Network-Hexagons: H3-LISP Based Mobility Network  
draft-barkai-lisp-nexagon-08

Abstract
This document specifies combined use of H3 and LISP for mobility-networks:
- Enabling real-time tile by tile indexed annotation of public roads
- For sharing: hazards, blockages, conditions, maintenance, furniture...
- Between MobilityClients producing-consuming road geo-state information
- Using addressable grid of channels of physical world state representation

Status of This Memo
This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 4, 2019.

Copyright Notice
Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

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
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 hexagon tile (~1sqm) in every road. The lisp network maps & encapsulates traffic between MobilityClients endpoint-identifiers (EID), and, addressable (HID=>EID) tile-states. States are aggregated byH3Service EIDs.

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

- vision, sensory, LIADR, AI applications - information producers
- driving-apps, smart-infrastructure, command & control - who consume it

This is achieved by putting the physical world on a shared addressable geo-state grid at the edge, a low-latency production-consumption indirection. Tile by tile based geo-state mobility-network solves key issues in today's vehicle to vehicle networking, where observed hazards are expected to be relayed or "hot-potato-tossed" (v2v without clear-reliable 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 no indication what so ever - using peer-to-peer propagation.

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 for 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 situation as it drives away.

Geo-state indirection also helps solve communicating advanced machine-vision and radar annotations. These are constantly evolving technologies, however, communicating the road enumerations they produce using peer-to-peer protocols poses a significant interoperability challenge - testing each new annotation by any sensor / OEM vendor and any other OEM and driving application vendor.

These peer-to-peer limitations are inherit 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. The H3-LISP mobility network solves limitations of direct vehicle to vehicle communication because it anchors per each geo-location: timing, security, privacy, interoperability. Anchoring is by MobilityClients communicating through in-network geo-states. Addressable tiles
are aggregated and maintained by LISP H3ServiceEIDs.

An important set of use-cases for state propagation of information to MobilityClients is to provide drivers heads-up alerts on hazards and obstacles 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) 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

- MobilityClientA has seen MobilityClientB (20-30 sec) 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, 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-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 hexagon mobility network.
EdgeRTRs decapsulate packets from ClientXTRs and 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

<table>
<thead>
<tr>
<th>0-</th>
<th>1-</th>
<th>2-</th>
<th>3-</th>
<th>4-</th>
<th>5-</th>
<th>6-</th>
<th>7-</th>
<th>8-</th>
<th>9-</th>
<th>A-</th>
<th>B-</th>
<th>C-</th>
<th>D-</th>
<th>E-</th>
<th>F-</th>
</tr>
</thead>
<tbody>
<tr>
<td>0123012301230123012301230123012301230123012301230123012301230123</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Subscription of MobilityClients to the mobility network is temporary-renewed while on the move and is not intended as 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 application specific virtual private network (VPN) that supports IPv6 EID addressable geo-state for 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 handful of customer-premise equipment (CPE/UE) types physically connected by wireline, or, by wireless spectrum to a specific service-provider. The Mobility VPN overlays potentially a number of wireless network providers and cloud-edge providers, and it involves dozens of Car-OEM, Driving-Applications, Smart-infrastructure vendors. It is therefore required 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

<table>
<thead>
<tr>
<th>MobilityClient</th>
<th>Domain Name Server</th>
<th>DIAMETER AAA</th>
<th>Mobility EdgeRTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>nslookup Nexagon</td>
<td>-----------------</td>
<td>--------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Mobility AAA IP</td>
<td>-----------------</td>
<td>--------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>ACR(AVP:IMSI/User/Password/Toyota)</td>
<td>-----------------</td>
<td>--------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>ACR(AVP ClientEID)</td>
<td>-----------------</td>
<td>--------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>ACA (Client::EID,EdgeRTR::RLOC)</td>
<td>-----------------</td>
<td>--------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>ACR (Interim)</td>
<td>-----------------</td>
<td>--------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Publish to H3ServiceEID / Subscribe MLDv2 H3ServiceEID</td>
<td>-----------------</td>
<td>--------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>multicast Updates from H3ServiceEIDs</td>
<td>-----------------</td>
<td>--------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>ACR (Interim)</td>
<td>-----------------</td>
<td>--------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>ACR (Interim)</td>
<td>-----------------</td>
<td>--------------</td>
<td>-----------------</td>
</tr>
</tbody>
</table>
Using this network-login / 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 a metro

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 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 EdgeRTRsRTRs throughout while moving.

II. Topological by any-cast

In this option we align an EdgeRTR with topological aggregation like in the Evolved Packet Core (EPC) solution. Mobility Clients currently 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
6. Mobility Unicast and Multicast

Which ever way a ClientXTR is homed to an Edge RTR 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 (option1) 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 IP
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
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, 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 to it, in which case the evolved-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:
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

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 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 nexagon layer3 v2v/v2i/c&c 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 the network 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 is the most sensitive
interface in the network as far as privacy-security.

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
a given client publishes in 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. The H3ServiceEIDs them selves of-course
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 basic communications.

8. Acknowledgments

This work is partly funded by the ANR LISP-Lab project #ANR-

9. IANA Considerations

I. Formal H3 to IPv6 EID mapping

II. State enum fields of H3 tiles:

Field 0x State Freshness (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 or Structural {
  0x - null
  1x - pothole
  2x - speed-bump low
  3x - speed-bump high
  4x - icy
  5x - flooded
  6x - snow-cover
  7x - snow-deep
  8x - construction cone
  9x - gravel
}

field 2x: Transient 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
  9x - light collision (fender bender)
  Ax - hard collision
  Bx - collision with casualty
  Cx - recent collision residues
  Dx - hard brake
  Ex - sharp cornering
}

field 3x: Traffic-light Cycle {
  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 - red now
}
field 4x: impacted tile from neighboring {
  0x - not impacted
  1x - epicenter
  2x - light yellow
  3x - yellow
  4x - light orange
  5x - orange
  6x - light red
  7x - red
  8x - light blue
  9x - blue
}

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

field 7x: 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 8x: Current Tile Speed {
  0x - queued
  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 9x: Lanes and Shoulders {
  0x - Lane >> Edge1
  1x - Lane >> Vertex1
  2x - Lane >> Edge2
  3x - Lane >> Vertex2
  4x - Lane >> Edge3
  5x - Lane >> Vertex3
  6x - Lane >> Edge4
  7x - Lane >> Vertex4
  8x - Lane >> Edge5
  9x - Lane >> Vertex5
  Ax - Lane >> Edge6
  Bx - Lane >> Vertex6
  Cx - Junction
  Dx - sidewalk
  Ex - shoulder
  Fx - ditch
}

filed Ax - reserved
filed Bx - reserved
filed Cx - reserved
filed Dx - reserved
filed Ex - reserved
filed 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,

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The
 Locator/ID Separation Protocol (LISP)", RFC 6830,
 DOI 10.17487/RFC6830, January 2013,
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 PaillissÃ©-Vilanova
Technical University of Catalonia
Barcelona
Spain
Email: jordip@ac.upc.edu

Dino Farinacci
lispers.net
San Jose, CA
USA

Email: farinacci@gmail.com