LISP Working Group
Internet-Draft
Intended status: Informational
Expires: December 30, 2020

S. Barkai
B. Fernandez-Ruiz
S. ZionB
R. Tamir
Nexar Inc.
A. Rodriguez-Natal
F. Maino
Cisco Systems
A. Cabellos-Aparicio
J. Paillissé Vilanova
Technical University of Catalonia
D. Farinacci
lispers.net
July 30,2020

Network-Hexagons: H3-LISP GeoState & Mobility Network
draft-ietf-lisp-nexagon-04

Abstract

This document specifies use of H3 and LISP to publish subscribe and reflect the real-time state and status of public spaces and public roads:
- Tile by tile, indexed annotation of streets & curbs in near real time
- Sharing hazards, blockages, parking, weather, maintenance, inventory..
- Between MobilityClients who produce and consume geo-state information
- Using geo-spatial IP channels of current state of the physical world

Table of Contents

1. Introduction 2
2. Requirements Language 3
3. Definition of Terms 3
4. Deployment Assumptions 4
5. Mobility Clients Network Services 4
6. Mobility Unicast-Multicast 5
7. Security Considerations 6
8. Acknowledgments 6
9. IANA Considerations 6
10. Normative References 8
Authors' Addresses 9

1. Introduction

(1) 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 (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.

(2) H3 is a geospatial indexing system using a hexagonal grid that can be (approximately) subdivided into finer and finer hexagonal grids, combining the benefits of a hexagonal grid with hierarchical subdivisions. H3 supports sixteen resolutions. Each finer resolution has cells with one seventh the area of the coarser resolution. Hexagons cannot be perfectly subdivided into seven hexagons, so the finer cells are only approximately contained within a parent cell. Each cell is identified by a 64bit HID.

(3) The Berkeley Deep Drive (BDD) Industry Consortium investigates state-of-the-art technologies in computer vision and machine learning for automotive applications, and, for taxonomy of published automotive scene classification.

These standards are combined to create an in-network state which reflects the condition of each hexagonal tile (~1sqm) in every road. The lisp network maps & encapsulates traffic between MobilityClient endpoint identifiers (EID), and addressable tile-objects (HID=>EID). objects are aggregated by H3Service

EIDs.

The H3-LISP mobility network bridges timing and location gaps between the production and consumption of information by MobilityClients:

- o vision, sensory, LIADR, AI applications -- information producers
- o driving-apps, map-apps, command & control -- information consumers

This is achieved by putting the physical world on a shared addressable state-grid typically at the edge. Tile by tile geo-state sharing using a brokered-network solves key issues in vehicle to vehicle information sharing. Challenges such as vision sensors when there are multiple perspectives, privacy and cyber when clients are directly communicating when they do not have to, and global geo pub-sub scenarios.

Given a situation observable by some end-points, it is unclear if the relevant end-points which need to know will receive consistent, conflicting, multiple, or no indications whatsoever. For example, when a vehicle experiences a sudden highway slow-down,"sees" many brake lights or "feels" accelerometer, there is no clear way for it to share this annotation with vehicles 20-30sec away to prevent a potential pile-up. Or, when a vehicle crosses an intersection, observing opposite-lane obstruction, construction, double-park, commercial loading, garbage truck, or stopped school-bus, there is no clear way for it to alert vehicles approaching that situation as it drives away.

Geo-state indirection also helps communicate advanced machine vision and/or radar annotations. These are constantly evolving technologies, and relaying road enumerations they produce, using peer-to-peer protocols, poses a significant interoperability challenge. It is hard to test each new annotation of any sensor or OEM vendor with any other driving application. Brokered IP multicast channels are themed, subscribing means interoperating.

These peer-to-peer limitations are inherent yet unnecessary, as in most road situations vehicles are not really proper peers. They just happen to be in the same place at the same time. H3-LISP mobility network solves the limitations of direct vehicle-to-vehicle communication by anchor brokers per geo-tile: timing, security, privacy, interoperability. Anchor brokering is achieved by MobilityClients communicating through in-network addressable geo-states. Addressable tiles are aggregated and maintained by LISP H3ServiceEIDs.

MobilityClients can provide drivers with heads-up alerts on hazards and obstacles beyond the line of sight of the driver and the in-car sensors: over traffic, around blocks, far-side junction, beyond road turns or surface curvatures. This highlights the importance of networks in providing road safety and the role networks play in future AV operation support systems (AV-OSS).

Beyond sharing use cases like finding freed-up curb-parking and help avoid construction zones, a mission critical use case for global geo-pub-sub has to do with supporting autonomous vehicle (AV) fleets.

Its now a consensus that as specific AV fleets start to role out, or regular cars with AV abilities become more perversive, that they need remote operations and remote take-over support. This means that for every M such cars there needs to be N human remote drivers ready to take over. These AV-OSS are put together by consortiums of multiple companies and are measured by their N/M.

Nexagon geo-channels role in this context is two fold:

1. flag tiles where the AV got confused because of current condition which is requiring remote intervention, so to steer other AVs away from this tile. This so not to pull-in more and more humans to intervene as more cars arrive.
2. fleets will heavily rely on platoons, convoys assembled on the road on the fly, driven at the head by a remote operator or human driver. AVs need to lock-on. Geo channels are used pub-sub these platoons as they drive by.

To summarize the H3-LISP solution outline:

(1) MicroPartition: 64bit indexed geo-spatial H3.r15 road-tiles
(2) EnumState: 64bit state values compile tile condition representation
(3) Aggregation: H3.r9 H3ServiceEID group individual H3.r15 road-tiles
(4) Channels: H3ServiceEIDs function as multicast state update channels
(5) Scale: H3ServiceEIDs distributed for in-network for latency-throughput
(6) Mapped Overlay: tunneled-network routes the mobility-network traffic
(7) Signal-free: tunneled overlay is used to map-register for mcast channels
(8) Aggregation: tunnels used between MobilityClients/H3ServiceEIDs <> edge
(9) Access: ClientXTRs/ServerXTRs tunnel traffic to-from the LISP EdgeRTRs
(10) Control: EdgeRTRs register-resolve H3ServiceEIDs and mcast subscription

```
|-0-|-1-|-2-|-3-|-4-|-5-|-6-|-7-|-8-|-9-|-A-|-B-|-C-|-D-|-E-|-F-|
|                         H3 Hexagon ID Key                     |
|-0-|-1-|-2-|-3-|-4-|-5-|-6-|-7-|-8-|-9-|-A-|-B-|-C-|-D-|-E-|-F-|
|                       H3 Hexagon State-Value                  |
|---------------------------------------------------------------|
```

```
                        ___                                 ___
 H3ServiceEIDs   ___  /     \          H3ServiceEIDs ___  /     \
          ___  /     | H3.r9 |                  ___  /     | H3.r9 |
        /     | H3.r9 \ ___ /                 /     | H3.r9 \ ___ /
       | H3.r9 \ ___ /  sXTR                 | H3.r9 \ ___ /  sXTR
```

```
          \ ___ /  sXTR    |                   \ ___ /  sXTR     |
            sXTR    |      |                     sXTR     |      |
             |      |      |                      |       |      |
             |      |      |                      |       |      |
             + -  - + - - EdgeRTR           EdgeRTR - + - + - -  +
                             ||  (   (   ((  ||
                          (                        )
                        (      Network Hexagons      )
                      (            H3-LISP              )
                        (      Mobility Network       )
                          ((                        )
                            ||  ((   (())   ()  ||
                            ||                  ||
                = = = = = = =                     = = = = = = =
               ||                                             ||
            EdgeRTR                                         EdgeRTR
           ..    ..                                      ..      ..
          ..       ..                                  ..          ..
    ((((|))))    ((((|))))                         ((((|))))    ((((|))))
       /|\    RAN   /|\                               /|\    RAN   /|\
      ..                                                            ..
      ..                                                            ..
      ..          Road tiled by 1 sqm H3.r15 ID-Ed Geo-States       ..
      ..                                                            ..
      ..                        ___    ___    ___                   ..
      ..       .............  /     \/     \/     \ << cXTR::MobilityClientB
      ..       - - - - - - -  H3.r15  H3.r15 H3.r15 - - - - - - -
     MobilityClientA::cXTR >> \ ___ /\ ___ /\ ___ /.........
```

- MobilityClientA has seen MobilityClientB near future, and, vice versa
- Clients share information using addressable shared-state routed by LISP Edge
- ClientXTR (cXTR): tunnel encapsulation through access network to LISP Edge
- ServerXTR (sXTR): tunnel encapsulation through cloud network to LISP Edge
- The H3-LISP Mobility overlay starts in the cXTR and terminates in the sXTR
- The updates are routed to the appropriate tile geo-state by the LISP network
- EdgeRTRs perform multicast replication to edges and then native or to cXTRs
- Clients receive tile-by-tile geo-state updates via the multicast channels

Each H3.r9 hexagon is an EID Service with corresponding H3 hexagon ID. Bound to that service is a LISP xTR, called a ServerXTR, specified to deliver encapsulated packets to and from the H3ServiceEID and LISP Edge. EdgeRTRs are used to re-tunnel packets from MobilityClients to H3ServiceEIDs. Each H3ServiceEID is also a multicast source for updating MobilityClients on the state of the H3.r15 tiles aggregated by the H3ServiceEID.

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. Definition of Terms

H3ServiceEID: Is an addressable aggregation of H3.r15 state-tiles. It is a designated source for physical world reported annotations, and an (s,g) source of multicast public-safety update channels. H3ServiceEID is itself an H3 hexagon, large enough to provide geo-spatial conditions context, but not too large as to over-burden (battery powered, cellular connected) subscribers with too much information. For Mobility Network it is H3.r9. It has a light-weight LISP protocol stack to tunnel packets aka ServerXTR. The EID is an IPv6 EID that contains the H3 64-bit address numbering scheme. See IANA consideration for details.

ServerXTR: Is a light-weight LISP protocol stack implementation that co-exists with H3ServiceEID process. When the server roams, the xTR roams with it. The ServerXTR encapsulates and decapsulates packets to/from EdgeRTRs.

MobilityClient: Is a roaming application that may be resident as part of an automobile, as part of a navigation application, part of municipal, state, of federal government command and control application, or part of live street view consumer type of application. It has a light-weight LISP protocol stack to tunnel packets aka ClientXTR.

MobilityClient EID: Is the IPv6 EID used by the Mobility Client applications to source packets. The destination of such packets are only H3ServiceEIDs. The EID format is opaque and is assigned as part of the MobilityClient network-as-a-service (NaaS) authorization.

ClientXTR: Is the light-weight LISP protocol stack implementation that is co-located with the Mobility Client application. It encapsulates packets sourced by applications to EdgeRTRs and decapsulates packets from EdgeRTRs.

EdgeRTR: Is the core scale and structure of the LISP mobility network. EdgeRTRs proxy H3ServiceEIDs and MobilityClient H3ServiceEID channel registration. EdgeRTRs aggregate MobilityClients and H3Services using tunnels to facilitate hosting-providers and mobile-hosting flexibility - for accessing the nexagon mobility network.
EdgeRTRs decapsulate packets from ClientXTRs, ServerXTRs and re-encapsulates packets to the clients and servers tunnels. EdgeRTRs glean H3ServiceEIDs and glean MobilityClient EIDs when it decapsulates packets. EdgeRTRs store H3ServiceEIDs and their own RLOC of where the H3ServiceEID is currently reachable from in the map-cache. These mappings are registered to the LISP mapping system so other EdgeRTRs know where to encapsulate for such EIDs. EdgeRTRs do not register MobilityClients' EIDs at the mapping service as

these are temporary-renewed while using the mobility network. Enterprises may provide their own client facing EdgeRTRs to mask their clients geo-whereabouts while using the mobility network.

4. Deployment Assumptions

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

(1) Unique 64-bit HID is associated with each H3 geo-spatial tile
(2) MobilityClients and H3ServiceEIDs share this well known index
(3) 64-bit BDD state value is associated with each H3-indexed tile
(4) Tile state is compiled 16 fields of 4-bits, or max 16 enums

```
 |-0-|-1-|-2-|-3-|-4-|-5-|-6-|-7-|-8-|-9-|-A-|-B-|-C-|-D-|-E-|-F-|
  0123012301230123012301230123012301230123012301230123012301230123
```

Subscription of MobilityClients to the mobility network is constantly renewed while on the move and is not intended as a means of basic connectivity. This is why MobilityClients use DNS/AAA to obtain temporary EIDs and EdgeRTRs and why they use (LISP) data-plane tunnels to communicate using their temporary EIDs with the dynamically assigned EdgeRTRs.

MobilityClient are otherwise unaware of the LISP network mechanism or mapping system and simply regard the data-plane tunnels as an application-specific virtual private network (VPN) that supports IPv6 EID addressable geo-state to publish (Ucast), Subscribe (Mcast) H3Services.

In order to get access to the MobilityVPN, MobilityClients first authenticate with the MobilityVPN AAA Server. DIAMETER based AAA is typically done at the provider edge (PE) by edge gateways. However, the typical case involves several types of customer equipment connected by wireline, or by wireless to a specific service provider. The Mobility VPN, on the other hand, potentially overlays a number of wireless networks and cloud-edge providers. It also involves dozens of Car-OEM, Driving-Applications, Smart-infrastructure vendors. This is why we require clients to first go through AAA in order to get both a MobilityClientEID and EdgeRTR gateway RLOC opened.

ClientXTR performs the following steps in order to use the mobility network:
1) obtain the address of the mobility network AAA server using DNS
2) obtain MobilityClientEID and EdgeRTR(s) from AAA server using DIAMETER
3) renew authorization from AAA while using the mobility network T1 minutes

```
  MobilityClient   Domain Name Server    DIAMETER AAA        Mobility EdgeRTR
          |                    |                   |                   |
          | nslookup nexagon   |                   |                   |
          |------------------->|                   |                   |
          |<-------------------|                   |                   |
          |  Mobility AAA IP   |                   |                   |
          |                    |                   |                   |
          |  AAR(AVP:IMSI/User/Password/Toyota)    |                   |
          |--------------------------------------->|                   |
          |                    |                   | ACR(AVP ClientEID)|
          |                    |                   |------------------>|
          |                    |                   |<------------------|
          |                    |                   | ACA(AVP ClientEID)|
          |    AAA (Client::EID,EdgeRTR::RLOC)     |                   |
          |<---------------------------------------|                   |
          |                    |                   |                   |
          .                                                            .
          .                                                            .
          .                                                            .
          |   Publish IPv6 H3ServiceEID, Subscribe MLDv2 H3ServiceEID
|                                                     .
          |----------------------------------------------------------->|
          .                                                            .
          .                                                            .
          |<-----------------------------------------------------------|
          |       Signal freeing multicast Updates from H3ServiceEIDs  |
          .                                                            .
          .                                                            .
          .                                                            .
          |                    |                   |                   |
          |               AAR(Interim)             |                   |
          |--------------------------------------->|   ACR (Interim)   |
          |                    |                   |------------------>|
          |                    |                   |<------------------|
          |                    |                   |   ACA (Interim)   |
          |<---------------------------------------|                   |
          |               AAA (Interim)            |                   |
```

Using this network login and re-login method we ensure that:
- the MobilityClientEIDs serve as credentials with the specific EdgeRTRs
- EdgeRTRs are not tightly coupled to H3.r9 areas for privacy/load-balance
- Mobility Clients do not need to update EdgeRTRs while roaming in an area

The same EdgeRTR may serve several H3.r9 areas for smooth ride continuity, and, several EdgeRTRs may load balance an H3.r9 area with high density of originating MobilityClient rides. When a MobilityClient ClientXTR is homed to EdgeRTR, it is able to communicate with H3ServiceEIDs.

5. Mobility Clients Network Services

The mobility network functions as a standard LISP VPN overlay.
The overlay delivers unicast and multicast packets across:

- multiple access-networks and radio-access specifications
- multiple edie providers, public, private, and hybrid clouds

We use data-plane XTRs in the stack of each mobility client and server. ClientXTRs and ServerXTRs are homed to one or more EdgeRTRs at the LISP edge. This structure allows for MobilityClients to "show up" at any time, behind any network provider in a given mobility network administrative domain, and for any H3ServiceEID to be instantiated, moved, or failed-over to any rack in any cloud-provider. The LISP overlay enables these roaming mobility network elements to communicate uninterrupted. This quality is insured by the LISP RFCs. The determination of identities for MobilityClients to always refer to the correct H3ServiceEID is insured by H3 geo-spatial HIDs.

There are two options for how we associate ClientXTRs with LISP EdgeRTRs:

I. Semi-random load-balancing by DNS/AAA

In this option we assume that in a given metro edge, a pool of EdgeRTRs can distribute the Mobility Clients load randomly between them and that EdgeRTRs are topologically more or less equivalent. Each RTR uses LISP to tunnel traffic to and from other EdgeRTRs for MobilityClient with H3Service exchanges.
MobilityClients can (multi) home to EdgeRTRs while moving.

II. Topological by anycast

In this option we align an EdgeRTR with topological aggregation like in Evolved Packet or 5GCore aggregation. Mobility Clients are roaming in an area home to that RTR and so is the H3 Server. There is only one hop across the edge overlay between clients and servers and mcast replication is more focused, but clients need to keep re-homing as they move.

To summarize the H3LISP mobility network layout:

(1) Mobility-Clients traffic is tunneled via data-plane ClientXTRs
    ClientXTRs are (multi) homed to EdgeRTR(s)
(2) H3ServiceEID traffic is tunneled via data-plane ServerXTR
    ServerXTRs are (multi) homed to EdgeRTR(s)
(3) EdgeRTRs use mapping service to resolve Ucast HIDs to RTR RLOCs
    EdgeRTRs also register to (Source, Group) H3ServiceEID multicasts

```
MobilityClients <> ClientXTR <Access Provider > EdgeRTR  v
                                                         v
v      << Map-Assisted Mobility-Network Overlay <<       v
v
```

>> EdgeRTR <Cloud Provider> ServerXTR <> H3ServiceEID

6. Mobility Unicast and Multicast

Regardless of the way a given ClientXTR was associated with an EdgeRTR, an authenticated MobilityClient EID can send: [64bitH3.15ID :: 64bitState] annotations to the H3.r9 H3ServiceEID. The H3.r9 EID can be calculated by clients algorithmically from the H3.15 localized annotation snapped-to-tile.

The ClientXTR encapsulates MobilityClient EID and H3ServiceEID in a packet sourced from the ClientXTR with the destination of the EdgeRTR RLOC LISP port.
EdgeRTRs then re-encapsulate annotation packets either to remote EdgeRTR (option 1) or to homed H3ServiceEID ServerXTR (option 2).
The remote EdgeRTR aggregating H3ServiceEIDs re-encapsulates MobilityClient EID to the ServerXTR and from there to the H3ServiceEID.

```
    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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ \
|Version| Traffic Class |           Flow Label                  |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |
|         Payload Length        |  Next Header  |   Hop Limit   |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |
|                                                               |  |
+                                                               +  |
|                                                               |  |
+                     Source MobilityClientEID                  +  |
|                                                               | IPv6
+                                                               +  |
|                                                               |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |
|                                                               |  |
+                                                               +  |
|                                                               |  |
+                        Dest H3ServiceEID                      +  |
|                                                               |  |
+                                                               +  |
|                                                               | /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Source Port = xxxx      |       Dest Port = xxxx        | \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ UDP
|           UDP Length          |        UDP Checksum           | /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ \
|  Type         |gzip |        Reserved         | Pair Count = X| Nexgon Header
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ /
|                                                               |
+                        64 Bit H3-R15 ID                       +
|                                                               |
```

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                        64 Bit State                           +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                        64 Bit H3-R15 ID                       +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                        64 Bit State                           +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

To Summarize Unicast:

```
   (1) MobilityClients can send annotations which are localized to an H3.r15
tile
       These annotations are sent to an H3.r9 mobility H3ServiceEIDs
   (2) MobilityClient EID and H3ServiceEID HID are encapsulated:
         XTR <> RTR <> RTR <> XTR
       * RTRs can map-resolve re-tunnel HIDs
   (3) RTRs re-encapsulate original source-dest to ServerXTRs
       ServerXTRs decapsulate packets to H3ServiceEID
```

Each H3.r9 Server is also an IP Multicast channel Source used to update subscribers on the aggregate state of the H3.r15 tiles in the H3.r9 Server. This forms a multipoint to multipoint state channel per H3 geo-location, where the H3 hairpin aggregation point has programable propagation functionality.

We use rfc8378 signal-free multicast to implement mcast channels in the overlay. The mobility network has many channels, with only a few thousands of subscribers per channel. MobilityClients driving through or subscribing to an H3.r9 area can explicitly issue an rfc4604 MLDv2 in order to subscribe, or, may be subscribed implicitly by the EdgeRTR gleaning to ucast HID destination.

The advantage of an explicit client MLDv2 registration as a trigger to rfc8378 is that the clients manage their own mobility mcast handover according to their location-direction vectors, and that it allows for otherwise silent, or, non annotating clients. The advantage of EdgeRTR implicit registration is less signaling required.

MLDv2 signaling messages are encapsulated between the ClientXTR and the LISP EdgeRTR, therefore there is no requirement for the underlying network to support native multicast. If native access multicast is supported (for example native 5G multicast), then MobilityClient registration to H3ServiceEID safety channels may be integrated with it, in which case the mobile packet-core (EPC) element supporting it (eNB) will use this standard to register with the appropriate H3.r9 channels in its area.

Multicast update packets are of the following structure:

```
    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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ \
|Version| Traffic Class |           Flow Label                  |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |
|         Payload Length        |  Next Header  |   Hop Limit   |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |
|                                                               |  |
+                                                               +  |
|                                                               |  |
+                     Source H3-R9 EID Address                  +  |
|                                                               | IPv6
+                                                               +  |
|                                                               |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |
|                                                               |  |
+                                                               +  |
|                                                               |  |
+                        Group Address                          +  |
|                                                               |  |
+                                                               +  |
|                                                               | /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Source Port = xxxx      |       Dest Port = xxxx        | \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ UDP
|           UDP Length          |        UDP Checksum           | /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ \
|                                                               | Nexagons
Header
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ /
~                           Nexagons Payload                    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Outer headers = 40 (IPv6) + 8 (UDP) + 8 (LISP) = 56
Inner headers = 40 (IPv6) + 8 (UDP) + 4 (Nexagon Header) = 52

1500 (MTU) - 56 - 52 = 1392 bytes of effective payload

Type 1:key-value, key-value.. 1392 / (8 + 8) = 87 pairs
Type 2:value, key,key,key.. (1392 - 8) / 8 = 173 H3-R15 IDs

```
 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   |gzip |        Reserved         | Pair Count = X| Nexagon
Header
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ /
|                                                               |
+                       64 Bit H3-R15 ID                        +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                       64 Bit State                            +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                       64 Bit H3-R15 ID                        +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                       64 Bit State                            +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
 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 =   2   |gzip |     Reserved            |H3R15 Count = X| Nexagon
Header
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ /
|                                                               |
+                       64 Bit State                            +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                       64 Bit H3-R15 ID                        +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                       64 Bit H3-R15 ID                        +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                       64 Bit H3-R15 ID                        +
|                                                               |
```

+-+

` The remote EdgeRTRs homing MobilityClients in turn replicate the packet to the
MobilityClients registered with them.

We expect an average of 600 H3.r15 tiles of the full 7^6 (~100K) possible in H3.r9 to be part of any road. The H3.r9 server can transmit the status of all 600 or just those with meaningful states based on updated SLA and policy.

To Summarize:

(1) H3LISP Clients tune to H3.r9 mobility updates using rfc8378
H3LISP Client issue MLDv2 registration to H3.r9 HIDs
ClientXTRs encapsulate MLDv2 to EdgeRTRs who register (s,g)

(2) ServerXTRs encapsulate updates to EdgeRTRs who map-resolve (s,g) RLOCs
EdgeRTRs replicate mobility update and tunnel to registered EdgeRTRs
Remote EdgeRTRs replicate updates to registered ClientXTRs

7. Security Considerations

The nexagon layer3 v2v/v2i/v2x network is inherently more secure and private then alternatives because of the indirection. No car or infrastructure element ever communicates directly with MobilityClients. All information is conveyed using shared addressable geo-state. MobilityClients are supposed to receive information only from network channels as a trusted broker without indication as to the origin of the information. This is an important step towards better privacy, security, extendability, and interoperability.

In order to be able to use the nexagon mobility network for a given period, the mobility clients go through a DNS/AAA stage by which they obtain their clientEID identifiers-credentials and the RLOCs of EdgeRTRs they may use as gateways to the network. This MobilityClient <> EdgeRTR interface is the most sensitive in this network to privacy and security considerations.

The traffic on the MobilityClient<>EdgeRTR interface is tunneled and its UDP content may be encrypted; still, the EdgeRTR will know based on the LISP headers alone the MobilityClient RLOC and H3-R9 (~0.1sqkm) geo-spatial area to which a given client publishes or subscribes to.

For this reason we envision the ability of enterprise or groups of users to "bring their own" EdgeRTRs. BYO-RTR masks individual clients' IP-RLOC to H3-R9 association and is pre-provisioned to be able to use the mapping system and be on a white-list of EdgeRTRs aggregating H3ServiceEIDs.

Beyond this sensitive hop, the mapping system does not hold MobilityClientEIDs, and remote EdgeRTRs are only aware of MobilityClient ephemeral EIDs, not their actual IP RLOC or any other mobile-device identifiers. EdgeRTRs register in the mapping (s,g) H3-R9 multicast groups, but which clients reside beyond which EdgeRTR is not in the mapping system, only the AAA server is aware of that. The H3ServiceEIDs themselves decrypt and parse actual H3-R15 annotations; they also consider during this the MobilityClientEID credentials to avoid "fake-news", but again these are only temporary EIDs allocated to clients in order to be able to use the mobility network and not for their actual IP.

8. Acknowledgments

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

9. IANA Considerations

I. Formal H3 to IPv6 EID mapping

II. State enum fields of H3 tiles:

```
Field 0x: Traffic Direction {
0x - null
1x - Lane North
2x - Lane North + 30
3x - Lane North + 60
4x - Lane North + 90
5x - Lane North + 120
6x - Lane North + 150
7x - Lane North + 180
8x - Lane North + 210
9x - Lane North + 240
Ax - Lane North + 270
Bx - Lane North + 300
Cx - Lane North + 330
Dx - junction
Ex - shoulder
Fx - sidewalk
}

field 1x: Persistent or Structural {
 0x - null
 1x - pothole light
 2x - pothole severe
 3x - speed-bump low
```

```
 4x - speed-bump high
 5x - icy
 6x - flooded
 7x - snow-cover
 8x - snow-deep
 9x - construction cone
 Ax - gravel
 Bx - choppy
 Cx - blind-curve
 Dx - steep-slope
 Ex - low-bridge
}

field 2x: Transient Condition {
 0x - null
 1x - pedestrian
 2x - bike scooter
 3x - stopped car / truck
 4x - moving car / truck
 5x - first responder vehicle
 6x - sudden slowdown
 7x - oversized over-height vehicle
 8x - red-light-breach
 9x - light collision (fender bender)
 Ax - hard collision / casualty
 Bx - collision course car/structure
 Cx - recent collision residues
 Dx - hard brake
 Ex - sharp cornering
 Fx - freeing-parking
}

field 3x: Traffic-light Cycle {
 0x - null
 1x - 1 seconds to green
 2x - 2 seconds to green
 3x - 3 seconds to green
 4x - 4 seconds to green
 5x - 5 seconds to green
 6x - 6 seconds to green
 7x - 7 seconds to green
 8x - 8 seconds to green
 9x - 9 seconds to green
 Ax - 10 seconds or less
 Bx - 20 seconds or less
 Cx - 30 seconds or less
 Dx - 60 seconds or less
 Ex - green now
 Fx - red now
}
```

```
field 4x: Impacted Tile from Neighboring {
 0x - null
 1x - epicenter
 2x - light yellow
 3x - yellow
 4x - light orange
 5x - orange
 6x - light red
 7x - red
 8x - light blue
 9x - blue
 Ax - green
 Bx - light green
}

field 5x: Transient, Cycle, Impacted, Valid for Next{
 0x - null
 1x - 1sec
 2x - 5sec
 3x - 10sec
 4x - 20sec
 5x - 40sec
 6x - 60sec
 7x - 2min
 8x - 3min
 9x - 4min
 Ax - 5min
 Bx - 10min
 Cx - 15min
 Dx - 30min
 Ex - 60min
 Fx - 24hours
}

field 6x: LaneRightsSigns {
 0x - null
 1x - yield
 2x - speedLimit
 3x - straightOnly
 4x - noStraight
 5x - rightOnly
 6x - noRight
 7x - rightStraight
 8x - leftOnly
 9x - leftStraight
 Ax - noLeft
 Bx - noUTurn
 Cx - noLeftU
 Dx - bikeLane
 Ex - HOVLane
 Fx - Stop
```

```
}

field 7x: MovementSigns {
0x - null
1x - keepRight
2x - keepLeft
3x - stayInLane
4x - doNotEnter
5x - noTrucks
6x - noBikes
7x - noPeds
8x - oneWay
9x - parking
Ax - noParking
Bx - noStandaing
Cx - noPassing
Dx - loadingZone
Ex - railCross
Fx - schoolZone
}

field 8x: CurvesIntersectSigns {
0x - null
1x - turnsLeft
2x - turnsRight
3x - curvesLeft
4x - curvesRight
5x - reversesLeft
6x - reversesRight
7x - windingRoad
8x - hairPin
9x - pretzelTurn
Ax - crossRoads
Bx - crossT
Cx - crossY
Dx - circle
Ex - laneEnds
Fx - roadNarrows
}

field 9x: Current Tile Speed {
0x - null
1x - < 5kmh
2x - < 10kmh
3x - < 15kmh
4x - < 20kmh
5x - < 30kmh
6x - < 40kmh
7x - < 50kmh
8x - < 60kmh
9x - < 80kmh
```

```
Ax - < 100kmh
Bx - < 120kmh
Cx - < 140kmh
Dx - < 160kmh
Ex - > 160kmh
Fx - queuedTraffic
}

field Ax: Vehicle / Pedestrian Traffic {
0x - null
1x - probability of ped/vehicle on tile close to 100%
2x - 95%
3x - 90%
4x - 85%
5x - 80%
6x - 70%
7x - 60%
8x - 50%
9x - 40%
Ax - 30%
Bx - 20%
Cx - 15%
Dx - 10%
Ex - 5%
Fx - probability of ped/vehicle  on tile close to 0%, empty
}

filed Bx - reserved AV problem type in tile {
0x - null
1x - stall
}
field Cx - reserved
field Dx - reserved
field Ex - reserved
field Fx - reserved platoon identifier types {
0x - null
1x - US 7Char plate,
}
```

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-07 (work in progress), December 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>.

[RFC8378] Farinacci, D., Moreno, V., "Signal-Free Locator/ID Separation Protocol (LISP) Multicast", RFC8378, DOI 10.17487/RFC8378, May 2018, <https://www.rfc-editor.org/info/rfc8378>.

Authors' Addresses

Sharon Barkai
Nexar
CA
USA

Email: sbarkai@gmail.com

Bruno Fernandez-Ruiz
Nexar
London
UK

Email: b@getnexar.com

S ZionB
Nexar
Israel

Email: sharon@fermicloud.io

Rotem Tamir
Nexar
Israel

rotem.tamir@getnexar.com

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

USA

Email: natal@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

Jordi Paillissé-Vilanova
Technical University of Catalonia
Barcelona
Spain

Email: jordip@ac.upc.edu

Dino Farinacci
lispers.net
San Jose, CA
USA

Email: farinacci@gmail.com