

Locator/ID Separation Protocol
Internet-Draft
Intended status: Experimental
Expires: January 7, 2016

M. Menth
A. Stockmayer
M. Schmidt
University of Tuebingen
July 6, 2015

LISP Hybrid Access
draft-menth-lisp-ha-00

Abstract

Hybrid access (HA) allows simultaneous usage of multiple access links. Advantages are increased bandwidth and improved resilience. This document presents LISP Hybrid Access (LISP-HA), a mechanism to provide HA based on LISP technology. The document discusses potential changes needed to perform dynamic load-balancing and per packet load-balancing, which both increase the efficiency of HA. To that end, modified usage of some fields in the LISP header is proposed. Discussed use cases include the bundling of multiple access technologies for mobile devices and residential access routers. Additionally, we provide some considerations how LISP-HA can be deployed by providers.

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 <http://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 January 7, 2016.

Copyright Notice

Copyright (c) 2015 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

(<http://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.	Terminology	3
3.	LISP-HA-Architecture	4
3.1.	Basic Operation	4
3.2.	Message Sequence Diagram for Basic Operation.	6
3.3.	Policy-Based Path Selection	8
3.4.	Operation of an HAxTR behind a NAT	8
3.5.	Extensions for Dynamic Load Balancing	10
3.6.	Extensions for Packet-Based Load Balancing	11
3.7.	Deployment Considerations	12
4.	Packet Formats	12
4.1.	Dataplane Header	12
4.2.	Control Message	13
5.	Use Cases	14
5.1.	Connecting Residential Users to the General Internet	14
5.2.	Smartphones with Mobile Node.	15
6.	Security Considerations	17
7.	Acknowledgements	17
8.	References	17
8.1.	Normative References	17
8.2.	Informative References	17
	Authors' Addresses	18

1. Introduction

Hybrid access (HA) enables a device to simultaneously use multiple access links both in upstream and downstream direction. A challenge of HA is to make load balancing of the traffic onto multiple paths transparent to endpoints. HA may be supported on various layers. Multilink PPP (ML-PPP) [RFC1990] offers support for fragmented protocol data units (PDU) on the same local network. Therefore, it cannot combine network layer paths so that it is unable to bundle paths provided by different Internet service providers. A network architecture for HA is presented in [I-D.lhwxyz-hybrid-access-network-architecture]. It focuses on bundling DSL and LTE for residential access by means of dedicated customer premises equipment (CPE) which does not support mobile devices in general. Multipath TCP (MP-TCP) leverages multiple paths

on the transport layer [RFC6824]. It requires both source and destination endpoint to support MP-TCP. MP-TCP proxies are currently discussed for inter-operation with non-MP-TCP nodes [I-D.hampel-mptcp-proxies-anchors]. HA can be also supported by MP-TCP, but that approach is limited to TCP traffic. Furthermore, supporting policies such as cheapest link first seems challenging with this approach.

LISP by itself has basic capabilities to support HA with static load balancing that is not agnostic to current link loads and characteristics. That means, a multihomed LISP xTR may perform inbound traffic engineering. This is achieved by setting appropriate weights for multiple RLOCs in register messages so that it receives traffic over more than a single interface. Outbound traffic may be sent over multiple interfaces according to local policies.

This document proposes LISP-HA as a novel solution for HA with improved load balancing capabilities for better resource efficiency.

2. Terminology

Hybrid Access (HA): Using two or more access lines to improve bandwidth and resilience; both technologies are used at the same time.

Mobile Node (MN): A LISP node that includes its own xTR and can connect to more than a single network at a time [I-D.meyer-lisp-mn].

Mapping System (MS): The LISP Mapping System as defined in RFC 6830 [RFC6830].

LISP Tunnel Router (xTR): A combination of ITR and ETR [RFC6830].

LISP Proxy Tunnel Router (PxTR): Used to communicate with the legacy Internet [RFC6830].

LISP Reencapsulating Tunnel Router (RTR): LISP router to forward LISP-TE packets [I-D.farinacci-lisp-te].

LISP NAT Traversal Router (NTR): A LISP router that allows to communicate with LISP nodes behind a NAT [I-D.ermagan-lisp-nat-traversal].

LISP Canonical Address Format (LCAF): Extension to the AFI type system to associate the AFI, e.g. with policies [I-D.farinacci-lisp-lcaf].

Explicit Locator Path (ELP): A mapping storing a sequence of RLOCs, indicating RTRs that receive and forward LISP packets to the next RLOC in that list; defined by LISP-TE [I-D.farinacci-lisp-te].

Hybrid Access xTR (HAXTR): An xTR with multiple RLOCs that supports LISP-HA.

Hybrid Access RTR (HARTR): An RTR that supports LISP-HA.

Hybrid Access Load Balancing (HALB): Function in HAXTR and HARTR that distributes traffic over multiple paths.

Hybrid Access Reorder and Feedback (HARF): Function in HAXTR and HARTR that reorders traffic sent by a corresponding HALB over multiple paths and provides feedback about the link condition to the HALBs.

3. LISP-HA-Architecture

This section describes the basic operation of LISP-HA as well as policy-based path selection, its operation with LISP nodes behind NATs, extensions for dynamic load balancing, and extensions for packet-based load balancing. Finally, we consider how it could be useful for providers to operate their own HARTR.

3.1. Basic Operation

LISP-HA allows to load-balance traffic over multiple paths between a HAXTR and HARTR. This is transparent to nodes not on the path between HAXTR and HARTR. Load-balancing works in both directions, therefore, both the HAXTR and the HARTR implement a HA Load Balancing function (HALB) and a HA Recombination function (HARF).

We present how LISP-HA makes EIDs globally reachable over multiple paths between HARTR and HAXTR. To that end, we consider the setting illustrated in Figure 1.

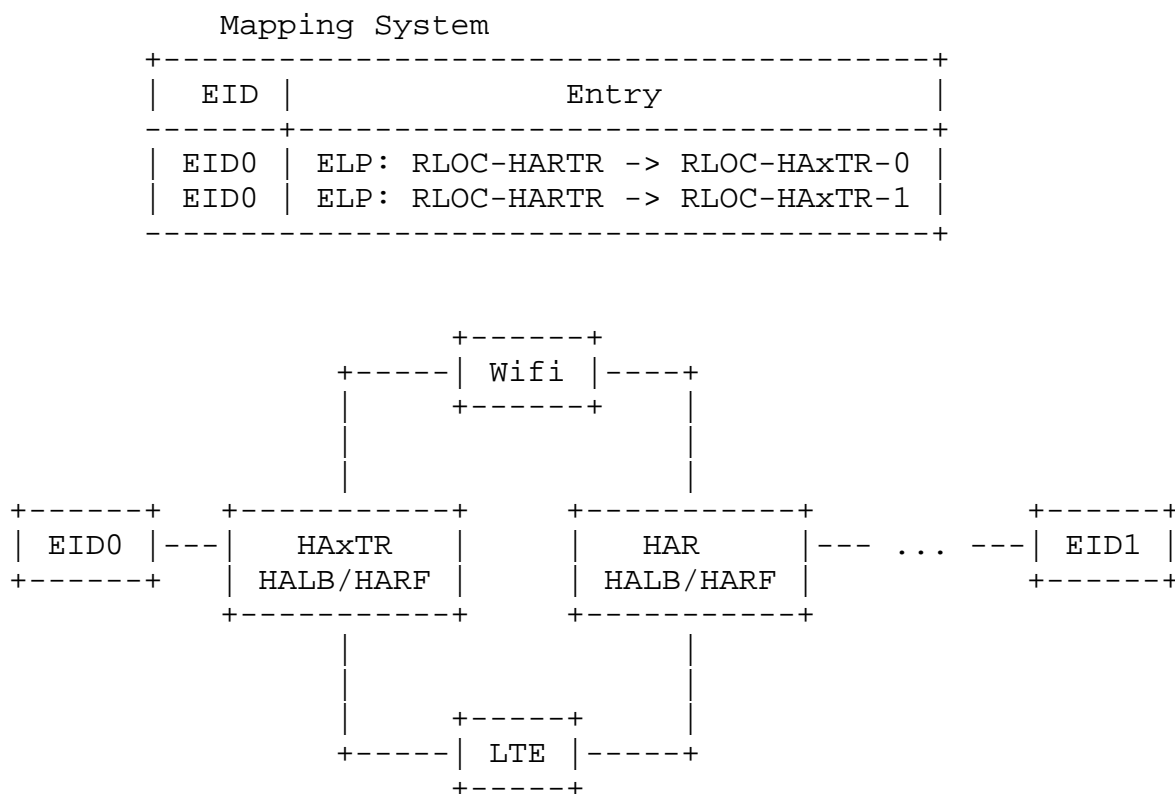


Figure 1: LISP-HA load-balances traffic between HAxTR and HARTR over multiple network layer paths.

A HAxTR is configured with the address of a HARTR and registers its EID prefixes at the MS. To that end, it uses explicit locator paths (ELPs), containing the RLOC of the HARTR in the penultimate position of the ELP and one of its own RLOCs in the last position of the ELP. The HAxTR must send one register message for each of its RLOCs and over the interface that corresponds to that RLOC so that the MS can detect whether the HAxTR is behind a NAT.

The HALB functions of the HAxTR and the HARTR distribute the traffic over multiple network layer paths between them. Flow-based or packet-based load-balancing may be supported.

Figure 2 shows a communication scenario between two LISP nodes. The HAxTR is connected to the Internet / LISP net over multiple access technologies and LISP-HA is applied between HAxTR and HARTR. The endpoints EID0 and EID1 exchange messages over the HAxTR, the HARTR, and the xTR. The figure shows the destination EIDs and RLOCs of these messages. The HAxTR/xTR add RLOCs to the messages through encapsulation, the HARTR exchanges the RLOCs, and the xTR/HAxTR

remove RLOCs from the messages through decapsulation. The HAxTR and xTR add RLOCs to the messages through encapsulation and the HARTR exchanges the RLOCs. In upstream direction, the HAxTR adds the RLOC of the HARTR and selects the path by choosing the appropriate interface. The HARTR exchanges its own RLOC by the RLOC of the xTR. In downstream direction, the xTR adds the RLOC of the HARTR as specified in the ELP. The HARTR exchanges this RLOC with the appropriate RLOC of the HAxTR. Thereby the desired path is selected.

Upstream example:

```

EID0  --->  HAxTR  --->  HARTR  --->  xTR  --->  EID1
          EID1          EID1          EID1          EID1
                RLOC-HARTR          RLOC-xTR

```

Downstream example:

```

EID0  <---  HAxTR  <---  HARTR  <---  xTR  <---  EID1
          EID0          EID0          EID0          EID0
                RLOC-HAxTR-i          RLOC-HARTR

```

Figure 2: Destination EIDs and RLOCs of a LISP packet between two communicating endpoints.

3.2. Message Sequence Diagram for Basic Operation.

Figure 3 and Figure 4 illustrate the basic operation of LISP-HA by a diagram showing all control and data messages exchanged to send data in upstream and downstream direction. In both cases we assume that the HAxTR and the xTR have registered their EIDs EID_0 and EID_1 at the MS, but the required mappings are not available in caches.

When the endhost with EID_0 starts sending data to EID_1 , it forwards them to its HAxTR. The HAxTR requests the mapping for RLOC of EID_1 from the MS to verify that EID_1 is globally reachable before sending it to the HARTR. The HAxTR encapsulates the packet with the configured address of the HARTR as destination RLOC, using the access line selected by its HALB. Upon receipt of the packet, the HARTR also requests the mapping for EID_1 , exchanges the destination RLOC in the packet with the RLOC of the xTR provided in the map-reply and sends the packet to the xTR. Upon receipt, the xTR decapsulates the LISP packet and forwards it to the endhost with EID_1 .

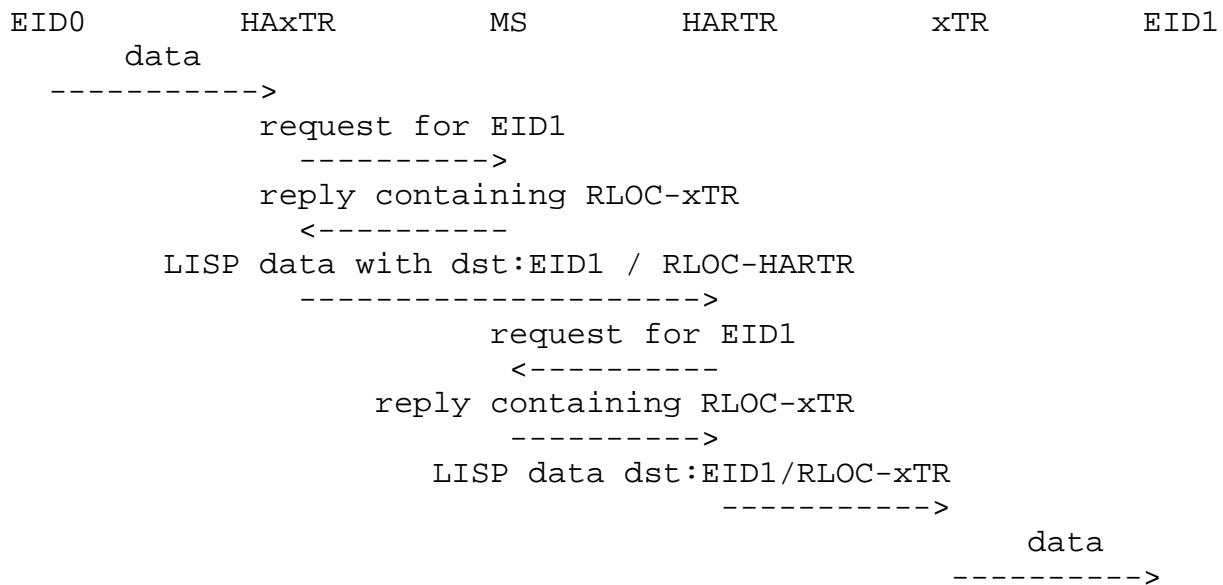


Figure 3: Message sequence diagram for upstream communication.

When endhost with EID1 starts sending data to EID0, it forwards them to its xTR. The HAxTR requests the mapping for EID0 from the MS, receives the ELP for EID0, encapsulates the packet with RLOC-HARTR, and sends it to the HARTR. Upon receipt of the packet, the HARTR requests the mapping for EID0 from the MS. The HALB function of the HARTR selects the path to the HAxTR, the HARTR exchanges the destination RLOC in the LISP packet with RLOC-HAxTR and sends the packet over the determined path. Upon receipt, the HAxTR decapsulates the LISP packet and forwards it to the endhost with EID0.

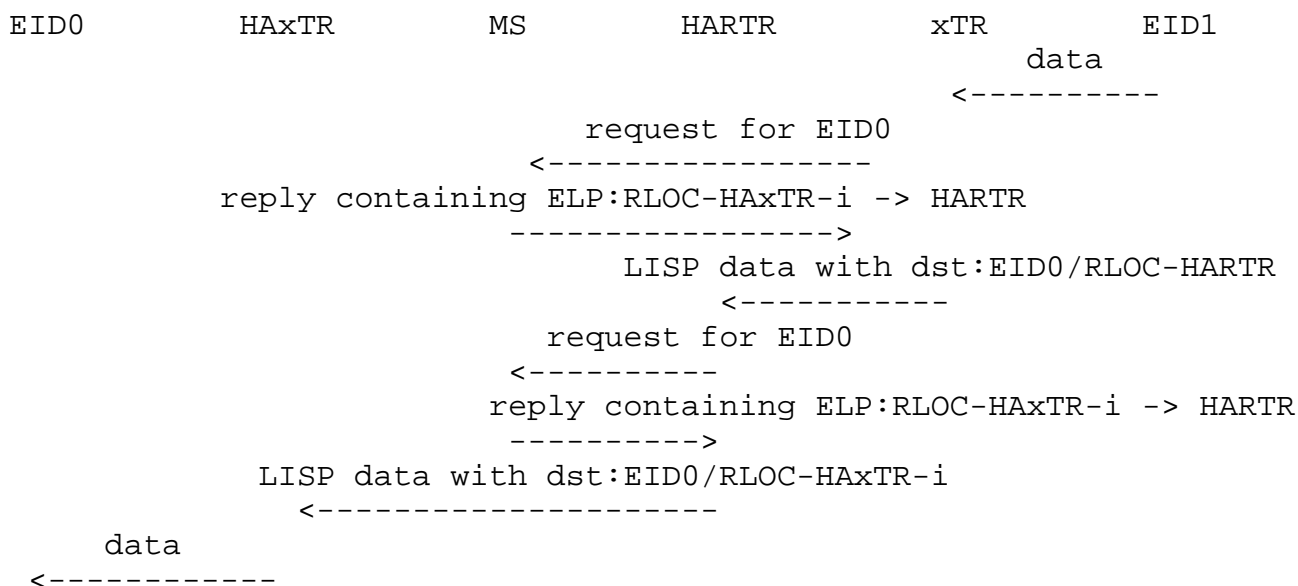


Figure 4: Message sequence diagram for downstream communication.

3.3. Policy-Based Path Selection

For specific kinds of traffic, e.g., for realtime communication, the usage of specific paths may be desired. LISP-HA supports such requirements through LISP LCAF extensions both on upstream and downstream. To that end, the HAxTR is configured with LCAF extensions, e.g., indicating that traffic for realtime communication must be forwarded over a specific path. The HAxTR uses this LCAF as local policy to encapsulate the traffic. In addition, the HAxTR registers the same LCAF at the MS. As a consequence, xTRs encapsulate traffic towards the HAxTR using the specific RLOC in the LCAF.

3.4. Operation of an HAxTR behind a NAT

A HAxTR may be located behind a NAT. We consider this case as this scenario is common for HAxTRs connected via LTE.

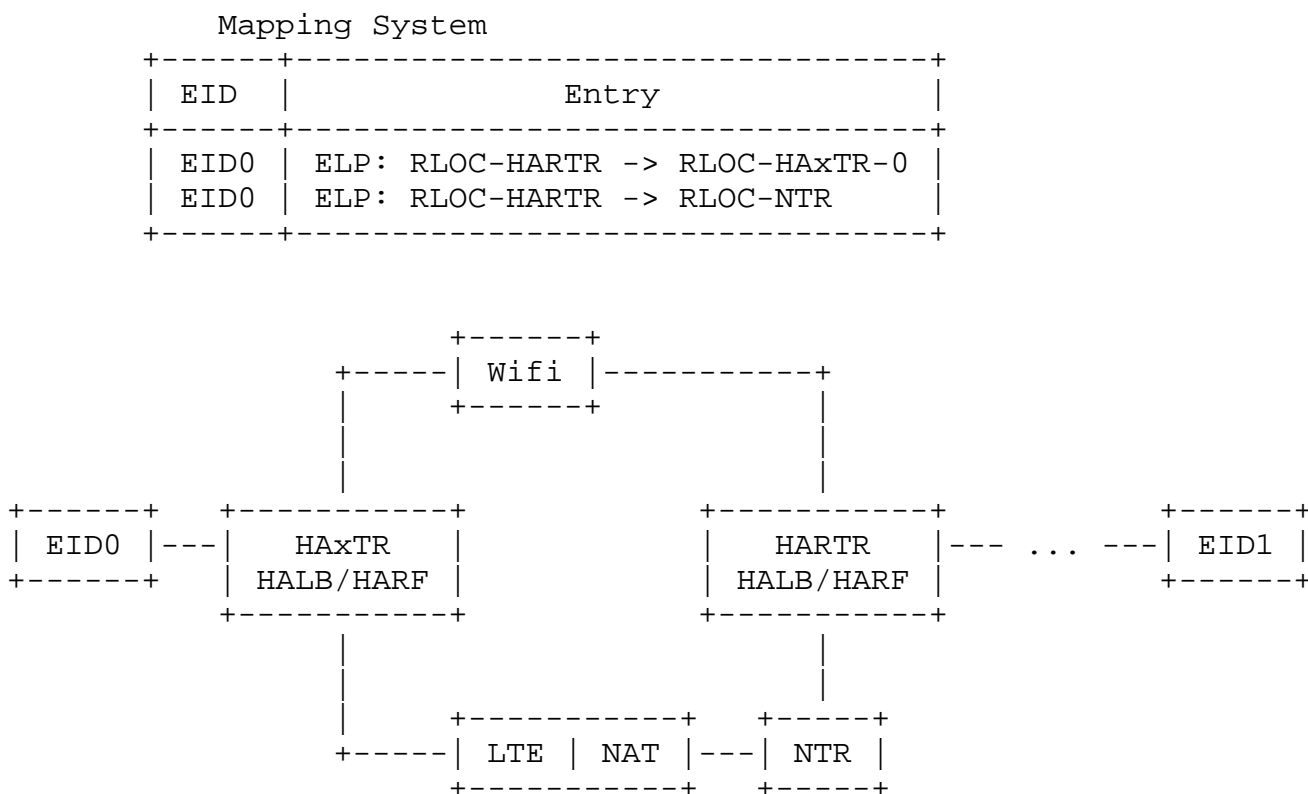


Figure 5: LISP-HA load-balances traffic between HAxTR and HARTR while part of the traffic traverses a NAT.

We show how LISP-HA leverages existing LISP NAT traversal [I-D.ermagan-lisp-nat-traversal] so that it does not require additional mechanisms to cope with NATs.

Figure 5 shows a HAxTR that is connected to the Internet / LISP net via LTE and through a NAT. The HAxTR registers its RLOC at the MS, the MS detects that the RLOC in the LISP register message differs from the RLOC in the outer header encapsulating it. As a consequence, the MS does not register the mapping and informs the HAxTR proposing a list of potential NAT Traversal Routers (NTRs). Then, the HAxTR selects one of the NTRs and registers again at the MS via this NTR. The NTR exchanges the RLOC in the mapping by its own RLOC (RLOC-NTR). As a result, traffic for the HAxTR is directed to the NTR which forwards it to the HAxTR. This mechanism works for LISP-HA only if the HAxTR registers its ELPs over the corresponding interfaces; otherwise the MS cannot securely detect that the HAxTR is behind a NAT.

The usage of general NTRs leads to triangle routing and adds significant delay which may be prohibitive. However, an NTR may be combined with the HARTR and configured with the HAxTR so that triangle routing and added path delay are avoided.

To cope with carrier grade NATs, messages need to be exchanged frequently enough over the NTR to avoid that NAT table entries are removed. This, however, is to be fixed for the NTR draft [I-D.ermagan-lisp-nat-traversal]. Moreover, HAxTR and HARTR exchange feedback messages for dynamic load balancing frequently enough so that NAT table entries will not be removed.

3.5. Extensions for Dynamic Load Balancing

One goal of HA is to increase a HAxTR's overall access bandwidth by combining the bandwidth of all available paths. However, static load balancing leads to statistical variations [Menth06p] so that some paths are already overloaded while others are underutilized. Dynamic load balancing takes the current load on the links into account and can achieve better resource utilization without overloading paths.

We propose dynamic load balancing for LISP-HA with the purpose to increase resource efficiency, thereby providing higher bandwidth to the user. A challenge is that path properties like available bandwidth and delay are possibly unknown and vary over time, especially if the path is shared and includes wireless links. Also flow rates vary over time and the rate of elastic flows depends on path characteristics. Nevertheless, incipient congestion can be inferred from increasing path-specific packet loss and delay. So the idea is to obtain feedback about path-specific packet loss and delay and leverage this information for improved load balancing. To that end, the HARF functions continuously monitor the quality of all paths perceived by transmitted traffic so that the HALB can leverage path-specific information about packet loss and delay for load balancing decisions. In the following we briefly explain how a pair of corresponding HALB and HARF functions on a one-directional path can obtain information about packet loss and delay.

To estimate packet loss, the HALB function equips the overall packet stream with sequence numbers before load-balancing them over multiple interfaces. These sequence numbers are also used for packet reordering if needed.

The HALB function counts the number of packets sent per path up to some checkpoint sequence number. The corresponding HARF function counts the number of packets received per path up to the same checkpoint sequence numbers and reports them to the corresponding HALB. The difference between the number of sent and received packets

between checkpoint sequence numbers gives an estimate about packet loss per path.

Measuring packet delay between HALB and HARF is rather difficult as their clocks may not be synchronized and have a drift. Therefore, only relative delays are measured over time.

The HALB equips packets with timestamps before sending them to an interface and the HARF computes the difference to its current time upon receipt, yielding a biased packet delay due to potentially missing clock synchronization. Assuming that all biased packet delays have the same bias, the evolution of the biased packet delay reflects the evolution of the real packet delay. The HARF reports to the HALB the path-specific numbers of recently received packets and delay information. Delay metrics may be minimum, maximum, average delay as well as a standard deviation. However, the metrics are not clear, yet, because the load-balancing algorithm is still under research. The same holds for the frequency of checkpoint sequence numbers and the frequency of feedback reports from the HARF to the HALB.

3.6. Extensions for Packet-Based Load Balancing

Per-flow load balancing forwards packets of a flow over the same path. Therefore, packets will arrive in order unless they are reordered for other reasons on the path. Thus, per-flow load balancing avoids additional packet reordering by LISP-HA. However, it is more difficult to efficiently use the bandwidth of existing paths with per-flow load-balancing than with per-packet load balancing. Moreover, as flow-based load balancing forwards packets of a single flow only over a single path, very large flows cannot profit from several paths.

Packet-based load balancing may cause reordered packets in particular if paths have different delay. As packet reordering has negative impact on some transport protocols and applications, it should either be avoided or packets should be reordered. We propose that the HARF function performs packet reordering if needed so that the HALB function can load-balance per packet if desired. However, packet reordering causes some additional delay leading to a tradeoff between reordered packets and increased delay.

A HALB may be configured to load-balance certain flows per flow and other flows per packet, depending on the needs of specific transport protocols or applications. In addition, the HARC may reorder packets from per-packet load balanced flows into order. To that end, such packets need to be marked with a "Reorder" flag so that other packets do not suffer from reordering delay.

The HARF leverages the sequence numbers of all packets for packet reordering. It holds back packets which require reordering before forwarding so long that it is unlikely that packets from the same flow with lower sequence numbers will be received. Details are subject to future work.

3.7. Deployment Considerations

HARTRs are infrastructure that need to be operated and cause costs in the first place. However, the operation of a HARTR allows to perform traffic engineering by appropriate load balancing. E.g., an LTE network operator prefers to offload its resources and can achieve that with the HARTR if other paths have sufficient capacity. Thus, ISPs may have interest to set up HARTRs to have strategic control over traffic forwarding in the network.

Apart from that, offering a HARTR as a service by a third party for a small fee may also be an option. Thereby, customers could book cheap contracts for LTE and residential access and combine these technologies via the third party HARTR.

4. Packet Formats

A modified LISP dataplane header is presented to convey information from HALB to HARF and a new LISP control message is proposed to report feedback information from HARF to HALB.

4.1. Dataplane Header

LISP-HA reuses some fields of the dataplane header of encapsulated LISP data packets to convey information for dynamic load balancing from the HALB to the HARF. The original dataplane header is illustrated in Figure 6. The usage of the fields is defined in [RFC6830].

```

x x x x 1 0 0 0
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|N|L|E|V|I|flags|           Nonce/Map-Version           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Instance ID           |           LSBs           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 6: Original dataplane header.

The Nonce was proposed for security purpose, but has no application in the LISP-HA context. An Instance ID is used to uniquely identify

the source in a virtual address space, which is neither applicable in the LISP-HA context. Three header flags are unused. Therefore we propose to reuse the Nonce and the Instance ID fields of the original dataplane header definition to convey sequence numbers and timestamps from the HALB to the HARF together with an indication whether a packet needs to be forwarded in-order. The modified LISP header is illustrated in Figure 7. The modified header fields are explained in the following.

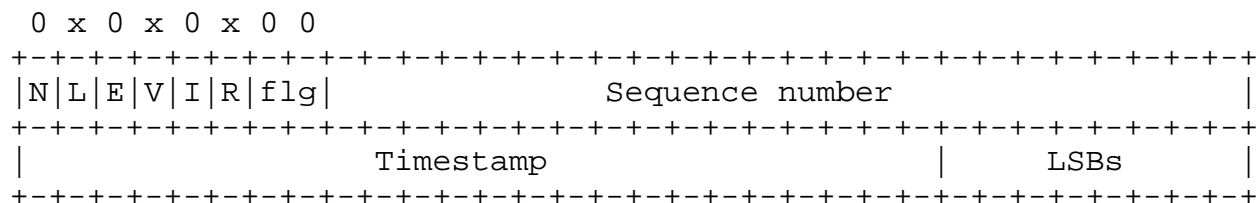


Figure 7: Modified LISP Dataplane Header.

R: Reorder flag to indicate whether a packet needs to be forwarded in-order. It is 1 if the packet has to be reordered, and it is 0 if the packet can be forwarded immediately.

Sequence number: Sequence number for packets sent from HALB to HARF, used for packet loss detection and reordering.

Timestamp: The lower 24 bits of the timestamp of the sender.

All other fields: All fields not explicitly described here have the same meaning as in [RFC6830].

In upstream direction the HAxTR sends packet with the modified header to the HARTR. The HARTR modifies the modified header in upstream direction to be compliant to [RFC6830]. In downstream direction the HARTR receives LISP packets with a header format compliant to [RFC6830] and modifies the header as proposed in this section. The HAxTR removes this header. NTRs may be located between HAxTR and HARTR, but they do not need to process the modified header fields. Therefore, only HAxTR and HARTR need to implement, understand, and process the modified header format.

4.2. Control Message

We propose the unused Type number 5 for LISP-HA Feedback Messages to report feedback about packet loss and delay from the HARF to the HALB. These messages contain information about the number of received packets between sequence number checkpoint and information

about packet delay as described in Section 3.5. The exact values and format is are subject to further research.

5. Use Cases

In the following, we describe two typical use cases of LISP-HA. The first use case explains how LISP-HA can be used for residential users to benefit from HA to connect to the Internet. In the second use case, we present how a mobile node, e.g. a smartphone, can use LISP-HA.

5.1. Connecting Residential Users to the General Internet

For customers with cable internet high bandwidth can not be guaranteed as cable internet is based on a shared medium. Especially in the evening hours when many customers need bandwidth at the same time, the rate can drop significantly. For those customers it would be a great benefit if they could bundle their, sometimes slow, cable access with LTE to improve their bandwidth. Figure 8 illustrates a potential solution using LISP-HA.

The HAxTR may be implemented in the home router, it has a public EID which it registers at the MS. The customer typically uses a private address space in his home LAN which is connected through the NAT of the home router. The HAxTR is connected to the Internet through a DSL and an LTE interface and there is a provider NAT on the LTE path. An NTR is used to make the HAxTR reachable vial LTE from the Internet. As LISP nodes cannot communicate directly with the Internet, the HAxTR is configured with an appropriate PxTR, to send traffic to non-LISP IP addresses. To minimize path stretch and delay, both NTR and PxTR may be integrated in the same box as the HARTR.

EID	Entry
EID0	ELP: RLOC-HARTR -> RLOC-HAxTR-0
EID1	ELP: RLOC-HARTR -> RLOC-NTR

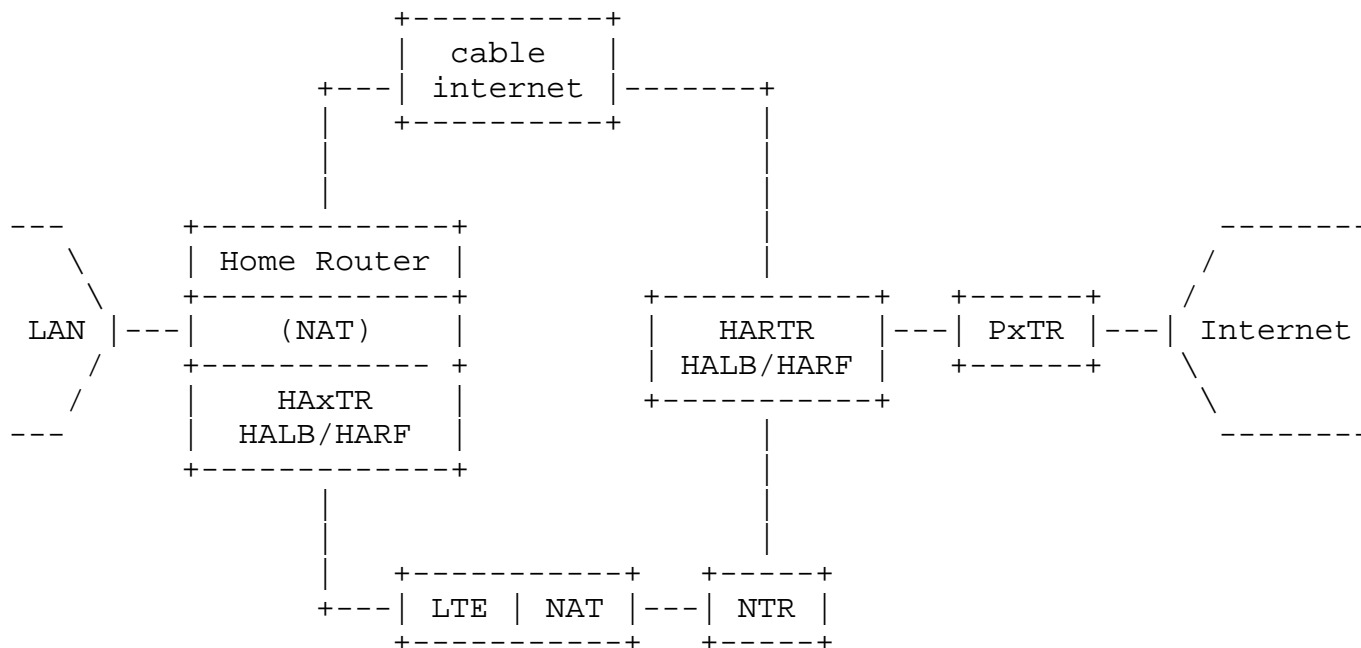


Figure 8: LISP-HA used for residential access to the Internet.

This solution may be attractive to users who are not even aware of LISP. Their traffic is just load-balanced over DSL and LTE between the home router and the HARTR and decapsulated by the PxrTR. In case of a public address space in the customer's LAN the HAxTR can register the entire subnet of the LAN as EID prefix at the MS. The PxrTR has to announce the EIDs registered by the HAxTR to the Internet so that traffic for the HAxTR is sent to PxrTR.

5.2. Smartphones with Mobile Node.

Today's smartphones include multiple radio interfaces that allow to connect to multiple access technologies like LTE and Wifi at the same time. The default policy that is implemented in smartphones is to offload traffic from the cellular network to Wifi access. However, it would be desirable to use both technologies to increase the available bandwidth for normal internet traffic like downloads and to select the Wifi connection for realtime applications like VoIP. In

Figure 9, we present a potential setup how a MN can use LISP-HA to realize the scenario.

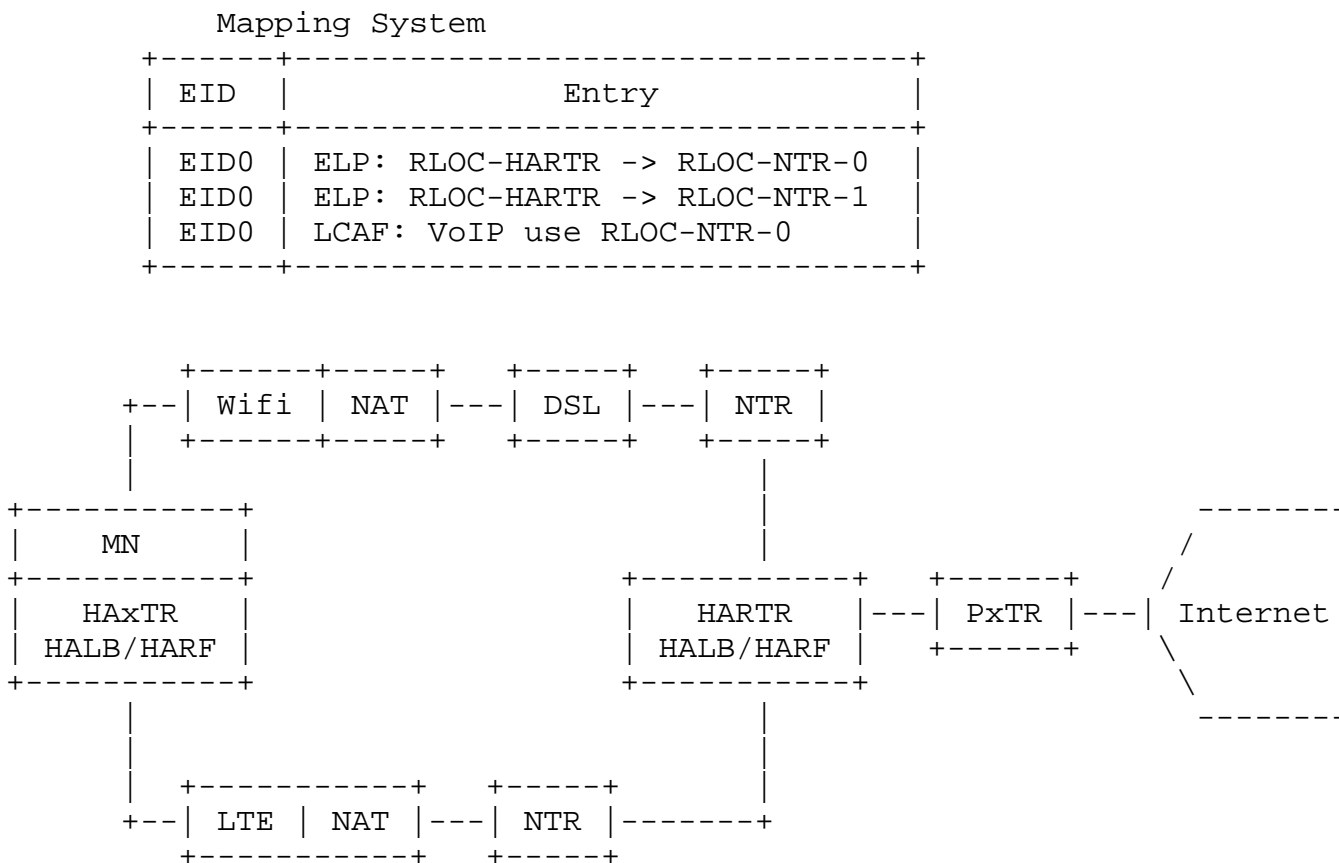


Figure 9: LISP-HA used for Smartphones with Mobile Node.

To realize HA on a MN, the MN has to implement its own HAxTR consisting of HALB and HARF. Typically, the MN has access to the Internet most of the time using cellular networks like LTE or HSDPA. So the MN registers its EID through the cellular network at the MS. A provider NAT in the LTE network is handled like described in Section 3.4. Wifi access becomes available if the MN is in reach of a public Wifi hotspot, in the home LAN of the user or other known Wifi networks. Once Wifi access is available, the MN registers its EID over Wifi, too. Additionally, the MN can register an LCAF for VoIP traffic to use Wifi. If Wifi is available no more because the MN left the Wifi cell, the MN should de-register the Wifi mappings at the MS.

6. Security Considerations

HA by itself does not raise any security concerns. However, LISP-HA is based on LISP so that the same security considerations apply as described in [RFC6830]. There is no authentication of endhosts at the xTR and no authentication between xTRs which allows every node to send any amount of traffic to any xTR which makes it vulnerable to DOS attacks. This also counts for the HARTR and the HAxTR. LISP traffic is not encrypted, so if it is required to encrypt the communication, this has to be realized by the endhosts.

7. Acknowledgements

The authors would like to thank Gerd Pflueger and Wilhelm Boeddinghaus for their helpful input and discussions.

8. References

8.1. Normative References

- [RFC1990] Sklower, K., Lloyd, B., McGregor, G., Carr, D., and T. Coradetti, "The PPP Multilink Protocol (MP)", RFC 1990, August 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6824] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, January 2013.
- [RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, January 2013.

8.2. Informative References

- [I-D.ermagan-lisp-nat-traversal]
Ermagan, V., Farinacci, D., Lewis, D., Skriver, J., Maino, F., and C. White, "NAT traversal for LISP", draft-ermagan-lisp-nat-traversal-07 (work in progress), February 2015.
- [I-D.farinacci-lisp-lcaf]
Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format (LCAF)", draft-farinacci-lisp-lcaf-10 (work in progress), July 2012.

[I-D.farinacci-lisp-te]

Farinacci, D., Kowal, M., and P. Lahiri, "LISP Traffic Engineering Use-Cases", draft-farinacci-lisp-te-08 (work in progress), March 2015.

[I-D.hampel-mptcp-proxies-anchors]

Hampel, G. and T. Klein, "MPTCP Proxies and Anchors", draft-hampel-mptcp-proxies-anchors-00 (work in progress), February 2012.

[I-D.lhwxz-hybrid-access-network-architecture]

Leymann, N., Heidemann, C., Wasserman, M., Xue, L., and M. Zhang, "Hybrid Access Network Architecture", draft-lhwxz-hybrid-access-network-architecture-02 (work in progress), January 2015.

[I-D.meyer-lisp-mn]

Farinacci, D., Lewis, D., Meyer, D., and C. White, "LISP Mobile Node", draft-meyer-lisp-mn-12 (work in progress), January 2015.

[Menth06p]

Milbrandt, J., Humm, K., and M. Menth, "Adaptive Bandwidth Allocation: Impact of Routing and Load Balancing on Tunnel Capacity Requirements", in Proceedings of 2nd Conference on Next Generation Internet Design and Engineering, Valencia, Spain , April 2006.

Authors' Addresses

Michael Menth
University of Tuebingen
room B302, Institute of Computer Science
Sand 13
Tuebingen 72076
Germany

Phone: +49 7071 29-70505
Email: menth@uni-tuebingen.de

Andreas Stockmayer
University of Tuebingen
room B305, Institute of Computer Science
Sand 13
Tuebingen 72076
Germany

Phone: +49 7071 29-70511
Email: andreas.stockmayer@uni-tuebingen.de

Mark Schmidt
University of Tuebingen
room B305, Institute of Computer Science
Sand 13
Tuebingen 72076
Germany

Phone: +49 7071 29-70510
Email: mark-thomas.schmidt@uni-tuebingen.de