LISP Working Group
Internet-Draft
Intended status: Experimental
Expires: January 1, 2020

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

September 1, 2019

Network-Hexagons: H3-LISP Based Mobility Network
draft-barkai-lisp-nexagon-06

Abstract

This document specifies combined use of H3 and LISP for mobility-networks:
- Enabling real-time tile by tile localized and indexed annotation of roads
- For sharing: hazards, blockages, conditions, maintenance, furniture..
- Between MobilityClients producing and consuming road-state information
- Via in-network-state, IPv6 addressable channel-grid of the physical world

Status of This Memo

Copyright Notice

Table of Contents

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 in-network-state which reflects the condition of each one-square-meter (~1sqm) hexagon road-tile. The lisp network maps & encapsulates traffic between MobilityClients endpoint-identifiers (EID, and, addressable (HID=>EID) tile-states, aggregated by H3Service EIDs.

The H3-LISP mobility network bridges timing-location gaps between the production and consumption of information by MobilityClients:
- vision, sensory, LIADR, AI applications - info-producers
- driving-apps, smart-infrastructure, command & control - info-consumers

This is achieved by putting the physical world on a shared addressable state-grid, an indirection.

The tile based geo-state mobility-network solves key issues in todays' vehicl to vehicle networking, where observed hazards are expected to be relayed or "hot-potato-tossed" (v2v) between vehicles without clear convergence i.e. given a situation observable by some of traffic it is unclear if the rest of the relevant traffic will receive consistent, conflicting, multiple, or non what so ever peer-to-peer v2v indication.

For example, when a vehicle experiences a sudden highway slow-down,"sees" man brake-lights or "feels" accelerometer, there is no clear way for it to share this annotation with vehicles 20-30 sec away, preventing potential pile-up. Or, when a vehicle crosses an intersection, observing opposite-lane obstruction - construction, double-park, commercial-loading / un-loading, garbage truck, or stopped school-bus - there is no clear way for it to alert vehicles turning in to that lane - as it crossed and drove away. Data may be replicated distorted or lost just like in a telephone-game.

These limitations are inherit since in most road situations vehicles are not really proper peers. They just happen to be in the same place at the same tim The H3-LISP mobility network solves limitations of direct vehicle to vehicle communication because it anchors per this place: timing, privacy, interoperability. This anchoring is done by MobilityClients (EIDs) communicating through in-network road-tile geo-states. Geo-states are aggregated and maintained by LISP addressable H3ServiceEIDs.

An important set of use-cases for state propagation of information to MobilityClients is to provide drivers heads-up alerts on hazards and obstacle beyond line of sight of both the drivers and in-car sensors: over traffic, around blocks, far-side-junction, beyond turns, and surface-curvatures. This highlights the importance of networks in providing road-safety.

To summarize the H3-LISP solution outline:

(1) Partition: 64bit indexed geo-spatial H3.r15 (~1sqm) road-tiles
(2) State: 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) Access: tunnels used between MobilityClients/H3ServiceEIDs <> LISP edge
(9) Access: ClientXTRs/ServerXTRs tunnel traffic to-from the LISP EdgeRTRs
(10) Control: EdgeRTRs register-resolve identity-location and mcast (s,g)

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

- MobilityClientA has seen MobilityClientB (20-30 sec) future, and, vice vers
- Clients share information using addressable shared-state routed by LISP Edg

- 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 networ

- 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, resident 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 source multicast address for updating MobilityClients on the state of the H3.r15 tiles aggregated-represented 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-exist 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 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. LISP RTRs decapsulate packets from ClientXTRs and ServerXTRs and re-encapsulate packets to ServerXTRs and ClientXTRs. The 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.

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
```

A MobilityClient which needs to use an H3-LISP mobility overlay network - instantiates a ClientXTR. It leverages DNS resolution to find EdgeRTR(s) in order to home to. ClientXTR is provisioned with an address for DNS resolvers, that help with the EdgeRTR discovery. The ClientXTR uses these DNS resolvers to resolve a query that includes the ClientXTR's current H3 index at resolution 9 (e.g. h3res9.example.net). To find its current H3.r9 index, the ClientXTR first translates its current geo-location to an H3 inde (e.g. gps snap-to-h3.r9).As a response to the query including the H3.r9 inde of the ClientXTR, the DNS resolver will return the IP address of the Edge RT that the ClientXTR can use to home to the H3-LISP mobility overlay.

The EdgeRTR discovery by the ClientXTR performed via DNS resolution so that:
1) EdgeRTRs are not tightly coupled to H3.r9 areas for easy load-balance
2) Mobility Clients do not need to constantly update EdgeRTR when it roams

In that sense, the same EdgeRTR may serve several H3.r9 areas for smooth ride continuity, and, several EdgeRTRs may load balance a 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-network-providers / radio-access-technologies.
- multiple cloud-edge hosting providers, public, private, hybrid.

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 (metro), 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 un-interrupted. This quality is insured by the LISP RFCs. The determinism of identities for MobilityClients to always refer to the correct H3ServiceEID is insured by H3 geospatial HIDs.

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

I. semi-random through DNS based load-balancing

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 mesh with the other RTRs in order to connect each Mobility Client with H3 Servers. Mobility Clients can (multi) home to the same RTR(s) throughout a ride.

II. geo-spatial, where a well known any-cast RTR aggregates H3.r9 hexagons

In this option we align an EdgeRTR with a geo-spatial cell area, very much like in Evolved Packet Core (EPC) solution. Mobility Clients currently roamin 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

Which ever way a ClientXTR is homed to an Edge RTR, DNS metro load-balance or well known geo-spatial map of IPs (a few 10Ks per large metro area), an authenticated MobilityClient EID can send: [64bitH3.15ID :: 64bitState] annotation to the H3.r9 H3ServiceEID. The H3.r9 IP HID can be calculated by clients algorithmically form the H3.15 localized snapped-to-tile annotation.

The ClientXTR encapsulates MobilityClient EID and H3ServiceEID in a packet sourced from the ClientXTR, destined to the EdgeRTR RLOC IP, Lisp port. EdgeRTRs then re-encapsulate annotation packets either to remote EdgeRTR (optionI) or to homed H3ServiceEID ServerXTR (option2).
The remote EdgeRTR aggregating H3ServiceEIDs re-encapsulates MobilityClient EID to ServerXTR and from there to the H3ServiceEID.

To Summarize Unicast:

(1) MobilityClients can send annotation state localized 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 used by clients to update H3.r15 tile state is also an I Multicast channel Source used to update subscribers on the aggregate state of the H3.r15 tiles in the H3.r9 Server.

We use rfc8378 signal free multicast to implement mcast channels in the overlay. The mobility network has many channels and relatively few subscribers per each. MobilityClients driving through or subscribing to a a 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 dest.

The advantage of explicit client MLDv2 registration trigger to rfc8378 is that the clients manage their own mobility mcast hand-over according to their location-direction moment vectors, and that it allows for otherwise silent, o 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 exampl native 5G multicast), then MobilityClient registration to H3ServiceEID safety channels may be integrated to it, in which case the evolved-packet-cor (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           | /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ \
|  Type =   X   |                  Reserved                     | Nexagons Head
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ /
~                       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|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                       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|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                       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 th 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 state based on update 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 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

I. Formal H3 to IPv6 EID mapping

II. State enum fields of H3 tiles:

```
Field 0x describes the "freshness" of the state {
 0x: less than 1Sec
 1x: less than 10Sec
 2x: less than 20Sec
 3x: less than 40Sec
 4x: less than 1min
 5x: less than 2min
 6x: less than 5min
 7x: less than 15min
 8x: less than 30min
 9x: less than 1hour
 Ax: less than 2hours
 Bx: less than 8hours
 Cx: less than 24hours
 Dx: less than 1week
 Ex: less than 1month
 Fx: more than 1month
}

field 1x: persistent weather or structural {
 0x - null
 1x - pothole
 2x - speed-bump
 3x - icy
 4x - flooded
 5x - snow-cover
 6x - snow-deep
```

```
 7x - construction cone
 8x - curve
}

field 2x: transient or moving obstruction {
 0x - null
 1x - pedestrian
 2x - bike
 3x - stopped car / truck
 4x - moving car / truck
 5x - first responder vehicle
 6x - sudden slowdown
 7x - oversized-vehicle
 8x - red-light-breach
}

field 3x: traffic-light timer countdown {
 0x - green now
 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 - 40 seconds or less
 Ex - 50 seconds or less
 Fx - minute or more left
}

field 4x: impacted tile from neighboring {
 0x - not impacted
 1x - light yellow
 2x - yellow
 3x - light orange
 4x - orange
 5x - light red
 6x - red
 7x - light blue
 8x - blue
}

field 5x: incidents {
 0x - clear
 1x - light collision (fender bender)
 2x - hard collision
```

```
 3x - collision with casualty
 4x - recent collision residues
 5x - hard brake
 6x - sharp cornering
}

field 6x - compiled tile safety rating {

}
field 7x: LaneRightsSigns {
 0x - stop
 1x - yield
 2x - speedLimit
 3x - straightOnly
 4x - noStraight
 5x - rightOnly
 6x - noRight
 7x - leftOnly
 8x - noLeft
 9x - noUTurn
 Ax - noLeftU
 Bx - bikeLane
 Cx - HOVLane
}

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

field 9x: CurvesIntersectSigns {
0x - turnsLeft
1x - turnsRight
2x - curvesLeft
3x - curvesRight
4x - reversesLeft
5x - reversesRight
```

6x - windingRoad
7x - hairPin
8x - 270Turn
9x - pretzelTurn
Ax - crossRoads
Bx - crossT
Cx - crossY
Dx - circle
Ex - laneEnds
Fx - roadNarrows
}

field Ax: Current Tile Speed {
0x - stopped
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 - < 180kmh
Fx - >= 200kmh
}

field Bx - reserved
field Cx - reserved
field Dx - reserved
field Ex - reserved
field Fx - reserved

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

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