |     +--rw dest?     simple-address
       |     |           |     +--:(key-value-address)
       |     |           |     |  +--rw key-value-address
       |     |           |     |     +--rw key?     simple-address
       |     |           |     |     +--rw value?   simple-address
       |     |           |     +--:(service-path)
       |     |           |        +--rw service-path
```

```
       |     |           |           +--rw service-path-id?   service-path-id-ty
pe
       |     |           |           +--rw service-index?     uint8
       |     |           +--rw time-to-live?       uint32
       |     |           +--rw creation-time?      yang:date-and-time
       |     |           +--rw authoritative?      bits
       |     |           +--rw static?             boolean
       |     |           +--rw (locator-list)?
       |     |              +--:(negative-mapping)
       |     |              |  +--rw map-reply-action?   map-reply-action
       |     |              +--:(positive-mapping)
       |     |                 +--rw rlocs
       |     |                    +--rw locator* [id]
       |     |                       +--rw id                    string
       |     |                       +--rw locator-address
       |     |                       |  +--rw address-type             lisp-addr
ess-family-ref
       |     |                       |  +--rw virtual-network-id?      instance-
id-type
       |     |                       |  +--rw (address)?
       |     |                       |     +--:(no-address)
       |     |                       |     |  +--rw no-address?              emp
ty
       |     |                       |     +--:(ipv4)
       |     |                       |     |  +--rw ipv4?                    ine
t:ipv4-address
       |     |                       |     +--:(ipv4-prefix)
       |     |                       |     |  +--rw ipv4-prefix?             ine
t:ipv4-prefix
       |     |                       |     +--:(ipv6)
       |     |                       |     |  +--rw ipv6?                    ine
t:ipv6-address
       |     |                       |     +--:(ipv6-prefix)
       |     |                       |     |  +--rw ipv6-prefix?             ine
t:ipv6-prefix
       |     |                       |     +--:(mac)
       |     |                       |     |  +--rw mac?                     yan
g:mac-address
       |     |                       |     +--:(distinguished-name)
       |     |                       |     |  +--rw distinguished-name?      dis
tinguished-name-type
       |     |                       |     +--:(as-number)
       |     |                       |     |  +--rw as-number?               ine
t:as-number
       |     |                       |     +--:(null-address)
       |     |                       |     |  +--rw null-address
       |     |                       |     |     +--rw address?   empty
       |     |                       |     +--:(afi-list)
       |     |                       |     |  +--rw afi-list
       |     |                       |     |     +--rw address-list*   simple-ad
dress
       |     |                       |     +--:(instance-id)
       |     |                       |     |  +--rw instance-id
       |     |                       |     |     +--rw iid?          instance-i
d-type
       |     |                       |     |     +--rw mask-length?   uint8
       |     |                       |     |     +--rw address?       simple-add
ress
       |     |                       |     +--:(as-number-lcaf)
       |     |                       |     |  +--rw as-number-lcaf
       |     |                       |     |     +--rw as?        inet:as-number
       |     |                       |     |     +--rw address?   simple-address
```

```
       |     |                        |     +--:(application-data)
       |     |                        |     |  +--rw application-data
       |     |                        |     |     +--rw address?            simpl
e-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?    uint
8
       |     |                        |     |     +--rw latitude-minutes?    uint
8
       |     |                        |     |     +--rw latitude-seconds?    uint
8
       |     |                        |     |     +--rw longitude?           bits
       |     |                        |     |     +--rw longitude-degrees?   uint
16
       |     |                        |     |     +--rw longitude-minutes?   uint
8
       |     |                        |     |     +--rw longitude-seconds?   uint
8
       |     |                        |     |     +--rw altitude?            int3
2
       |     |                        |     |     +--rw address?             simp
le-address
       |     |                        |     +--:(nat-traversal)
       |     |                        |     |  +--rw nat-traversal
       |     |                        |     |     +--rw ms-udp-port?        uint1
6
       |     |                        |     |     +--rw etr-udp-port?       uint1
6
       |     |                        |     |     +--rw global-etr-rloc?    simpl
e-address
       |     |                        |     |     +--rw ms-rloc?            simpl
e-address
       |     |                        |     |     +--rw private-etr-rloc?   simpl
e-address
       |     |                        |     |     +--rw rtr-rlocs*          simpl
e-address
       |     |                        |     +--:(explicit-locator-path)
       |     |                        |     |  +--rw explicit-locator-path
       |     |                        |     |     +--rw hop* [hop-id]
       |     |                        |     |        +--rw hop-id      string
       |     |                        |     |        +--rw address?    simple-add
ress
       |     |                        |     |        +--rw lrs-bits?   bits
       |     |                        |     +--:(source-dest-key)
       |     |                        |     |  +--rw source-dest-key
       |     |                        |     |     +--rw source?   simple-address
       |     |                        |     |     +--rw dest?     simple-address
       |     |                        |     +--:(key-value-address)
       |     |                        |     |  +--rw key-value-address
       |     |                        |     |     +--rw key?     simple-address
       |     |                        |     |     +--rw value?   simple-address
```

```
        |     |                        |     +--:(service-path)
        |     |                        |        +--rw service-path
        |     |                        |           +--rw service-path-id?   servic
e-path-id-type
        |     |                        |           +--rw service-index?     uint8
        |     |                        +--rw priority?             uint8
```

```
       |     |                       +--rw weight?               uint8
       |     |                       +--rw multicast-priority?   uint8
       |     |                       +--rw multicast-weight?     uint8
       |     +--ro counters
       |        +--ro map-registers-in?              yang:counter32
       |        +--ro map-registers-in-auth-failed?  yang:counter32
       |        +--ro map-notify-records-out?        yang:counter32
       |        +--ro proxy-reply-records-out?       yang:counter32
       |        +--ro map-requests-forwarded-out?    yang:counter32
       +--rw mapping-system-type?   lisp:mapping-system-ref
       +--ro summary
       |  +--ro number-configured-sites?   uint32
       |  +--ro number-registered-sites?   uint32
       |  +--ro af-datum
       |     +--ro af-data* [address-type]
       |        +--ro address-type              lcaf:lisp-address-family-ref
       |        +--ro number-configured-eids?   uint32
       |        +--ro number-registered-eids?   uint32
       +--ro counters
          +--ro map-registers-in?              yang:counter32
          +--ro map-registers-in-auth-failed?  yang:counter32
          +--ro map-notify-records-out?        yang:counter32
          +--ro proxy-reply-records-out?       yang:counter32
          +--ro map-requests-forwarded-out?    yang:counter32
```

5.2. Module Definition

```
   <CODE BEGINS> file "ietf-lisp-mapserver@2018-03-05.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-yang-types {
       prefix yang;
       revision-date 2013-07-15;
     }

     organization
       "IETF LISP (Locator/ID Separation Protocol) Working Group";
     contact
       "WG Web:   <http://tools.ietf.org/wg/lisp/>
        WG List:  <mailto:lisp@ietf.org>
```

```
        Editor:   Vina Ermagan
                  <mailto:vermagan@cisco.com>

        Editor:   Alberto Rodriguez-Natal
                  <mailto:natal@cisco.com>

        Editor:   Reshad Rahman
                  <mailto:rrahman@cisco.com>";
     description
       "This YANG module defines the generic parameters for a LISP
        Map-Server. The module can be extended by vendors to define
        vendor-specific parameters and policies.

        Copyright (c) 2018 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 XXXX; see
        the RFC itself for full legal notices.
       ";
     // RFC Ed.: replace XXXX with actual RFC number and remove
     // this note
     reference "RFC XXXX";

     revision 2018-03-05 {
       description
         "Initial revision.";
       reference
         "https://tools.ietf.org/html/rfc6833";
     }

     identity ms {
       base lisp:lisp-role;
         description
           "LISP Map-Server.";
     }

     grouping ms-counters {
         description "Group that defines map-server counters.";
         container counters {
           config false;
           description "Container for the counters";
```

```
           leaf map-registers-in {
             type yang:counter64;
             description "Number of incoming Map-Register messages";
           }

           leaf map-registers-in-auth-failed {
             type yang:counter64;
             description
               "Number of incoming Map-Register messages failed
               authentication";
           }

           leaf map-notify-records-out {
             type yang:counter64;
             description
               "Number of outgoing Map-Notify records";
           }

           leaf proxy-reply-records-out {
             type yang:counter64;
             description
               "Number of outgoing proxy Map-Reply records";
           }

           leaf map-requests-forwarded-out {
             type yang:counter64;
             description
               "Number of outgoing Map-Requests forwarded to ETR";
           }
         }
       }

     augment "/lisp:lisp/lisp:lisp-router-instances"
     + "/lisp:lisp-router-instance" {
       when "lisp:lisp-role/lisp:lisp-role-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 {
         presence "LISP Map-Server operation enabled";
         description
           "Map-Server parameters.";
           container sites{
             description
               "Sites to accept registrations from.";
```

```
             list site {
               key site-id;
               description
                 "Site that can send registrations.";
               leaf site-id {
                 type uint64;
                 description "Site ID";
               }
               container auth-key {
                 description
                 "Site authentication key.";
                 leaf auth-key-value {
                   type string;
                   description
                   "Clear text authentication key";
                 }
                 leaf-list auth-key-type {
                   type lisp:auth-key-type;
                   description
                   "Authentication key type.";
                 }
               }
             }
           }
         container virtual-network-ids {
           description
             "Sites for which the Map-Server accepts registrations.";
           list virtual-network-identifier {
             key "vni";
             description
               "Virtual network instances in the Map-Server.";
             leaf vni {
               type lcaf:instance-id-type;
               description
                 "Virtual network identifier.";
             }
             container mappings {
               description
                 "EIDs registered by device.";
               list mapping {
                 key "eid-id";
                 description
                   "List of EIDs registered by device.";
                 leaf eid-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-list site-id {
                  type uint64;
                  description "Site ID";
                }
                leaf more-specifics-accepted {
                    type boolean;
                    default "false";
                    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.";
                }
                container mapping-records {
                  description
                    "Datastore of registered mappings.";
                  list mapping-record{
                    key xtr-id;
                    description
                      "Registered mapping.";
                    leaf xtr-id {
                      type lisp:xtr-id-type;
                      description "xTR ID";
                    }
                    leaf site-id {
                      type uint64;
                      description "Site ID";
                    }
                    uses lisp:mapping;
                  }
                }
              }
            }
            uses ms-counters;
          }
        }
```

```
          leaf mapping-system-type {
            type lisp:mapping-system-ref;
            description
              "A reference to the mapping system";
          }

          container summary {
            config false;
            description "Summary state information";

            leaf number-configured-sites {
              type uint32;
              description "Number of configured LISP sites";
            }
            leaf number-registered-sites {
              type uint32;
              description "Number of registered LISP sites";
            }
            container af-datum {
              description "Number of configured EIDs per each AF";

              list af-data {
                key "address-type";
                description "Number of configured EIDs for this AF";
                leaf address-type {
                  type lcaf:lisp-address-family-ref;
                  description "AF type";
                }
                leaf number-configured-eids {
                  type uint32;
                  description "Number of configured EIDs for this AF";
                }
                leaf number-registered-eids {
                  type uint32;
                  description "Number of registered EIDs for this AF";
                }
              }
            }
          }
          uses ms-counters;
        }
      }
    }
    <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:lisp-router-instances/lisp:lisp-router-instance:
    +--rw map-resolver!
       +--rw mapping-system-type?   lisp:mapping-system-ref
       +--rw ms-address?            inet:ip-address
```

6.2. Module Definition

```
<CODE BEGINS> file "ietf-lisp-mapresolver@2018-03-05.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
    "WG Web:   <http://tools.ietf.org/wg/lisp/>
     WG List:  <mailto:lisp@ietf.org>

     Editor:   Vina Ermagan
               <mailto:vermagan@cisco.com>

     Editor:   Alberto Rodriguez-Natal
               <mailto:natal@cisco.com>

     Editor:   Reshad Rahman
               <mailto:rrahman@cisco.com>";
  description
    "This YANG module defines the generic parameters for a LISP
     Map-Resolver. The module can be extended by vendors to define
     vendor-specific parameters and policies.

     Copyright (c) 2018 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 XXXX; see
     the RFC itself for full legal notices.
    ";
  // RFC Ed.: replace XXXX with actual RFC number and remove
  // this note
  reference "RFC XXXX";

  revision 2018-05-03 {
    description
      "Initial revision.";
    reference
      "https://tools.ietf.org/html/rfc6833";
  }
  identity mr {
    base lisp:lisp-role;
    description
      "LISP Map-Resolver.";
  }
  augment "/lisp:lisp/lisp:lisp-router-instances"
  +"/lisp:lisp-router-instance" {
    when "lisp:lisp-role/lisp:lisp-role-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 {
      presence "LISP Map-Resolver operation enabled";
      description
        "Map-Resolver parameters.";
      leaf mapping-system-type {
        type lisp:mapping-system-ref;
        description
        "A reference to the mapping system";
      }
      leaf ms-address {
        when "../mapping-system-type='lisp-mr:single-node-mapping-system'";
        type inet:ip-address;
        description
          "address to reach the Map Server when "
```

```
          + "lisp-mr:single-node-mapping-system is being used.";
      }
    }
  }
}
<CODE ENDS>
```

7. LISP-Address-Types Module

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

7.1. Module Definition

```
<CODE BEGINS> file "ietf-lisp-address-types@2018-03-05.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 2013-07-15;
  }
  import ietf-yang-types {
    prefix yang;
    revision-date 2013-07-15;
  }
  organization
    "IETF LISP (Locator/ID Separation Protocol) Working Group";
  contact
    "WG Web:   <http://tools.ietf.org/wg/lisp/>
     WG List:  <mailto:lisp@ietf.org>

     Editor:   Vina Ermagan
               <mailto:vermagan@cisco.com>

     Editor:   Alberto Rodriguez-Natal
               <mailto:natal@cisco.com>

     Editor:   Reshad Rahman
               <mailto:rrahman@cisco.com>";
  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) 2018 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 XXXX; see
        the RFC itself for full legal notices.

       ";
     // RFC Ed.: replace XXXX with actual RFC number and remove
     // this note
     reference "RFC XXXX";

     revision 2018-03-05 {
       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 = 'laddr: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 = 'laddr: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 = 'laddr: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 = 'laddr: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 = 'laddr: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 = 'laddr: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 = 'laddr: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 = 'laddr: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 {
                length "1..64";
              }
              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 = 'laddr: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 = 'laddr: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 = 'laddr: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>
```

8. Acknowledgments

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

9. IANA Considerations

This memo includes no request to IANA.

10. Security Considerations

Security Considerations TBD

11. Normative References

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

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <https://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, DOI 10.17487/RFC6832, January 2013, <https://www.rfc-editor.org/info/rfc6832>.

[RFC6833] Fuller, V. and D. Farinacci, "Locator/ID Separation Protocol (LISP) Map-Server Interface", RFC 6833, DOI 10.17487/RFC6833, January 2013, <https://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, <https://www.rfc-editor.org/info/rfc6836>.

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

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

Authors' Addresses

Vina Ermagan
Cisco Systems
San Jose, CA
USA

Email: vermagan@cisco.com

Alberto Rodriguez-Natal
Cisco Systems
San Jose, CA
USA

Email: natal@cisco.com

Florin Coras
Cisco Systems
San Jose, CA
USA

Email: fcoras@cisco.com

Carl Moberg
Cisco Systems
San Jose, CA
USA

Email: camoberg@cisco.com

Reshad Rahman
Cisco Systems
Canada

Email: rrahman@cisco.com

Albert Cabellos-Aparicio
Technical University of Catalonia
Barcelona
Spain

Email: acabello@ac.upc.edu

Fabio Maino
Cisco Systems
San Jose, CA
USA

Email: fmaino@cisco.com

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: September 4, 2018

A. Rodriguez-Natal
V. Ermagan
F. Maino
Cisco Systems
A. Cabellos
Technical University of Catalonia
March 3, 2018

LISP control-plane for Identifier Locator Addressing (ILA)
draft-rodrigueznatal-ila-lisp-00

Abstract

This document specifies how to use the LISP control-plane to support an Identifier Locator Addressing (ILA) data-plane. In particular, it describes how ILA data-plane components can use the LISP control-plane to dynamically resolve and register Identifier-to-Locator mappings as well as Endpoint Address to Identifier/Locator mappings.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The Identifier Locator Addressing (ILA) [I-D.herbert-intarea-ila] is an IPv6 data-plane protocol that relies on address splitting for ID/location separation. Part of the IPv6 address expresses the

Identifier of an endpoint, the immutable identity of the node (e.g. task, end-host, mobile device, etc), while another part represents its Locator, the topological location of the endpoint, which can be dynamic. The Locator defines where the Identifier is currently attached to the network and is used to route the packets through the ILA domain. To do so, ILA Locators are prepended to the ILA Identifier to form a routable ILA address (bitwise equivalent to an IPv6 address).

The Identifier of an endpoint is unique and permanent for its lifetime, meanwhile its locator is not fixed and subject to change over time. A control-plane protocol to resolve Identifier-to-Locator mappings is needed in order to use the ILA data-plane. The ILA data-plane is agnostic to the control-plane mechanism in place and therefore different control-plane protocols have been proposed [I-D.lapukhov-bgp-ila-afi] [I-D.herbert-ila-ilamp]. This document specifies how the Locator/ID Separation Protocol (LISP) control-plane [I-D.ietf-lisp-rfc6833bis] can be used to support the operation of the ILA data-plane, including the resolution of the Identifier-to-Locator mappings and the Endpoint Address to ILA Identifier/Locator mappings.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. LISP Overview

TBD

4. ILA LCAF

To support ILA mappings and associated data using the LISP control-plane the following LISP Canonical Address Format (LCAF) [RFC8060] is introduced. The ILA LCAF type defines several subtypes to carry different ILA information. This document refers to the different subtypes of the ILA LCAF using the syntax "ILA-Subtype" LCAF (e.g. ILA-Identifier). All the ILA subtypes follow the format defined below:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           AFI = 16387         |     Rsvd1     |     Flags     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Type = ILA   |Subtype| Rsvd2 |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         ILA Subtype...                        ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

ILA LCAF

The fields specific to the the ILA LCAF Type are defined below. For definition of the rest of fields see the LCAF specification [RFC8060].

Subtype: This is the ILA LCAF Subtype. This field indicates which particular ILA format the ILA LCAF encodes. Currently the following Subtypes are defined:

0x0: Reserved. Reserved for future use.

0x1: Identifier Subtype. Used to encode ILA Identifiers as described in Section 4.1

0x2: Locator Subtype. Used to encode ILA Locators as described in Section 4.2

0x3: SIR Prefix Subtype. Used to encode SIR Prefixes as described in Section 4.3

4.1. Identifier Subtype

The Identifier subtype (aka ILA-Identifier LCAF) can be used to carry ILA Identifiers in the LISP control-plane signaling. The ILA specification [I-D.herbert-intarea-ila] defines different Identifiers formats that can be used in the data-plane. The same Identifier formats are described in this document for the ILA-Identifier LCAF Subtype. When used in this document, each Identifier format is referred by the code point defined in [I-D.herbert-intarea-ila], e.g. "ILA-Identifier-0x3" LCAF. The ILA data-plane formats are bitwise compatible with their correspondent LCAF formats. It is thus possible for an ILA data-plane device to resolve the Locator for a particular ILA Identifier even if the ILA data-plane device does not understand that particular Identifier type.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           AFI = 16387         |     Rsvd1     |     Flags     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Type = ILA   |  0x1  |E|Rsvd2|             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type|0|                                                       |
+-+-+-+-+              Identifier (60 bits)                     |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

ILA-Identifier LCAF

The fields specific to the the ILA-Identifier LCAF Subtype are defined below.

Type: This is the Type of Identifier as defined in [I-D.herbert-intarea-ila].

E: This is the "Endpoint Address Mapping" bit. If the ’E’ bit is set to 1, it indicates that the Identifier is being resolved to an Endpoint Address (see Section 6.2). The ’E’ bit is set to 0 otherwise.

Identifier: This is a variable length field that encodes an ILA Identifier. The length of the Identifier can be inferred from the Length field of the LCAF. The representation above uses an Identifier size of 64 bits (including the first 4 bits with the Type).

Using Identifier size of 64 bits (including the first 4 bits with the Type), the different types of Identifiers are encoded in the ILA-Identifier LCAF as described below. The common part of the ILA-Identifier LCAF is not shown.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0x0 |0|                                                       |
+-+-+-+-+              Interface Identifier                     |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Interface Identifier (0x0)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0x1 |0|                                                       |
+-+-+-+-+            Locally Unique Identifier                  |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Locally Unique Identifier (0x1)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0x2 |0|                     VNID                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Virtual IPv4                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Virtual IPv4 Identifier (0x2)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0x3 |0|                     VNID                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Virtual Unicast IPv6 (lower 32 bits)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Virtual Unicast IPv6 Identifier (0x3)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0x4 |0|                     VNID                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Scope |      Virtual Multicast IPv6 (lower 28 bits)           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Virtual Multicast IPv6 Identifier (0x4)

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | 0x5 |0|         Non-Local Address Identifier                  |
    +-+-+-+-+                      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+|
    |                              |              0x0               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Non-Local Address Identifier (0x5)

4.2. Locator Subtype

The Locator subtype (aka ILA-Locator LCAF) can be used to carry ILA Locators in the LISP control-plane signaling.

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |           AFI = 16387         |     Rsvd1     |     Flags     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Type = ILA   |  0x2  |C|Rsvd2|             Length            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                      Locator (64 bits)                        |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

ILA LCAF

The fields specific to the the ILA-Locator LCAF Subtype are defined below.

C: This is the "Checksum-Adjustment-Needed" bit. If the ’C’ bit is set to 1 it indicates that an ILA data-plane device has to compute the checksum adjustment as described in [I-D.herbert-intarea-ila] when sending ILA packets to this Locator. The ’C’ bit is set to 0 otherwise. See Section 7.7 for more details.

Locator: This is a variable length field that encodes an ILA Locator. The length of the Locator can be inferred from the Length field of the LCAF. The representation above uses an Locator size of 64 bits.

4.3. SIR Prefix Subtype

The SIR Prefix subtype (aka ILA-SIR LCAF) can be used to encode the SIR prefix when different ILA domains co-exist as described in Section 7.5.

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           AFI = 16387         |     Rsvd1     |     Flags     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Type = ILA   |  0x3  | Rsvd2 |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|SIR Prefix Len |                  SIR Prefix ...               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                       ... SIR Prefix                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             AFI = x           |          Address...           ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

ILA-SIR LCAF

The fields specific to the the ILA-SIR LCAF Subtype are defined below.

SIR Prefix Len: This field indicates the length of the SIR Prefix that follows.

Locator: This is a variable length field that encodes an ILA SIR Prefix.

5. Device Roles and Provision

The ILA specification [I-D.herbert-intarea-ila] defines different ILA data-plane devices (i.e. devices that can perform ILA transformations of SIR addresses to/from ILA addresses), namely ILA-Router (ILA-R), ILA-Host (ILA-H) and ILA-Node (ILA-N). This documents relies on the terminology introduced in [I-D.herbert-intarea-ila] but uses ILA-Router and ILA-Node denominations to distinguish between ILA data-plane devices that have a complete map-cache set (ILA-R) versus those that only have an incomplete map-cache (ILA-N). For the purpose of this document, it is assumed that an ILA-N has endpoints assigned to it to which it has direct connectivity (if it is an ILA-Host it may be even hosting the endpoints itself). On the contrary, no endpoint assignment is assumed for an ILA-R (although not precluded). This section describes in general terms the role and required provisioning of the different devices involved in an ILA-LISP deployment, for operational details of these devices see Section 6

To avoid verbosity on the description of the provisioning requirements listed below, there are two things that are assumed to be configured in all the devices belonging to a given ILA domain:

- o SIR prefix(es) of the ILA domain: This prefix serves to identify traffic belonging to the ILA domain. It is also used to differentiate across different ILA domains where several domains share the same infrastructure.

- Control-Plane Identifier: a given Identifier within the ILA domain is reserved to be used when sending control-plane messages across devices. This Identifier is concatenated with the Locator of the device that the control-plane message is addressed towards. When receiving an ILA packet with this special Identifier, the packet will be delivered to the control-plane process of the device.

5.1. Map Server (MS)

A MS has the complete mapping information for all the Identifiers in the ILA domain. If the Identifier space is divided into different shards, then each MS is responsible for a particular shard of Identifiers. Then, for the Identifiers of its shard, the MS has the complete mapping information. The mapping information at an MS can be populated by the ILA devices registering their local mappings and/or by an external source. A MS has to be pre-provisioned with the following:

Shard index: of the shard the MS is responsible for (if any).

5.2. Map Resolver (MR)

A MR receives requests for mappings from ILA data-plane devices and forwards them to the appropriate MS. If needed, a MR is also able to find which MS is associated with a particular shard. See Section 7.3 for a discussion on different options regarding how a MR can find the appropriate MS to forward a given mapping request.

5.3. ILA-Router (ILA-R)

An ILA-R has a complete map-cache for the mappings in the domain. If shards are used, then each ILA-R is assigned to a particular shard of Identifiers for which it has a complete map-cache of mappings. An ILA-R subscribes (as described in Section 6.1.3) to a MS (or to the MS responsible for its shard) to populate and keep its map-cache updated. Normally, an ILA-R has no endpoint (e.g. task, user-endpoint, etc) directly attached. Instead, it serves to translate packets that were not translated by an ILA-N. To do so, as described in [I-D.herbert-intarea-ila], an ILA-R announces the SIR prefix (plus its shard index if needed) as an "anycast" address on the underlay to attract traffic towards itself. See also Section 7.2 for discussion on the case of co-locating the MS with the ILA-R. An ILA-R has to be pre-provisioned with the following:

Shard index: of the shard the ILA-R is assigned to.

5.4. ILA-Node (ILA-N)

This document uses the term ILA-Node to refer to an ILA translating device that does not have a complete map-cache, in contrast with ILA-Router that has a complete map-cache for the domain (or its shard). Each ILA-N has a set of endpoints (and associated Identifiers) assigned to it for which it has complete mapping information. An ILA-N registers its local mapping information into a MS (or set of MSs) as described in Section 6.1.2. A given ILA-N may have Identifiers from different shards, in that case per each Identifier it has to register the mapping information in the appropriate MS (the one handling the shard of the Identifier). Contrary to an ILA-R, the ILA-N does not have a full map-cache for remote Identifiers, but rather it populates its map-cache on demand (following the mechanisms described in Section 6.1.1) based on the actual data-plane traffic. An ILA-N has to be pre-provisioned with the following:

Identifiers: that the ILA-N is responsible for (if they are not created or auto-discovered by the ILA-N).

Locators: to use on the ILA underlay (if they are not created or auto-discovered by the ILA-N).

VNID / Tenant-Prefix pairs: if virtualization is used (see also Section 6.2).

MR Locator: to request mappings for remote identifiers.

MS Locator: of the MS (or MSs) responsible for the Identifiers assigned to the ILA-N.

Checksum adjustment setting: to indicate if the ILA-N has to perform or not checksum adjustment (see also Section 7.7).

6. Operation

An ILA data-plane can leverage the LISP control-plane to support different aspects of its operation. The main function provided by the LISP control plane is resolving the Identifier to Locator mappings. In addition, ILA can also use the LISP control-plane to dynamically learn the ILA Identifier associated to an Endpoint Address. These two steps can also be combined into a single resolution exchange.

6.1. ILA Identifier to Locator Mappings

This section describes how ILA devices can use the LISP control-plane to resolve, register and keep updated the Identifier to Locator mappings required for the operation of the ILA data-plane.

6.1.1. Resolution

When an ILA-N has to send traffic towards a remote Identifier for which it does not have the associated Locator, it has to obtain it first from a MS. To do so, it follows the mechanisms described in [I-D.ietf-lisp-rfc6833bis] and sends a Map-Request towards one of its configured MRs. This Map-Request includes as EID the ILA Identifier of the remote endpoint encoded using the LCAF defined in Section 4.1 As a response to this Map-Request, the ILA-N will get a Map-Reply from the MS with the Locator(s) associated with the remote Identifier (if any). Locators are carried in the Map-Reply as RLOCs and are encoded using the LCAF defined in Section 4.2. In the current ILA specification, Identifiers are considered to be in a flat, non-hierarchical space. Therefore, when resolving a single Identifier the "EID mask-len" of the Map-Request and Map-Reply is set to the length of the Identifier. As specified in [I-D.ietf-lisp-rfc6833bis], an ILA-N can use the priority and weight information conveyed in the Map-Reply message to load balance data-plane traffic across the different Locators for the remote Identifier. While the mapping is being resolved via the Map-Request/Map-Reply process, the ILA-N can send the data packets to the underlay using the SIR address. In that way, they can be attracted and translated at an ILA-R. See also Section 8.2 for discussion on how the ILA-N can protect itself from malicious endpoints trying to artificially force map-cache misses (and subsequent Map-Requests).

6.1.2. Registration

An ILA-N registers its local Identifier-to-Locator mappings in the appropriate MSs (i.e. those handling its Identifiers) by sending Map-Register messages following the process documented in [I-D.ietf-lisp-rfc6833bis]. To do so, it uses the ILA-Identifier LCAF defined in Section 4.1 as EID in the Map-Register. Similarly to the mapping resolution process, the "EID mask-len" of the Map-Register is fixed to the length of the Identifier. The ILA-N includes its local Locators in the Map-Register using the ILA-Locator LCAF defined in Section 4.2. As described in [I-D.ietf-lisp-rfc6833bis], the mapping registration may happen periodically as well as when there is a change in the mapping(s) that the ILA-N is registering.

6.1.3. PubSub

When requesting a mapping to populate its map-cache, an ILA-N can subscribe to updates on the mapping using the mechanisms described in [I-D.rodrigueznatal-lisp-pubsub]. Similarly, an ILA-R subscribes using [I-D.rodrigueznatal-lisp-pubsub] to receive updates for all the Identifier-Locator mappings in the domain (or in its shard). To subscribe to all Identifier mappings in the ILA domain, the ILA-R sets the "EID mask-len" in the Map-Request to 0 and uses an ILA-Identifier LCAF with all the Identifier bits set to 0. To subscribe to all Identifier mappings in a particular shard, the ILA-R sets the "EID mask-len" in the Map-Request to the shard index length and uses an ILA Identifier LCAF with the proper shard index set and the rest of the Identifier bits set to 0.

6.1.4. Mobility

Mobility of Identifiers is supported by the mechanisms described in [I-D.ietf-lisp-eid-mobility]. As described there, when an Identifier moves to a different ILA-N, its previous ILA-N is notified with the new Locator(s) for the Identifier. When traffic is received at the old Locator, the ILA-N there can use the updated Identifier-Locator mapping information to replace the old Locator with the new Locator and forward the traffic back to the underlay. In the interim between the ILA-N detects that the Identifier has moved but the notification with the new Locator is yet to be received, the ILA-N can translate received traffic for the Identifier to the SIR address and forward it back to the underlay (to be intercepted, translated and forwarded by an ILA-R).

Following [I-D.ietf-lisp-eid-mobility], when the old ILA-N receives traffic addressed for the Identifier that is no longer locally connected, it sends a Solict-Map-Request (SMR) to the Locator associated with the source Identifier to inform it that it should update its map-cache. Note that when ILA is used as the data-plane, the source Locator may not be present in the received data packet and a mapping resolution (to find the ILA-N that originated the packet) may be needed before the SMR can be sent. Note also that, if the data packet was translated and sent by an ILA-R, the source Identifier will not resolve to the Locator of the ILA-R (but instead to the ILA-N where the source Identifier is attached). For this version of the document, the case of sending this SMR to an ILA-R is not considered.

6.2. Endpoint Address to ILA Identifier/Locator Mappings

The ILA data-plane [I-D.herbert-intarea-ila] defines some cases where the address used by an endpoint is not a SIR address. In those cases, the Endpoint Address needs to be mapped to an ILA Identifier before an ILA address (or SIR address) can be formed. These mappings of Endpoint Addresses to ILA Identifiers can be statically provisioned at the ILA-N or can also be resolved via the LISP control-plane. There are currently two cases defined in [I-D.herbert-intarea-ila] where an endpoint does not use a SIR address and requires a mapping of Endpoint Address to ILA Identifier.

- o Virtualization: In virtualization scenarios, the endpoints use virtual addresses (with a Tenant Prefix in the case of IPv6) rather than SIR addresses. Before packets can be sent over the ILA underlay, the Tenant Prefix has to be converted into a VNID. Instead of pre-provisioning the Tenant Prefix to VNID pairs in advance, the ILA data-plane can also use the LISP control-plane to resolve the mapping of Tenant Prefix to VNID. Dynamic resolution instead of static provisioning can be specially useful for cases of cross-communication between different virtual networks (since the mapping of a remote Virtual Prefix to VNID may not be available at the ILA-N). Once the ILA-N has resolved the VNID associated with a Tenant Prefix, it can cache this information and only request Identifier to Locator mappings for new remote Endpoint Addresses using the same Tenant Prefix. Note that for virtualization cases using IPv4, the current version of this document assumes that the VNID has to be pre-provisioned since there is not Tenant Prefix that can be resolved into a VNID.

- o Non-Local Addresses: ILA uses the concept of Non-Local Addresses to refer to Endpoint Addresses that do not belong to the SIR prefix(es) of the domain. To use Non-Local Addresses with an ILA data-plane, they need to be first converted into an ILA identifier (of 44 bits in the current ILA specification). The LISP control-plane can be used by the ILA data-plane to retrieve the mappings of Non-Local Addresses to Identifiers (on packet transmission) and of Identifiers to Non-Local Addresses (on packet reception). If the ILA data-plane devices are also performing the assignment of Non-Local Addresses to ILA Identifiers, the LISP control-plane can also be used to register this assignment into the Mapping System.

This section covers the resolution (including reverse resolution) and registration of Endpoint Address mappings. Contrary to the Identifier to Locator mappings, the mappings of Endpoint Address to Identifier are not expected to change once they have been established. Therefore, the cases of PubSub and mobility are not considered in this section.

6.2.1. Resolution

As with Identifier to Locator mapping resolution, the resolution of Endpoint Address to Identifier is done via the Map-Request/Map-Reply exchange specified in [I-D.ietf-lisp-rfc6833bis]. It is assumed that in the scenario where LISP is used to resolve Endpoint Address to Identifier mappings, the MR is able to find the MS storing the requested Endpoint Address mapping.

When Endpoint Addresses are carried as EIDs in LISP control messages they are encoded using the same format the endpoint is using (i.e. IPv6 in the currently defined cases). The Identifier associated to the Endpoint Address is returned in the Map-Reply as an RLOC with priority 255 and encoded using the LCAF defined in Section 4.1. Note that when resolving an Endpoint Address to Identifier mapping, the Identifier to Locator mapping can be included as well. In other words, the Locators can also be encoded as RLOCs in the Map-Reply returned by the MS.

In some cases (such in the Non-Local Address case) some ILA devices may need to perform a reverse resolution of the Endpoint Address mapping (i.e. obtain the Endpoint Address associated with a given Identifier). In those cases, the Identifier is sent as EID in the Map-Request with the ’E’ bit (defined in Section 4.1) set to ’1’. The ’E’ bit is used to signal that the requested mapping is "Identifier to Endpoint Address" and distinguish the request from the default "Identifier to Locator" resolution that is triggered when sending an Identifier in the Map-Request. On this reverse resolution, the MS will return the Endpoint Address in the Map-Reply encoded as an RLOC with priority 255.

6.2.2. Registration

When the assignment of Endpoint Address to ILA Identifier is performed by the ILA-N, the ILA-N can register this assignment into its MS(s). The ILA-N encodes the Endpoint Address as EID (using the same format the endpoint is using) and the Identifier as RLOC with priority 255 (using the ILA-Identifier LCAF). It is assumed that the MS(s) assigned to the ILA-N are able to understand and store the Endpoint Address to Identifier mappings generated by the ILA-N. Similarly to the resolution case, the Identifier to Locator mapping can be also included when registering the Endpoint Address mapping via means of providing too the Locators as RLOCs in the Map-Register message.

7. Deployment Considerations

This section discusses different options and deployment scenarios to consider when deploying an ILA data-plane using a LISP control-plane.

7.1. Protocol Transport

LISP as defined in [I-D.ietf-lisp-rfc6833bis] runs over a UDP transport, however the exact same signaling can be used over a TCP transport without affecting the protocol operation. If a TCP transport is available, then the mechanisms described in [I-D.kouvelas-lisp-map-server-reliable-transport] can also be used to optimize the LISP control-plane protocol operation when this runs over a reliable channel.

7.2. ILA-R and MS Co-location

The logical functions of a MS and an ILA-R serving the same domain (or shard) can be co-located and assigned to the same box. In that case, the ILA-R does not need to subscribe to the mappings in the domain (or shard) since they are locally available.

In this co-location scenario it is also possible for the MS+ILA-R box to send an unsolicited Map-Notify message (as described in [I-D.rodrigueznatal-lisp-pubsub]) to populate the map-cache of an ILA-N sending SIR packets towards the MS+ILA-R. This can be done instead (or in addition) to the ILA-N sending a Map-Request message to populate its cache.

7.3. Mapping System Internal Resolution

For small deployments where each MS has the complete mapping information for the domain, the MRs may just be provisioned with the Locators for all the MSs. They can then do load balancing across the MSs based on different metrics (e.g. latency, load, etc)

If the domain is split into shards, there are different ways for a MR to find the MS that corresponds to a given shard. Some options can include LISP-DDT [RFC8111] or LISP-ALT [RFC6836] for instance.

There is also the option that a backend database is used as Mapping System, in which case both the MRs and MSs are just interfaces to interact with the backend database. In that scenario, the database internal implementation will find the appropriate instance that is hosting the requested mapping.

7.4. Mapping System Replication and Synchronization

For reliability and latency purposes, several MSs can be deployed for the same domain or the same shard. In that scenario, it is required to have a mechanism to synchronize the mapping information across them. One option is that, as described in [I-D.ietf-lisp-rfc6833bis], the ILA-Ns register their local mappings to several MSs. Alternatively, when a backend database is in place, mechanisms specific to the database implementation can be leveraged to provide synchronization across different replicas.

7.5. Multiple ILA Domains

When different ILA domains co-exist using the same infrastructure, it may be needed to distinguish the particular domain to which an Identifier belongs. In that case, the Identifiers and mappings must be qualified with the appropriate ILA domain for the control-plane operation. This can be done by prepending the ILA-SIR LCAF described in Section 4.3 to the ILA Identifiers or Endpoint Addresses sent as EIDs in LISP messages.

7.6. Proactive Mapping Push

Optionally, when a MS receives a Map-Request for an Identifier, it can send a proactive Map-Notify towards the ILA-N associated with that Identifier. In this Map-Notify the MS includes the mapping associated with the Identifier that triggered the Map-Request. This will pre-populate the map-cache of the destination ILA-N and provide the ILA-N the mapping required to handle the returning traffic. To support this mode of operation (and following [I-D.ietf-lisp-rfc6833bis]), the source ILA-N must include in the Map-Request the source Identifier that triggered the Map-Request (encoded as "Source EID" using the ILA-Identifier LCAF defined in Section 4.1).

7.7. Checksum Adjustment per Locator

While performing ILA transformations, ILA data-plane devices optionally perform checksum adjustments to keep the transport checksum neutral to the transformation. As an alternative to statically configuring the checksum-neutral adjustment option per ILA-N (or ILA-R), the Locators associated with a particular Identifier can be qualified with the requested selection regarding checksum-neutral adjustment. Then, the need to perform or not this checksum adjustment when sending traffic to a particular Locator can be stored and retrieved from the MSs encoded in the ILA Locator LCAF defined in Section 4.2.

8. Security Considerations

8.1. Signaling Protection

Map-Register, Map-Notify and Map-Reply messages have a field for authentication data. As described in [I-D.ietf-lisp-rfc6833bis] a shared key is required between the data-plane devices and their associated MSs to sign and secure the signaling. Additional authentication and integrity protection can be enabled by using [I-D.ietf-lisp-sec]. Complementary, if a TCP session is in place between the ILA data-plane elements and the LISP control-plane components, then TLS can be used to provide authentication and integrity protection.

8.2. Map-Cache Attacks

Malicious endpoints can try to deplete the map-cache and/or overload the Map-Request channel of an ILA-N. To prevent against these attacks, the ILA-N should implement efficient heavy hitters counters such as Count-Min Sketch [CMS] to prevent data-plane traffic from certain endpoints to trigger further control-plane processing once a threshold has been reached. In addition, similar mechanisms can be used to protect popular map-cache entries from eviction when the map-cache space is being depleted.

9. Acknowledgments

The authors would like to thank Sri Gundavelli and Marc Portoles-Comeras for their comments and feedback while writing this document.

10. IANA Considerations

Following the guidelines of [RFC5226], this document requests IANA to update the "LISP Canonical Address Format (LCAF) Types" Registry defined in [RFC8060] to allocate the following assignment:

Value #	LISP LCAF Type Name	Reference
TBD	ILA	Section 4

Table 1: ILA LCAF assignment

11. Normative References

[CMS] Cormode, G. and S. Muthukrishnan, "An improved data stream summary: the count-min sketch and its applications", Journal of Algorithms 55(1), 58-75, April 2005.

[I-D.herbert-ila-ilamp]
Herbert, T., "Identifier Locator Addressing Mapping Protocol", draft-herbert-ila-ilamp-00 (work in progress), December 2017.

[I-D.herbert-intarea-ila]
Herbert, T. and P. Lapukhov, "Identifier-locator addressing for IPv6", draft-herbert-intarea-ila-00 (work in progress), October 2017.

[I-D.ietf-lisp-eid-mobility]
Portoles-Comeras, M., Ashtaputre, V., Moreno, V., Maino, F., and D. Farinacci, "LISP L2/L3 EID Mobility Using a Unified Control Plane", draft-ietf-lisp-eid-mobility-01 (work in progress), November 2017.

[I-D.ietf-lisp-rfc6833bis]
Fuller, V., Farinacci, D., and A. Cabellos-Aparicio, "Locator/ID Separation Protocol (LISP) Control-Plane", draft-ietf-lisp-rfc6833bis-07 (work in progress), December 2017.

[I-D.ietf-lisp-sec]
Maino, F., Ermagan, V., Cabellos-Aparicio, A., and D. Saucez, "LISP-Security (LISP-SEC)", draft-ietf-lisp-sec-14 (work in progress), October 2017.

[I-D.kouvelas-lisp-map-server-reliable-transport]
Cassar, C., Leong, J., Lewis, D., Kouvelas, I., and J. Arango, "LISP Map Server Reliable Transport", draft-kouvelas-lisp-map-server-reliable-transport-04 (work in progress), September 2017.

[I-D.lapukhov-bgp-ila-afi]
Lapukhov, P., "Use of BGP for dissemination of ILA mapping information", draft-lapukhov-bgp-ila-afi-02 (work in progress), October 2016.

[I-D.rodrigueznatal-lisp-pubsub]
Rodriguez-Natal, A., Ermagan, V., Leong, J., Maino, F., Cabellos-Aparicio, A., Barkai, S., Farinacci, D., Boucadair, M., Jacquenet, C., and S. Secci, "Publish/Subscribe Functionality for LISP", draft-rodrigueznatal-lisp-pubsub-02 (work in progress), February 2018.

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

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <https://www.rfc-editor.org/info/rfc5226>.

[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, <https://www.rfc-editor.org/info/rfc6836>.

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

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

Authors' Addresses

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

Email: natal@cisco.com

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

Email: vermagan@cisco.com

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

Email: fmaino@cisco.com

Albert Cabellos
Technical University of Catalonia
Barcelona
Spain

Email: acabello@ac.upc.edu

LISP Working Group A. Rodriguez-Natal
Internet-Draft Cisco Systems
Updates: 6830 (if approved) A. Cabellos-Aparicio
Intended status: Experimental Technical University of Catalonia
Expires: March 30, 2018 V. Ermagan
F. Maino
Cisco Systems
S. Barkai
Fermi Serverless
September 26, 2017

MS-originated SMRs
draft-rodrigueznatal-lisp-ms-smr-04

Abstract

This document extends [RFC6830] to allow Map Servers to send SMR messages.

This extension is intended to be used in some SDN deployments that use LISP as a southbound protocol with (P)ITRs that are compliant with [RFC6830]. In this use-case mapping updates do not come from ETRs, but rather from a centralized controller that pushes the updates directly to the Mapping System. In such deployments, Map Servers will benefit from having a mechanism to inform directly (P)ITRs about updates in the mappings they are serving.

Status of This Memo

Table of Contents

1. Introduction

The Locator/ID Separation Protocol (LISP) [RFC6830] splits current IP addresses in two different namespaces, Endpoint Identifiers (EIDs) and Routing Locators (RLOCs). LISP uses a map-and-encap approach that relies in two entities, the Mapping System and the Tunnel Routers. The Tunnel Routers are deployed at LISP sites edge points and perform encapsulation and decapsulation of LISP data packets. The Mapping System is a distributed database that stores and disseminates EID-RLOC bindings across different Map-Servers. LISP Tunnel Routers keep a cache of EID-RLOC mappings pulled from the Mapping System.

There are several ways to keep this cache updated as described in [RFC6830]. Among them, the Solicit Map-Request (SMR) message allows to explicitly signal (P)ITRs to let them know that some of their cached mappings may be outdated. However, vanilla LISP as described in [RFC6830] only considers SMR messages to be sent by an ETR. This document extends [RFC6830] to cover the case where SMRs can be sent also by a Map Server (MS).

This document introduces changes in the MS specification allowing them to send SMR messages, however it does not require any modification in the (P)ITRs. This document is backwards compatible and enables upgraded MS to interoperate via SMRs with legacy (P)ITRs that only implement [RFC6830].

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Map Server extension

This document enables MS to generate and send SMR messages towards (P)ITRs. SMRs originated in a MS follow the same format described in [RFC6830]. Besides the fact that they are sent from a MS, there is no difference between an SMR originated in an ETR and one originated in a MS.

When a MS generates an SMR, it uses as source-EID the EID-prefix it wants the (P)ITR to send the SMR-invoked Map-Request for. The EID included in the EID-record field is the one belonging to the (P)ITR the MS sends the SMR towards. As source locator for the SMR message, the MS uses one of its available locators. This has implications in the processing of the SMR at the (P)ITR as described in Section 4

When the MS has to send an SMR is implementation specific. However, as specified in [RFC6830] and noted in Section 7, SMRs MUST be rate-limited. It must be noted as well that, as described in Section 3, a MS that sends an SMR may not receive the SMR-invoked Map-Request that the (P)ITR generates as response to the SMR.

3. Interoperability with legacy (P)ITRs

This document introduces no changes in the specification of (P)ITRs and thus it is backwards compatible with legacy equipment only compliant with [RFC6830]. However, since SMRs were designed to be sent by ETRs, and legacy (P)ITRs expect to receive SMRs only from ETRs, the implications of sending SMRs from a MS are discussed in this section.

As indicated in Section 2, the MS generates the SMR message using one of its locators as source locator. However, this locator will not be present in the Locator-Set cached for that EID-prefix at the (P)ITR. Following [RFC6830], upon receiving the SMR message, the (P)ITR will check if the source locator is in the Locator-Set cached for that EID-record. Since it is not, the (P)ITR will send the SMR-invoked

Map-Request always to the Mapping System and never to the source locator of the SMR message. This means that a MS can not force an SMR-invoked Map-Request to be sent directly towards itself. However, it is possible that the Mapping System in use is instantiated (even partially) by the MS originator of the SMR. In that case, it may be that the SMR-invoked Map Request will eventually reach the MS, either directly or after being internally forwarded through the Mapping System.

4. Deployment considerations

The extension defined in this document may be useful in scenarios where the MS wants to signal (P)ITRs about changes on mappings it is serving. For instance, when the MS is keeping track of the (P)ITRs that are requesting its mappings and wants to inform them intermediately whenever a mapping is updated.

SDN deployments that use LISP as a southbound protocol are particularly suitable to take advantage of this extension. On the SDN scenario, mapping updates will unlikely come from ETRs, but rather from a centralized entity that pushes the updates directly to the Mapping System. In such deployments, Map Servers will benefit from having a mechanism to inform directly (P)ITRs about updates in the mappings they are serving.

Due to scalability and security concerns, it is RECOMMENDED that this extension is only applied in intra-domain scenarios where all LISP devices are within a single administrative domain.

To limit the impact of the extension and to ease its integration with the rest of LISP signaling and operation, it is RECOMMENDED that the MS only sends SMR messages for those mappings it is proxy-replying for.

5. Acknowledgments

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

As described in [RFC6830], the SMR messages and the SMR-invoked Map-Request MUST be rate-limited. This does not change with the extension proposed in this document.

The (P)ITRs receiving SMRs from the MS will send Map-Request messages to the Mapping System to retrieve authoritative mappings. It is

RECOMMENDED that the security mechanism described in [I-D.ietf-lisp-sec] and [RFC8111] are in place to secure the mapping retrieval and protect against unsolicited messages or hijacking attacks.

8. Normative References

[I-D.ietf-lisp-sec]
Maino, F., Ermagan, V., Cabellos-Aparicio, A., and D. Saucez, "LISP-Security (LISP-SEC)", draft-ietf-lisp-sec-13 (work in progress), September 2017.

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

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

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

Authors’ Addresses

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

Email: natal@cisco.com

Albert Cabellos-Aparicio
Technical University of Catalonia
Barcelona
Spain

Email: acabello@ac.upc.edu

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

Email: vermagan@cisco.com

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

Email: fmaino@cisco.com

Sharon Barkai
Fermi Serverless
CA
USA

Email: sharon@fermicloud.io

LISP Working Group
Internet-Draft
Intended status: Experimental
Expires: April 21, 2018

A. Rodriguez-Natal
V. Ermagan
J. Leong
F. Maino
Cisco Systems
A. Cabellos-Aparicio
Technical University of Catalonia
S. Barkai
Fermi Serverless
D. Farinacci
lispers.net
M. Boucadair
C. Jacquenet
Orange
S. Secci
LIP6 UPMC
October 18, 2017

Publish/Subscribe Functionality for LISP
draft-rodrigueznatal-lisp-pubsub-01

Abstract

This document specifies an extension to the use of Map-Request to enable Publish/Subscribe (PubSub) operation for LISP.

Table of Contents

1. Introduction

The Locator/ID Separation Protocol (LISP) [RFC6830] splits current IP addresses in two different namespaces, Endpoint Identifiers (EIDs) and Routing Locators (RLOCs). LISP uses a map-and-encap approach that relies on (1) a Mapping System (basically a distributed database) that stores and disseminates EID-RLOC mappings and on (2) LISP tunnel routers (xTRs) that encapsulate and decapsulate data packets based on the content of those mappings.

ITRs/RTRs/PITRs pull EID-to-RLOC mapping information from the Mapping System by means of an explicit request message. [RFC6830] indicates how ETRs can tell ITRs/RTRs/PITRs about mapping changes. This document presents a Publish/Subscribe (PubSub) extension in which the Mapping System can notify ITRs/RTRs/PITRs about mapping changes. When this mechanism is used, mapping changes can be notified faster and can be managed in the Mapping System versus the LISP sites.

In general, when an ITR/RTR/PITR wants to be notified for mapping changes for a given EID-prefix, the following steps occur:

(1) The ITR/RTR/PITR sends a Map-Request for that EID-prefix.

(2) The ITR/RTR/PITR sets the Notification-Requested bit (N-bit) on the Map-Request and includes its xTR-ID.

(3) The Map-Request is forwarded to one of the Map-Servers that the EID-prefix is registered to.

(4) The Map-Server creates subscription state for the ITR/RTR/PITR on the EID-prefix.

(5) The Map-Server sends a Map-Notify to the ITR/RTR/PITR to acknowledge the successful subscription.

(6) When there is an RLOC-set change for the EID-prefix, the Map-Server sends a Map-Notify message to each ITR/RTR/PITR in the subscription list.

(7) Each ITR/RTR/PITR sends a Map-Notify-Ack to acknowledge the received Map-Notify.

This operation is repeated for all EID-prefixes for which ITR/RTR/PITR want to be notified. The ITR/RTR/PITR can set the N-bit for several EID-prefixes within a single Map-Request

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Deployment Assumptions

The specification described in this document makes the following deployment assumptions:

(1) A unique 128-bit xTR-ID identifier is assigned to each xTR.

(2) Map-Servers are configured in proxy-reply mode, i.e., they are solicited to generate and send Map-Reply messages for the mappings they are serving.

(3) There can be either a soft-state or hard-state security association between the xTRs and the Map-Servers.

The distribution of xTR-IDs and the management of security associations are out of the scope of this document.

4. Map-Request Additions

Figure 1 shows the format of the updated Map-Request [I-D.ietf-lisp-rfc6833bis] to support the PubSub functionality.

```
        0                   1                   2                   3
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |Type=1 |A|M|P|S|p|s|m|I|   Reserved  |   IRC   | Record Count  |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         Nonce . . .                           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         . . . Nonce                           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |         Source-EID-AFI        |   Source EID Address  ...     |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |         ITR-RLOC-AFI 1        |    ITR-RLOC Address 1  ...    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                              ...                              |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |         ITR-RLOC-AFI n        |    ITR-RLOC Address n  ...    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     / |N|   Reserved  | EID mask-len  |        EID-Prefix-AFI         |
   Rec +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     \ |                       EID-Prefix  ...                         |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                   Map-Reply Record  ...                       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                               |
       +                                                               +
       |                                                               |
       +                            xTR-ID                             +
       |                                                               |
       +                                                               +
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Map-Request with N-bit and xTR-ID

The meaning of the fields is exactly the same as defined in [I-D.ietf-lisp-rfc6833bis]. The only addition is a flag bit in the EID-Record field. The meaning of this flag bit is as follows:

Notification-Requested bit (N-bit): the first bit in the EID-Record section of a Map-Request message. The N-bit of an EID-record is set to 1 to specify that the xTR wants to be notified of updates for that mapping record.

The PubSub functionality requires to include an xTR-ID in the Map-Request. This is done by setting the xTR-ID bit (I-bit) defined in [I-D.ietf-lisp-rfc6833bis]. When the I-bit of a Map-Request message is set, a 128-bit xTR-ID field is appended to the end of the Map-Request, immediately following the last EID-Record (or the Map-Reply Record, if present). The xTR-ID field uniquely identifies each xTR of a given LISP deployment. Provisioning of unique xTR-IDs is out of the scope of this document.

5. Mapping Request Subscribe Procedures

The xTR subscribes for RLOC-set changes for a given EID-prefix by sending a Map-Request to the Mapping System with the N-bit set on the EID-Record. The xTR builds a Map-Request according to [RFC6830] but also does the following:

(1) The xTR MUST set the I-bit of the Map-Request message to 1, to specify the presence of an xTR-ID field that uniquely identifies the xTR.

(2) The xTR MUST set the N-bit to 1 for each EID-Record to which the xTR wants to subscribe.

The Map-Request is forwarded to the appropriate Map-Server through the Mapping System. This document does not assume that a Map-Server is pre-assigned to handle the subscription state for a given xTR. The Map-Server that receives the Map-Request will be the Map-Server responsible to notify that specific xTR about future mapping changes for the subscribed mapping records.

Upon reception of the Map-Request, the Map-Server processes it as described in [RFC6830]. Upon processing, for each EID-Record that has the N-bit set to 1, the Map-Server proceeds adding the xTR-ID contained in the Map-Request to the list of xTR that have requested to be subscribed to that mapping record.

If the xTR-ID is added to the list, the Map-Server MUST send a Map-Notify message back to the xTR to acknowledge the successful subscription. The Map-Server MUST follow the specification in

Section 6.1.7 of [RFC6830] to build the Map-Notify with the following considerations.

(1) The Map-Server MUST use the nonce from the Map-Request as the nonce for the Map-Notify.

(2) The Map-Server MUST use its security association with the xTR (see Section 3) to compute the authentication data of the Map-Notify.

(3) The Map-Server MUST send the Map-Notify to one of the ITR-RLOCs received in the Map-Request.

When the xTR receives a Map-Notify with a nonce that matches one in the list of outstanding Map-Request messages sent with an N-bit set, it knows that the Map-Notify is to acknowledge a successful subscription. The xTR processes this Map-Notify as described in [RFC6830] with the following considerations. The xTR MUST use its security association with the Map-Server (see Section 3) to validate the authentication data on the Map-Notify. The xTR MUST use the Map-Notify to populate its map-cache with the returned EID-prefix and RLOC-set.

The subscription of an xTR-ID to the list of subscribers for the EID-Record may fail for a number of reasons. For example, because of local configuration policies (such as white/black lists of subscribers), or because the Map-Server has exhausted the resources to dedicate to the subscription of that EID-Record (e.g., the number of subscribers excess the capacity of the Map-Server).

If the subscription fails, the Map-Server MUST send a Map-Reply to the originator of the Map-Request, as described in [RFC6830]. This is also the case when the Map-Server does not support PubSub operation. The xTR processes the Map-Reply as specified in [RFC6830].

If an xTR-ID is successfully added to the list of subscribers for an EID-Record, the Map-Server MUST extract the ITR-RLOCs present in the Map-Request, and store the association between the xTR-ID and those RLOCs. Any already present state regarding ITR-RLOCs for the same xTR-ID MUST be overwritten.

If the Map-Request only has one ITR-RLOC with AFI = 0 (i.e. Unknown Address), the Map-Server MUST remove the subscription state for that xTR-ID. In this case, the Map-Server MUST send the Map-Notify to the source RLOC of the Map-Request. When the TTL for the EID-record expires, the EID-prefix is removed from the Map-Server’s subscription cache. On EID-Record removal, the Map-Server notifies the subscribers via a Map-Notify with TTL equal 0.

6. Mapping Notification Publish Procedures

The publish procedure is implemented via Map-Notify messages that the Map-Server sends to xTRs. The xTRs acknowledge the reception of Map-Notifies via sending Map-Notify-Ack messages back to the Map-Server. The complete mechanism works as follows.

When a mapping stored in a Map-Server is updated (e.g. via a Map-Register from an ETR), the Map-Server MUST notify the subscribers of that mapping via sending Map-Notify messages with the most updated mapping information. The Map-Notify message sent to each of the subscribers as a result of an update event MUST follow the exact encoding and logic defined in [RFC6830] for Map-Notify, except for the following:

(1) The Map-Notify MUST be sent to one of the ITR-RLOCs associated with the xTR-ID of the subscriber.

(2) The nonce of the Map-Notify MUST be randomly generated by the Map-Server.

(3) The Map-Server MUST use its security association with the xTR to compute the authentication data of the Map-Notify.

When the xTR receives a Map-Notify with a nonce not present in any list of previously sent nonces, and an EID not local to the xTR, the xTR knows that the Map-Notify has been received due to an update on the RLOC-set of a cached mapping.

The xTR processes the received Map-Notify as specified in [RFC6830], with the following considerations. The xTR MUST use its security association with the Map-Server (see Section 3) to validate the authentication data on the Map-Notify. The xTR MUST use the mapping information carried in the Map-Notify to update its internal map-cache. The xTR MUST acknowledge the Map-Notify by sending back a Map-Notify-Ack (specified in [I-D.ietf-lisp-rfc6833bis]), with the nonce from the Map-Notify, to the Map-Server. If after a configurable timeout, the Map-Server has not received back the Map-Notify-Ack, it CAN try to send the Map-Notify to a different ITR-RLOC for that xTR-ID.

7. Security Considerations

The way to provide a security association between the ITRs and the Map-Servers must be evaluated according to the size of the deployment. For small deployments, it is possible to have a shared key (or set of keys) between the ITRs and the Map-Servers. For larger and Internet-scale deployments, scalability is a concern and further study is needed.

8. Acknowledgments

This work is partly funded by the ANR LISP-Lab project #ANR-13-INFR-009 (https://lisplab.lip6.fr).

9. IANA Considerations

This document makes no request to IANA.

10. Normative References

[I-D.ietf-lisp-rfc6833bis]
Fuller, V., Farinacci, D., and A. Cabellos-Aparicio, "Locator/ID Separation Protocol (LISP) Control-Plane", draft-ietf-lisp-rfc6833bis-06 (work in progress), October 2017.

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

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

Authors’ Addresses

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

Email: natal@cisco.com

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

Email: vermagan@cisco.com

Johnson Leong
Cisco Systems
170 Tasman Drive
San Jose, CA
USA

Email: joleong@cisco.com

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

Email: fmaino@cisco.com

Albert Cabellos-Aparicio
Technical University of Catalonia
Barcelona
Spain

Email: acabello@ac.upc.edu

Sharon Barkai
Fermi Serverless
CA
USA

Email: sharon@fermicloud.io

Dino Farinacci
lispers.net
San Jose, CA
USA

Email: farinacci@gmail.com

Mohamed Boucadair
Orange
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Christian Jacquenet
Orange
Rennes 35000
France

Email: christian.jacquenet@orange.com

Stefano Secci
LIP6 UPMC
France

Email: stefano.secci@lip6.fr

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 4, 2018

F. Templin, Ed.
G. Saccone
Boeing Research & Technology
G. Dawra
LinkedIn
A. Lindem
Cisco Systems, Inc.
March 03, 2018

A Simple BGP-based Mobile Routing System for the Aeronautical Telecommunications Network
draft-templin-atn-bgp-06.txt

Abstract

The International Civil Aviation Organization (ICAO) is investigating mobile routing solutions for a worldwide Aeronautical Telecommunications Network with Internet Protocol Services (ATN/IPS). The ATN/IPS will eventually replace existing communication services with an IPv6-based service supporting pervasive Air Traffic Management (ATM) for Air Traffic Controllers (ATC), Airline Operations Controllers (AOC), and all commercial aircraft worldwide. This informational document describes a simple and extensible mobile routing service based on industry-standard BGP to address the ATN/IPS requirements.

Copyright Notice

Table of Contents

1. Introduction

The worldwide Air Traffic Management (ATM) system today uses a service known as Aeronautical Telecommunications Network based on Open Systems Interconnection (ATN/OSI). The service is used to augment controller to pilot voice communications with rudimenatary short text command and control messages. The service has seen successful deployment in a limited set of worldwide ATM domains.

The International Civil Aviation Organization [ICAO] is now undertaking the development of a next-generation replacement for ATN/OSI known as Aeronautical Telecommunications Network with Internet Protocol Services (ATN/IPS). ATN/IPS will eventually provide an IPv6-based service supporting pervasive ATM for Air Traffic Controllers (ATC), Airline Operations Controllers (AOC), and all commercial aircraft worldwide. As part of the ATN/IPS undertaking, a new mobile routing service will be needed. This document presents a candidate approach based on the Border Gateway Protocol (BGP) [RFC4271].

Aircraft communicate via wireless aviation data links that typically support much lower data rates than terrestrial wireless and wired-line communications. For example, some Very High Frequency (VHF)-based data links only support data rates on the order of 32Kbps and an emerging L-Band data link that is expected to play a key role in future aeronautical communications only supports rates on the order of 1Mbps. Although satellite data links can provide much higher data rates during optimal conditions, like any other aviation data link they are subject to errors, delay, disruption, signal intermittence, degradation due to atmospheric conditions, etc. The well-connected ground domain ATN/IPS network should therefore treat each safety-of-flight critical packet produced by (or destined to) an aircraft as a precious commodity and strive for an optimized Traffic Engineering service that provides the highest possible degree of reliability.

The ATN/IPS is an IPv6-based [RFC8200] overlay network that assumes a worldwide connected Internetworking underlay for carrying tunneled ATM communications. The Internetworking underlay could be manifested as a private collection of long-haul backbone links (e.g., fiberoptics, copper, SATCOM, etc.) interconnected by high-performance networking gear such as bridges, switches, and routers. Such a private network would need to connect all ATN/IPS participants worldwide, and could therefore present a considerable cost for a large-scale deployment of new infrastructure. Alternatively, the ATN/IPS could be deployed as a secured overlay over the existing global public Internet. For example, ATN/IPS nodes could be deployed as part of an SD-WAN or an MPLS-WAN that rides over the public Internet via secured tunnels.

The ATN/IPS further assumes that each aircraft will receive an IPv6 Mobile Network Prefix (MNP) that accompanies the aircraft wherever it travels. ATCs and AOCs will likewise receive IPv6 prefixes, but they would typically appear in static (not mobile) deployments such as air traffic control towers, airline headquarters, etc. Throughout the rest of this document, we therefore use the term "MNP" when discussing an IPv6 prefix that is delegated to any ATN/IPS end system, including ATCs, AOCs, and aircraft. We also use the term Mobility Service Prefix (MSP) to refer to an aggregated prefix assigned to the ATN/IPS by an Internet assigned numbers authority, and from which all MNPs are delegated (e.g., up to 2**32 IPv6 /64 MNPs could be delegated from the MSP 2001:db8::/32).

Connexion By Boeing [CBB] was an early aviation mobile routing service based on dynamic updates in the global public Internet BGP routing system. Practical experience with the approach has shown that frequent injections and withdrawals of MNPs in the Internet routing system can result in excessive BGP update messaging, slow routing table convergence times, and extended outages when no route is available. This is due to both conservative default BGP protocol timing parameters (see Section 6) and the complex peering interconnections of BGP routers within the global Internet infrastructure. The situation is further exacerbated by frequent aircraft mobility events that each result in BGP updates that must be propagated to all BGP routers in the Internet that carry a full routing table.

We therefore consider an approach using a BGP overlay network routing system where a private BGP routing protocol instance is maintained between ATN/IPS Autonomous System (AS) Border Routers (ASBRs). The private BGP instance does not interact with the native BGP routing system in the connected Internetworking underlay, and BGP updates are unidirectional from "stub" ASBRs (s-ASBRs) to a very small set of "core" ASBRs (c-ASBRs) in a hub-and-spokes topology. The Asymmetric Extended Route Optimization (AERO) architecture [I-D.templin-aerolink] is used to support mobility and route optimization services, where the BGP s-ASBRs are one and the same as AERO Servers and the BGP c-ASBRs are one and the same as AERO Relays. No extensions to the BGP protocol are necessary.

The s-ASBRs for each stub AS connect to a small number of c-ASBRs via dedicated high speed links and/or tunnels across the Internetworking underlay using industry-standard encapsulations (e.g., Generic Routing Encapsulation (GRE) [RFC2784], IPsec [RFC4301], etc.). The s-ASBRs engage in external BGP (eBGP) peerings with their respective c-ASBRs, and only maintain routing table entries for the MNPs currently active within the stub AS. The s-ASBRs send BGP updates for MNP injections or withdrawals to c-ASBRs but do not receive any BGP updates from c-ASBRs. Instead, the s-ASBRs maintain default routes with their c-ASBRs as the next hop, and therefore hold only partial topology information.

The c-ASBRs connect to other c-ASBRs using iBGP peerings over which they collaboratively maintain a full routing table for all active MNPs currently in service. Therefore, only the c-ASBRs maintain a full BGP routing table and never send any BGP updates to s-ASBRs. This simple routing model therefore greatly reduces the number of BGP updates that need to be synchronized among peers, and the number is reduced further still when localized mobility events within stub ASes (i.e., "intra-domain" mobility events) are processed within the AS instead of being propagated to the core. BGP Route Reflectors (RRs) [RFC4456] can also be used to support increased scaling properties.

The remainder of this document discusses the proposed BGP-based ATN/IPS mobile routing service.

2. Terminology

The terms Autonomous System (AS) and Autonomous System Border Router (ASBR) are the same as defined in [RFC4271].

The terms "AERO Client", "AERO Proxy", "AERO Server", and "AERO Relay" are the same as defined in [I-D.templin-aerolink].

The following terms are defined for the purposes of this document:

Air Traffic Managemnet (ATM)
The worldwide service for coordinating safe aviation operations.

Air Traffic Controller (ATC)
A government agent responsible for coordinating with aircraft within a defined operational region via voice and/or data Command and Control messaging.

Airline Operations Controller (AOC)
An airline agent responsible for tracking and coordinating with aircraft within their fleet.

Aeronautical Telecommunications Network with Internet Protocol Services (ATN/IPS)
A future aviation network for ATCs and AOCs to coordinate with all aircraft operating worldwide. The ATN/IPS will be an IPv6-based overlay network service that connects access networks via tunneling over an Internetworking underlay.

Internetworking underlay A connected wide-area network that supports overlay network tunneling and connects Radio Access Networks to the rest of the ATN/IPS.

Radio Access Network (RAN)
An aviation radio data link service provider’s network, including radio transmitters and receivers as well as suppporting ground-domain infrastructure needed to convey a customer’s data packets to the outside world. The term RAN is intended in the same spirit as for cellular operator networks and other radio-based Internet service provider networks. For simplicity, we also use the term RAN to refer to ground-domain networks that connect AOCs and ATCs without any aviation radio communications.

Core Autonomous System Border Router (c-ASBR) A BGP router located in the hub of a hub-and-spokes overlay network topology. Each c-ASBR is also an AERO Relay.

Stub Autonomous System Border Router (s-ASBR) A BGP router configured as a spoke in a hub-and-spokes overlay network topology. Each s-ASBR is also an AERO Server.

Client An ATC, AOC or aircraft that connects to the ATN/IPS as a leaf node. The Client could be a singleton host, or a router that connects a mobile network.

Proxy A node at the edge of a RAN that acts as a proxy go-between between Clients and Servers.

Mobile Network Prefix (MNP) An IPv6 prefix that is delegated to any ATN/IPS end system, including ATCs, AOCs, and aircraft.

Mobility Service Prefix (MSP) An aggregated prefix assigned to the ATN/IPS by an Internet assigned numbers authority, and from which all MNPs are delegated (e.g., up to 2**32 IPv6 /64 MNPs could be delegated from the MSP 2001:db8::/32).

3. ATN/IPS Routing System

The proposed ATN/IPS routing system comprises a private BGP instance coordinated between ASBRs in an overlay network via tunnels over the Internetworking underlay (where the tunnels between neighboring ASBRs are set up as part of the BGP peering configuration.) The overlay does not interact with the native BGP routing system in the connected undelying Internetwork, and each c-ASBR advertises only a small and unchanging set of MSPs into the Internetworking underlay routing system instead of the full dynamically changing set of MNPs. (For example, when the Internetworking underlay is the global public Internet the c-ASBRs advertise the MSPs in the public BGP Internet routing system.) The routing system is discussed in detail in [I-D.templin-aerolink].

In a reference deployment, one or more s-ASBRs connect each stub AS to the overlay using a shared stub AS Number (ASN). Each s-ASBR further uses eBGP to peer with one or more c-ASBRs. All c-ASBRs are members of the same core AS, and use a shared core ASN. Since the private BGP instance is separate from the global public Internet BGP routing system, the ASBRs can use either a private ASN per [RFC6996] or simply use public ASNs noting that the ASNs may overlap with those already assigned in the Internet. (A third alternative would be to procure globally-unique public ASNs, but cost and maintenance requirements must be conisdered.)

The c-ASBRs use iBGP to maintain a synchronized consistent view of all active MNPs currently in service. Figure 1 below represents the reference deployment. (Note that the figure shows details for only two s-ASBRs (s-ASBR1 and s-ASBR2) due to space constraints, but the other s-ASBRs should be understood to have similar Stub AS and MNP arrangements.) The solution described in this document is flexible enough to extend to these topologies.

```
     ...........................................................
   .                                                             .
   .               (:::)-.  <- Stub ASes ->  (:::)-.             .
   .   MNPs-> .-(:::::::::)             .-(:::::::::) <-MNPs     .
   .            ‘-(::::)-’                ‘-(::::)-’             .
   .             +-------+                +-------+              .
   .             |s-ASBR1|                |s-ASBR2|              .
   .             +--+----+                +-----+-+              .
   .                 \                         /                 .
   .                  \eBGP                   /eBGP              .
   .                   \                     /                   .
   .                    +-------+   +-------+                    .
   .          eBGP+-----+c-ASBR1|   +c-ASBR2+-----+eBGP          .
   .   +-------+ /      +--+----+   +-----+-+      \ +-------+   .
   .   |s-ASBRn+/       iBGP\   (:::)-.  /iBGP      \+s-ASBR3|   .
   .   +-------+            .-(::::::::)             +-------+   .
   .       .            .-(::::::::::::::)-.                     .
   .       .           (::::  Core AS   :::)                     .
   .   +-------+         ‘-(::::::::::::::)-’        +-------+   .
   .   |s-ASBR7+\      iBGP/‘-(:::::::-’\iBGP       /+s-ASBR4|   .
   .   +-------+ \      +-+-----+   +----+--+      / +-------+   .
   .          eBGP+-----+c-ASBRn|   |c-ASBR3+-----+eBGP          .
   .                    +-------+   +-------+                    .
   .                   /                     \                   .
   .                  /eBGP                   \eBGP              .
   .                 /                         \                 .
   .            +---+---+                 +-----+-+              .
   .            |s-ASBR6|                 |s-ASBR5|              .
   .            +-------+                 +-------+              .
   .                                                             .
   .                                                             .
   .   <------------ Internetworking Underlay  -------------->   .
    ............................................................
```

Figure 1: Reference Deployment

In the reference deployment, each s-ASBR maintains routes for active MNPs that currently belong to its stub AS. In response to "Inter-domain" mobility events, each S-ASBR will dynamically announces new MNPs and withdraws departed MNPs in its eBGP updates to c-ASBRs.

Since ATN/IPS end systems are expected to remain within the same stub AS for extended timeframes, however, intra-domain mobility events (such as an aircraft handing off between cell towers) are handled within the stub AS instead of being propagated as inter-domain eBGP updates.

Each c-ASBR configures a black-hole route for each of its MSPs. By black-holing the MSPs, the c-ASBR will maintain forwarding table entries only for the MNPs that are currently active, and packets destined to all other MNPs will correctly incur ICMPv6 Destination Unreachable messages [RFC4443] due to the black hole route. (This is the same behavior as for ordinary BGP routers in the Internet when they receive packets for which there is no route available.) The c-ASBRs do not send eBGP updates for MNPs to s-ASBRs, but instead originate a default route. In this way, s-ASBRs have only partial topology knowledge (i.e., they know only about the active MNPs currently within their stub ASes) and they forward all other packets to c-ASBRs which have full topology knowledge.

Scaling properties of this ATN/IPS routing system are limited by the number of BGP routes that can be carried by the c-ASBRs. A 2015 study showed that BGP routers in the global public Internet at that time carried more than 500K routes with linear growth and no signs of router resource exhaustion [BGP]. A more recent network emulation study also showed that a single c-ASBR can accommodate at least 1M dynamically changing BGP routes even on a lightweight virtual machine. Commercially-available high-performance dedicated router hardware can support many millions of routes.

Therefore, assuming each c-ASBR can carry 1M or more routes, this means that at least 1M ATN/IPS end system MNPs can be serviced by a single set of c-ASBRs and that number could be furhter increased by using RRs. Another means of increasing scale would be to assign a different set of c-ASBRs for each set of MSPs. In that case, each s-ASBR still peers with one or more c-ASBRs from each set of c-ASBRs, but the s-ASBR institutes route filters so that it only sends BGP updates to the specific set of c-ASBRs that aggregate the MSP. For example, if the MSP for the ATN/IPS deployment is 2001:db8::/32, a first set of c-ASBRs could service the MSP segment 2001:db8::/40, a second set could service 2001:db8:0100::/40, a third set could service 2001:db8:0200::/40, etc.

In this way, each set of c-ASBRs services a specific set of MSPs that they inject into the Internetworking underlay native routing system, and each s-ASBR configures MSP-specific routes that list the correct set of c-ASBRs as next hops. This BGP routing design also allows for natural incremental deployment, and can support initial small-scale deployments followed by dynamic deployment of additional ATN/IPS infrastructure elements without disturbing the already-deployed base.

4. ATN/IPS Multilink and Mobility Service

ATN/IPS end system multilink and mobility services are based on the AERO architecture [I-D.templin-aerolink], where end systems connect to aviation data link service provider Radio Access Networks (RANs). ATN/IPS end systems such as aircraft act as AERO Clients and may connect to multiple RANs at once, for example, when they have both a satellite link and an L-Band link activated simultaneously. Clients register all of their active data link connections with one or more AERO Servers which also act as s-ASBRs as discussed in Section 3. Clients may connect to Servers either directly, or via an AERO Proxy at the edge of the RAN. The Proxy function corresponds to the manner in which web proxies communicate with web servers on behalf of clients in secured domains such as corporate enterprise networks.

Figure 2 shows the ATN/IPS multilink and mobility model where Clients connect to RANs via aviation data links. Clients register their RAN addresses with a nearby Server, where the registration process may be brokered by a Proxy at the edge of the RAN.

```
            Data Link "A"     +--------+  Data Link "B"
                 +----------- | Client |-----------+
                /             +--------+            \
               /                                     \
              /                                       \
           (:::)-.                                  (:::)-.
      .-(:::::::::) <- Radio Access Networks -> .-(:::::::::)
        `-(::::)-'                                `-(::::)-'
         +-------+                                +-------+
    ...  | Proxy |  ...........................   | Proxy |  ...
   .     +-------+                                +-------+     .
   .                                                            .
   .                                                            .
   .                        +--------+              (:::)-.     .
   .                        | Server | eBGP <-> .-(:::::::::)   .
   .                        |(s-ASBR)|            `-(::::)-'    .
   .                        +--------+     ATN/IPS BGP Overlay  .
   .                                                            .
   .                                                            .
   .   <------------- Internetworking Underlay ------------->   .
    ............................................................
```

Figure 2: ATN/IPS Multilink and Mobility Architecture

In this model, when a Client logs into a RAN it specifies a nearby Server (s-ASBR) that it has selected to connect to the ATN/IPS. The login process is brokered by a Proxy at the border of the RAN, which then conveys the connection request to the Server via tunneling across the Internetworking underlay. The Server then registers the address of the Proxy as the address for the Client, and the Proxy forwards the Server's reply to the Client. If the Client connects to multiple RANs, the Server will register the addresses of all Proxies along with their Quality of Service (QoS) preferences as addresses through which the Client can be reached.

Once the Client has registered its data link addresses with the Server via one or more Proxies, the Proxies can signal fine-grained events like QoS changes to the Server on behalf of the Clients. For example, if a data link signal is fading, the Proxy can inform the Server without involvement of the Client. Moreover, if the RAN supports intra-domain route injection, the Client can avoid encapsulation and send and receive all of its packets unencapsulated since the RAN will natively route them to and from the Proxy. The Proxy will then tunnel the packets to and from the Server across the Internetworking underlay so that the Client need not incur any over-the-air encapsulation on performance-constrained aviation data links.

The Server represents all of its active Clients as MNP routes in the ATN/IPS BGP routing system. The Server's stub AS therefore consists of the set of all of its active Clients. The Server injects the MNPs of its active Clients and withdraws the MNPs of its departed Clients via BGP updates to c-ASBRs. Since Clients are expected to remain associated with their current Servers for extended periods, the level of MNP injections and withdrawals in the BGP routing system will be on the order of the numbers of network joins, leaves and Server handovers for aircraft operations (see: Section 6). It is important to observe that fine-grained events such as Client mobility and QoS signaling are coordinated only by Proxies and Servers, and do not involve other ASBRs in the routing system. In this way, localized events are not propagated into the global BGP routing system.

5. ATN/IPS Route Optimization

ATN/IPS end systems will frequently need to communicate with correspondents associated with other s-ASBRs. In the ASBR peering topology discussed in Section 3, this can initially only be accommodated by including multiple ASBRs-to-ASBR tunnel segments in the forwarding path. In many cases, it would be desirable to eliminate extraneous ASBR tunnel segments from this "dogleg" route so that packets can traverse a minimum number of tunneling hops across the Internetworking underlay using the AERO route optimization service [I-D.templin-aerolink].

A route optimization example is shown in Figure 3 and Figure 4 below. In the first figure, packets sent from Client1 to Client2 are transmitted across the source RAN to Proxy1 without encapsulation. Proxy1 then tunnels the packets to Server 1 (s-ASBR1), which tunnels them to Relay 1 (c-ASBR1), which tunnels them to Relay2 (c-ASBR2), which tunnels them to Server2 (s-ASBR2), which finally tunnels them to Proxy2. In the second figure, the optimized route tunnels packets directly from Proxy1 to Proxy2 without involving the ASBRs.

```
         +---------+                             +---------+
         | Client1 |                             | Client2 |
         +---v-----+                             +-----^---+
             *                                         *
             *                                         *
           (:::)-.                                   (:::)-.
      .-(:::::::::)  <- Radio Access Networks ->  .-(:::::::::)
        '-(::::)-'                                 '-(::::)-'
         +--------+                              +--------+
    ...  | Proxy1 |  .........................   | Proxy2 |  ...
   .     +--------+                              +--------+     .
   .             **                               **            .
   .              **                             **             .
   .               **                           **              .
   .           +---------+                 +---------+          .
   .           | Server1 |                 | Server2 |          .
   .           |(s-ASBR1)|                 |(s-ASBR2)|          .
   .           +--+------+                 +-----+---+          .
   .                 \  **      Dogleg      **   /              .
   .              eBGP\  **     Route      **   /eBGP           .
   .                   \  **==============**   /                .
   .                   +---------+   +---------+                .
   .                   | Relay1  |   | Relay2  |                .
   .                   |(c-ASBR1)|   |(c-ASBR2)|                .
   .                   +---+-----+   +----+----+                .
   .                       +--------------+                     .
   .                             iBGP                           .
   .                                                            .
   .   <------------- Internetworking Underlay ------------->   .
    ............................................................
```

Figure 3: Dogleg Route Before Optimization

```
         +---------+                             +---------+
         | Client1 |                             | Client2 |
         +---v-----+                             +-----^---+
             *                                         *
             *                                         *
           (:::)-.                                   (:::)-.
      .-(::::::::)  <- Radio Access Networks -> .-(::::::::)
        `-(::::)-'                                `-(::::)-'
         +--------+                              +--------+
    ...  | Proxy1 |  .........................  | Proxy2 |  ...
   .     +------v-+                              +--^-----+     .
   .             *                                  *           .
   .              *=================================*           .
   .                                                            .
   .           +---------+                  +---------+         .
   .           | Server1 |                  | Server2 |         .
   .           |(s-ASBR1)|                  |(s-ASBR2)|         .
   .           +--+------+                  +-----+---+         .
   .                 \                           /              .
   .              eBGP\                         /eBGP           .
   .                   \                       /                .
   .                   +---------+   +---------+                .
   .                   | Relay1  |   | Relay2  |                .
   .                   |(c-ASBR1)|   |(c-ASBR2)|                .
   .                   +---+-----+   +----+----+                .
   .                       +--------------+                     .
   .                             iBGP                           .
   .                                                            .
   .   <------------- Internetworking Underlay ------------->   .
    ............................................................
```

Figure 4: Optimized Route

The route optimization is accommodated by control message signaling between the Proxies and ASBRs. When the Proxy nearest the source sends a route optimization request, the request is forwarded toward the Server and nearest the destination. If the request is authentic, the destination Server provides the source Proxy with the address of the destination Proxy so that unnecessary tunnel segments are eliminated and direct Proxy-to-Proxy tunneling is enabled. At the same time, the destination Server keeps track of the source Proxies it has sent route optimization messages to so it can quickly update them if network mobility or Quality of Service (QoS) conditions change.

Note that route optimization can fail if Proxy1 cannot tunnel packets directly to Proxy2 due to some form of blockage in the Internetworking underlay such as filtering middle-boxes. It is also necessary for Proxy1 to detect and adjust to failure of Proxy2 through receipt of a Server's IPv6 Neighbor Advertisement message and/or Neighbor Unreachability Detection (NUD) [RFC4861]. Note also that the Servers still maintain state so they can echo link QoS update messages coming from the RANs to inform correspondents of QoS changes (e.g., a link signal strength fading, a data link connection loss, etc.).

Finally, each s-ASBR always has a default route and can therefore always send packets via the dogleg route through a c-ASBR even if a route optimized path has been established. The direct paths between s-ASBRs and c-ASBRs are tunnels are maintained by BGP peering session keepalives such that, if a link or an ASBR goes down, BGP will detect the failure and readjust the routing tables. However, ASBRs and the links that interconnect them are expected to be secured as highly-available and fault tolerant critical infrastructure such that peering session failures should be extremely rare.

6. BGP Protocol Considerations

The number of eBGP peering sessions that each c-ASBR must service is proportional to the number of s-ASBRs in the system. Network emulations with lightweight virtual machines have shown that a single c-ASBR can service at least 100 eBGP peerings from s-ASBRs that each advertise 10K MNP routes (i.e., 1M total). It is expected that robust c-ASBRs can service many more peerings than this - possibly by multiple orders of magnitude. But even assuming a conservative limit, the number of s-ASBRs could be increased by also increasing the number of c-ASBRs. Since c-ASBRs also peer with each other using iBGP, however, larger-scale c-ASBR deployments may need to employ an adjunct facility such as BGP Route Reflectors (RRs)[RFC4456].

The number of aircraft in operation at a given time worldwide is likely to be significantly less than 1M, but we will assume this number for a worst-case analysis. Assuming a worst-case average 1 hour flight profile from gate-to-gate with 10 Server transitions per flight, the entire system will need to service at most 10M BGP updates per hour (2778 updates per second). This number is within the realm of the peak BGP update messaging seen in the global public Internet today [BGP2]. Assuming a BGP update message size of 100 bytes (800bits), the total amount of BGP control message traffic to a single c-ASBR will be less than 2.5Mbps which is a nominal rate for modern data links.

Industry standard BGP routers provide configurable parameters with conservative default values. For example, the default hold time is 90 seconds, the default keepalive time is 1/3 of the hold time, and the default MinRouteAdvertisementinterval is 30 seconds for eBGP peers and 5 seconds for iBGP peers (see Section 10 of [RFC4271]). For the simple mobile routing system described herein, these parameters can and should be set to more aggressive values to support faster neighbor/link failure detection and faster routing protocol convergence times. For example, a hold time of 3 seconds and a MinRouteAdvertisementinterval of 0 seconds for both iBGP and eBGP.

C-ASBRs will be using EBGP both in the ATN/IPS and the Internetworking Underlay with the ATN/IPS unicast IPv6 routes resolving over Internetworking Underlay routes. Consequently, c-ASBRs and potentially s-ASBRs will need to support separate local ASes for the two BGP routing domains and routing policy or assure routes are not propagated between the two BGP routing domains. From a conceptual and operational standpoint, the implementation should provide isolation between the two BGP routing domains (e.g., separate BGP instances).

7. Implementation Status

The BGP routing topology described in this document has been modeled in realistic network emulations showing that at least 1 million MNPs can be propagated to each c-ASBR even on lightweight virtual machines. No BGP routing protocol extensions need to be adopted.

8. IANA Considerations

This document does not introduce any IANA considerations.

9. Security Considerations

ATN/IPS ASBRs on the open Internet are susceptible to the same attack profiles as for any Internet nodes. For this reason, ASBRs should employ physical security and/or IP securing mechanisms such as IPsec [RFC4301], TLS [RFC5246], etc.

ATN/IPS ASBRs present targets for Distributed Denial of Service (DDoS) attacks. This concern is no different than for any node on the open Internet, where attackers could send spoofed packets to the node at high data rates. This can be mitigated by connecting ATN/IPS ASBRs over dedicated links with no connections to the Internet and/or when ASBR connections to the Internet are only permitted through well-managed firewalls.

ATN/IPS s-ASBRs should institute rate limits to protect low data rate aviation data links from receiving DDoS packet floods.

This document does not include any new specific requirements for mitigation of DDoS.

10. Acknowledgements

This work is aligned with the FAA as per the SE2025 contract number DTFAWA-15-D-00030.

This work is aligned with the NASA Safe Autonomous Systems Operation (SASO) program under NASA contract number NNA16BD84C.

This work is aligned with the Boeing Information Technology (BIT) MobileNet program.

11. References

11.1. Normative References

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/info/rfc4443>.

[RFC4451] McPherson, D. and V. Gill, "BGP MULTI_EXIT_DISC (MED) Considerations", RFC 4451, DOI 10.17487/RFC4451, March 2006, <https://www.rfc-editor.org/info/rfc4451>.

[RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, DOI 10.17487/RFC4456, April 2006, <https://www.rfc-editor.org/info/rfc4456>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

11.2. Informative References

[BGP] Huston, G., "BGP in 2015, http://potaroo.net", January 2016.

[BGP2] Huston, G., "BGP Instability Report, http://bgpupdates.potaroo.net/instability/bgpupd.html", May 2017.

[CBB] Dul, A., "Global IP Network Mobility using Border Gateway Protocol (BGP), http://www.quark.net/docs/Global_IP_Network_Mobility_using_BGP.pdf", March 2006.

[I-D.ietf-lisp-ddt] Fuller, V., Lewis, D., Ermagan, V., Jain, A., and A. Smirnov, "LISP Delegated Database Tree", draft-ietf-lisp-ddt-09 (work in progress), January 2017.

[I-D.templin-aerolink] Templin, F., "Asymmetric Extended Route Optimization (AERO)", draft-templin-aerolink-81 (work in progress), February 2018.

[ICAO] ICAO, I., "http://www.icao.int/Pages/default.aspx", February 2017.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <https://www.rfc-editor.org/info/rfc2784>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://www.rfc-editor.org/info/rfc5246>.

[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, <https://www.rfc-editor.org/info/rfc6836>.

[RFC6996] Mitchell, J., "Autonomous System (AS) Reservation for Private Use", BCP 6, RFC 6996, DOI 10.17487/RFC6996, July 2013, <https://www.rfc-editor.org/info/rfc6996>.

Authors' Addresses

Fred L. Templin (editor)
Boeing Research & Technology
P.O. Box 3707
Seattle, WA 98124
USA

Email: fltemplin@acm.org

Greg Saccone
Boeing Research & Technology
P.O. Box 3707
Seattle, WA 98124
USA

Email: gregory.t.saccone@boeing.com

Gaurav Dawra
LinkedIn
USA

Email: gdawra.ietf@gmail.com

Acee Lindem
Cisco Systems, Inc.
USA

Email: acee@cisco.com