LISP Working Group
Internet-Draft
Intended status: Experimental
Expires: October 4, 2019

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

May 4 2019

H3-LISP Based Mobility Network
draft-barkai-lisp-nexagon-02

Abstract

This document specifies combined use of H3 and LISP for mobility-networks:
- Enabling real-time tile-by-tile localized-annotation of road-conditions
- Sharing of road annotations: hazards, blockages, maintenance, furniture
- Between MobilityClients, which produce-consume road-tile state information
- Using formal in-network-state addressable-indexed-maintained by H3Servers.

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 reflecting the condition of each one-square-meter (1sqm) hexagon road-tile. The lisp network maps traffic between MobilityClients endpoint identifiers (EID), and, hex-id (HID to EID) addressable tile-states. States maintained by H3Servers.

The H3-LISP mobility network bridges timing-location gaps between the

production and consumption of information by MobilityClients:
- vision, sensory, LIADR, AI information producers
- apps/smart-infrastructure information consumers
And is achieved by mobility in-network addressable-state indirection.

This capability addresses a key issue in todays' vehicle to vehicle (v2v) networking, where observed hazards are expected to be "hot-potato-tossed" between vehicles without clear convergence. For example, when a vehicle experiences a sudden highway slow-down, by "seeing" breaks light-up in-front, an-or by accelerometer, there is no clear way for it to share this annotation with vehicles which are 20-30 seconds away, potentially preventing major pile-ups, especially on icy or poor visibility conditions. Or, when a vehicle crosses an intersection, observing opposite=lane-obstruction - construction, double-park, commercial loading / un-loading, garbage truck, or stopped schoo -bus - there is no clear way for it to alert vehicles turning in to that lane as it drives straight on its away.

The H3-LISP mobility network solves this limitation of direct vehicle to vehicle communication by MobilityClients (EIDs) communicating with in-network road-tile in-network states. These states are aggregated-maintained by LISP EID addressable H3Servers. An important set of use-cases involves propagation of condition information to MobilityClients to provide drivers heads-up alert on hazards and obstacles beyond line of sight: over traffic, around blocks, far-side-junction, beyond turns and surface-curvatures. This highlights the importance of networks in providing road-safety greater then any isolated or autonomous vehicle safety technology.

To summarize the H3-LISP solution outline:

(1) Partition: Geo-spatial H3.r15 (1sqm) road-tiles indexed by 64bit HIDs
(2) Geo-spatial tile-state values complied to 64bit condition representation
(3) Geo-spatial H3Servers use H3.r9 resolution to aggregate H3.r15 road-tiles
(4) H3Servers function also as multicast channels of H3.r15 state updates
(5) H3Servers are distributed for in-network scale, latency, and throughput
(6) An overlay tunneled-network is used to map the mobility-network traffic
(7) Tunneled overlay network is used to implement signal-free mcast channels
(8) Tunnels also used between MobilityClients/H3Servers <> and the LISP edge
(9) ClientXTRs and ServerXTRs tunnel traffic to and from the LISP EdgeRTRs
(10) EdgeRTRs register-resolve identity-location as well as mcast registratio

```
|-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                  |
|---------------------------------------------------------------|
```

---   ---   ---

```
                         /     \          /     \          /    \         H3Servers
                        | H3-R9 |        | H3-R9 |        |      |
                    ---  \ ___ / ---  --- \_____/ ---  --- \____/ ---
                        ServerXTR        ServerXTR        ServerXTR
                                        \    |     /
                                          EdgeRTRs
                                             ||
                                 <LISP MapAssisted Overlay>
                                             ||
                                          EdgeRTRs
                                            /|\
                    ((((|))))                                   ((((|))))
                       /|\                                         /|\
                       RAN                                         RAN

                ...................../    \/    \/    \....<< ClientXTR
           - - - - - - - - - - - -  H3-R15  - - H3-R15 - - - - - - - -    Mobility Clients
                   ClientXTR >> ....\____/\____/\____/..........
```

Each H3.r9 hexagon is a server with corresponding H3 ID. Bound to that server is a LISP xTR, called a ServerXTR, resident to deliver encapsulated packets t and from the H3Server and the LISP Edge. EdgeRTRs are used to re-tunnel packe from MobilityClients to that H3Server. Each H3Server HID is also a source multicast address for updating MobilityClients as to the state of the H3.r15 tiles contained in the H3.r9 H3Server.

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

H3Server: Is a server process which maintains state in part of a H3 geo-spati grid. H3Server is responsible for a single hexagon in a given resolution. In this specification each H3Server is responsible for one H3.r9 hexagon. An H3Server can roam between hosting locations for maintenance or failover it will always have the same EID IPv6 address based on its HID hexagon ID. The H3Server application sends unicast and multicast packets from its EID. It has a light-weight LISP protocol stack to tunnel packets aka ServerXTR.

H3Server 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 an H3Server 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 H3Server EIDs. Th 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 H3Server EIDs and glean MobilityClient EIDs when it decapsulates packets. EdgeRTRs store H3Server EIDs and their own RLOC of where the H3Server 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 H3Servers share this well known index
(3) A 64-bit BDD state value is associated with each H3 tile
(4) Tile state is compiled 16 fields of 4-bits or 16 enums

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

When a MobilityClient wants to join an H3-LISP mobility network, it first instantiates a ClientXTR. It then leverages DNS resolution to find EdgeRTR(s it can home to. The ClientXTR is provisioned with an anycast address for the DNS resolvers, that help with the EdgeRTR discovery. The ClientXTR uses thes anycasted 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.res9 index, the ClientXTR first translates its current geo-location to an H3 index (e.g. gps snap-to-res9-hex).As a response to the query including the H3.res9 index of the ClientXTR, the DNS resolver will return the IP address of the Edge RTR that the ClientXTR can use to home to the H3-LISP mobility overlay. The EdgeRTR discovery by the ClientXTR is performed via DNS resolution so that 1) EdgeRTRs are not tightly coupled to H3.r9 areas, and 2) the car does not need to update its EdgeRTR every time i roams to another H3.r9 area. In that sense, the same EdgeRTR may serve several H3.r9 areas, and, several EdgeRTRs may serve the same H3.r9 area. When a MobilityClient::ClientXTR is homed to an EdgeRTR it is ready to communicate with state H3Servers and leverage/support the mobility network.

5. Mobility Clients-Network-Servers

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, or 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 H3Server to be instantiated, moved, or failed-ove 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 H3Servers is insured by H3 geospatial HIDs. LISP location-identity-separation makes HIDs network addressable elements.

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) H3Server 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) H3Server HID multicasts

```
MobilityClients <> ClientXTR <Network Provider > EdgeRTR
                             ||
          < Map-Assisted Mobility-Network Overlay>
                             ||
      EdgeRTR <Cloud Provider> ServerXTR <> H3Servers
```

6. Mobility Unicast and Multicast

Which ever way a ClientXTR is homed to an Edge RTR, via DNS metro load-balanc or via a 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 HID server. 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 the H3Server HID 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 H3Server ServerXTR (option2).
The remote EdgeRTR aggregating H3Servers re-encapsulates MobilityClient EID, H3Server HID to the ServerXTR and from there to the H3Server.

To Summarize Unicast:

```
(1) Mobility Clients can send annotation state localized an H3.r15 tile
    These annotations are sent to an H3.r9 mobility server
(2) Source Client EID and Dest HID are encapsulated XTR <> RTR <> RTR <> XTR
    * RTRs can map-resolve re-tunnel HIDs to remote RTR RLOC
(3) RTRs re-encapsulate original source-dest to ServerXTRs
    ServerXTRs decapsulate packet and serve the original packet to H3Server
```

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 one. MobilityClients driving through or subscribing to a a H3.r9 area can explicitly issue an rfc4604 MLDv2 in-order to subscribe, or, it 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 its allows for otherwise silent, or, non annotating clients. The advantage of EdegRTR implicit registration is less signaling required. At any case MLDv2 signaling messages are encapsulate 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 eMBMS native 5G), then MobilityClient registration to H3Server road-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.

EdgeRTRs note the subscribed MobilityClient stack XTRs and register as channe subscribers in the mapping system (Source, Group) entry. This is done at the first subscription request, if additional MobilityClients homed to the same EdgeRTR register for the same channels the EdgeRTR registration covers them.

Upon receiving a multicast packet the EdgeRTR homing H3.r9 Servers resolve the (S,G) remote EdgeRTRs registered for the channel and replicates the packe ` 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 10K 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 IGMP-Report registration to H3.r9 HIDs
    ClientXTRs encapsulate IGMP-report 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
}

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
 10x - noLeftU
 11x - bikeLane
 12x - HOVLane
}
```

```
field 8x: MovementSigns {
0x - noPass
1x - keepRight
2x - keepLeft
3x - stayInLane
4x - doNotEnter
5x - noTrucks
6x - noBikes
7x - noPeds
8x - oneWay
9x - parking
10x - noParking
11x - noStandaing
12x - loadingZone
13x - truckRoute
14x - railCross
15x - School
}

field 9x: CurvesIntersectSigns {
0x - turnsLeft
1x - turnsRight
2x - curvesLeft
3x - curvesRight
4x - reversesLeft
5x - reversesRight
6x - windingRoad
7x - hairPin
8x - 270Turn
9x - pretzelTurn
10x - crossRoads
11x - crossT
12x - crossY
13x - circle
14x - laneEnds
15x - roadNarrows
}
field Ax - reserved
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: sharon.barkai@getnexar.com

Bruno Fernandez-Ruiz
Nexar
London
UK

Email: b@getnexar.com

Ohad Serfaty
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