 The definitive version of this paper has been published in 22nd International Teletraffic Congress (ITC), 2010, 10.1109\/itc.2010.5608725.

Improvements to LISP Mobile Node

Michael Menth, Dominik Klein, and Matthias Hartmann
University of Würzburg, Institute of Computer Science, Germany

***Abstract*— The Locator/Identifier Separation Protocol (LISP) is a new routing architecture for the Internet that separates local and global routing. It offers more flexibility to edge networks and has the potential to reduce the growths of the BGP routing tables. Recently, a concept for mobility in LISP (LISP Mobile Node, LISP-MN) was presented. We analyze LISP-MN and show that it needs double mapping lookups in all LISP gateways, leads to triangle routing under some conditions, and requires double encapsulation. We propose gradual improvements to LISP-MN that avoid these drawbacks under many conditions.**

I. Introduction

The current interdomain routing faces scalability and flexibility problems. More and more edge networks want to do multihoming, traffic engineering, and provider changes without renumbering their equipment. This requires provider-independent (PI) addresses, which add more entries to the rapidly growing BGP routing tables [1]. The maximum allowed IP prefix length is limited by ISPs, so that companies with a relatively small PI address space are still very restricted in using such advanced techniques. These are drivers for a more scalable and flexible Internet addressing and routing.

The currently favored solution is the separation of global Internet routing and local routing/addressing in edge networks [2]. Communication sessions with other nodes are established using identifier addresses (IDs), which might also be used for local routing. IDs are not advertised in global BGP routing, therefore, a globally routable locator is added to each packet to send them over the Internet. The locator for an ID can be requested from a special mapping system. This architecture decouples the combined identification and location functions of today's IP addresses. Edge networks can change their Internet service provider (i.e., their locators) while keeping their identifier address space. Traffic engineering capabilities and routing scalability are also improved [3].

The Internet Engineering Task Force (IETF) currently standardizes the Locator/Identifier Separation Protocol (LISP) [4]. It implements routing separation and fits well into today's Internet routing concept. Edge networks need to be upgraded with new gateways but hosts in so-called LISP domains do not need to be changed. Interworking between LISP domains and the legacy Internet is possible [5]. An architecture for the integration of mobile nodes is also provided (LISP Mobile Node, LISP-MN) [6]. It allows multi-homing for mobile nodes and does not need home and foreign agents like in mobile IPv4 [7] so that triangle routing can be avoided to some extent.

This work was funded by the Federal Ministry of Education and Research of the Federal Republic of Germany (support code 01 BK 0800, G-Lab, http://www.german-lab.de/). The authors alone are responsible for the content of the paper.

In this paper, we study the encapsulation and forwarding structure of LISP-MN and point out several of its shortcomings. LISP-MN requires LISP gateways to generally perform double mapping lookups, it leads to triangle routing under some conditions, and it sometimes needs double encapsulation of data packets. We propose gradual improvements to the LISP-MN that can often avoid the listed disadvantages.

Section II illustrates LISP and its interworking techniques with the non-LISP Internet. Section III reviews the current LISP mobility architecture and analyzes its encapsulation and forwarding structure. Section IV proposes gradual improvements to LISP-MN and Section V summarizes this work.

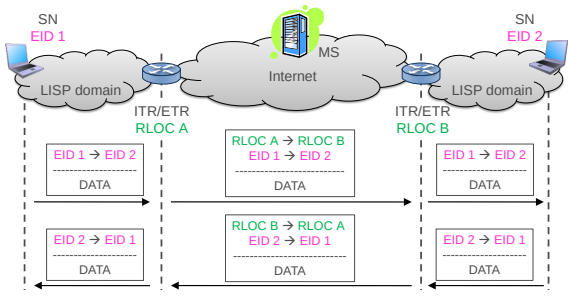
II. Locator/Identifier Separation Protocol (LISP)

In this section we review basics of the LISP architecture ([4], draft-version 06), its interworking mechanisms ([5], draft-version 01), and of LISP-MN ([6], draft-version 01), a proposal for the integration of mobile nodes in a LISP-based Internet.

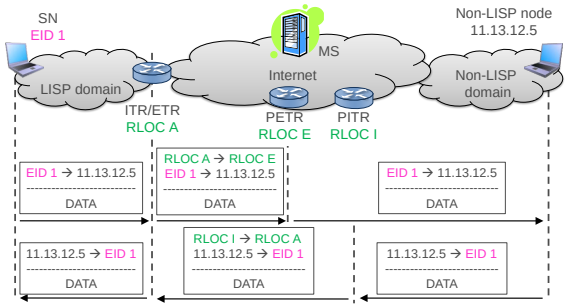
A. LISP

LISP separates addressing in edge networks from addressing in transit networks. The IP address of a "stationary node" (SN) in a LISP domain is called endpoint identifier (EID). It is routable only in the SN's LISP domain. In contrast to EIDs, globally routable IP addresses are called routing locators (RLOCs). Nodes in the non-LISP Internet are designated as "non-LISP nodes". Two SNs in the same LISP domain communicate with each other like non-LISP nodes communicate today. SNs in different LISP domains communicate with each other through the gateways that separate their LISP domains from the global Internet. They have ingress and egress tunnel router (ITR/ETR) functionality. A mapping system (abbreviated MS in figures and algorithms) returns EID-to-RLOC mappings upon map-requests. The RLOC serves to locate the ETR of the LISP domain hosting the node with a specific EID in the global Internet. If a map-request contains an EID for which no locator is registered, the mapping system returns a negative map-reply. When a SN sends a packet to a SN in another LISP domain, the destination EID is not routable in the source domain and forwarded to a default ITR. This ITR queries the mapping system with the destination EID and receives the RLOC of a destination ETR. To reduce communication overhead, this query may be answered from a local cache at the ITR [8]. The ITR then encapsulates the packet with that RLOC as destination address and its own RLOC as source address and tunnels it through the Internet to the destination ETR. The destination ETR decapsulates the packet which is then carried to the destination node using the

EID which is site-locally routable. Figure 1(a) illustrates the source and destination addresses in the headers of packets that are exchanged between nodes in different LISP domains.



(a) Communication between two stationary nodes (SNs) in different LISP domains.



(b) Communication between a SN and a non-LISP node.

Fig. 1. Encapsulation and forwarding in LISP.

B. Interworking with the non-LISP Internet

Interworking methods between LISP domains and the non-LISP Internet are described in [5]. When a LISP node communicates with a non-LISP node in the non-LISP Internet, the ITR receives a negative map-reply from the mapping system upon lookup. Then, the ITR may send the packet without encapsulation to the destination node. However, Internet service providers (ISPs) often ensure that they forward only packets with addresses belonging to their customers. This source address filtering is usually implemented by a unicast Reverse Path Forwarding (uRPF) check in provider edge routers. As EIDs are not routable by these routers, packets that have an EID as source address in the outermost IP header are dropped. In [5, Sect. 9], some workarounds based on configuration of provider edge (PE) routers are suggested to bypass this mechanism for known non-routable EID blocks belonging to customer networks (e.g., by including static routes to the EID space of the site in the PE router). Alternatively, a proxy ETR (PETR) is introduced, which is located in a network that can forward packets without uRPF check. ITRs are configured with the RLOC of some PETR and tunnel traffic destined to non-LISP nodes to this PETR. This prevents that packets with EIDs as source address are dropped by the provider edge router. To minimize the path stretch caused by triangle routing, the ITR is configured with the RLOC of a PETR that is located close to the edge network. This is shown in Figure 1(b).

We now consider the reverse direction. When a non-LISP node addresses a packet towards an EID, the packet cannot be routed in the global Internet. Therefore, so-called proxy ITRs (PITRs) are introduced. They announce via anycast all IP prefixes reserved for EIDs so that they attract traffic destined to EIDs. The PITR queries the mapping system with the destination EID of a packet, receives the RLOC of the destination ETR, and encapsulates the packet. The PITR forwards the packet to the ETR which decapsulates it. Then, the packet is carried to the destination node using the EID. PETRs and PITRs also help LISP networks to connect to IPv6 networks when intermediate networks do not support IPv6.

Network address translation (NAT) is another interworking proposal [5, Sect. 6]. However, with NAT, only SNs in LISP domains can establish communication with non-LISP nodes outside the LISP domain, but not vice-versa. We do not consider this method in the following.

C. The LISP-MN Architecture

A LISP "mobile node" (MN) has a permanent EID which is used for identification but not for forwarding. In contrast to non-LISP nodes and SNs in LISP domains, MNs have upgraded LISP-MN networking stacks. When a MN roams into a network, it receives a care-of-address (e.g., via DHCP) under which it is locally reachable in the destination domain and registers it as locator (LOC) in the mapping system[1]. When the MN roams into a non-LISP network, the obtained care-of-address is globally reachable[2] and serves as RLOC for the MN. When the MN roams into a LISP domain, the obtained care-of-address is only site-locally reachable and serves just as local locator (LLOC)[3]. All LLOCs of a LISP domain are pre-registered in the mapping system together with the RLOCs of the domain's ETRs.

LISP-MN assumes that a MN forms a separate LISP domain and implements ITR/ETR functionality for incoming and outgoing traffic except for DHCP traffic (see [6, Sect. 6]). To send traffic, a MN must encapsulate outgoing traffic to some ETR or PETR, i.e., it must be configured with the RLOC of a PETR. To receive traffic, the traffic must be tunneled to the MN from some ITR, PITR, or another MN.

LISP-MN implicitly expects enhanced functionality for normal ITRs and PITRs to communicate with MNs in other LISP domains. When a packet is sent from a LISP domain to a MN in another LISP domain, the ITR receives the outbound packet addressed to the EID of the MN. It queries the mapping system with the destination EID of that packet and encapsulates the packet to the returned locator which is the LLOC of the corresponding MN. Then, the ITR queries the mapping system again with the returned LLOC and encapsulates the packet with the returned locator which is the RLOC of the ETR

[1]We assume that MNs are configured with appropriate addresses to access the mapping system.

[2]We consider only networks that are not behind NATs.

[3]The concept "LLOC" was not proposed in [6], but we use it to facilitate the distinction between site-locally and globally routable locators (LLOCs, RLOCs).

of the destination LISP domain (see [6, Sect. 9]). Thus, two lookups are needed. As it is not possible to infer the locator type (RLOC, LLOC) from the returned mapping, ITRs and PITRs must always perform two mapping lookups as they do not know a priori whether the packet is destined to a MN. If the corresponding node is not a MN in a LISP domain, the second lookup yields a negative map-reply and the packet is encapsulated just once. In contrast to ITRs and PITRs, MNs query the mapping system and encapsulate packets only once.

III. Analysis of LISP-MN

In this section, we illustrate the encapsulation and forwarding structure of LISP-MN under different conditions which has not been presented in the draft. Then, we summarize observed disadvantages of LISP-MN.

A. Encapsulation and Forwarding Structure of LISP-MN

In the following, we look at 9 different scenarios where MNs are involved in communication and illustrate packet encapsulation and forwarding with LISP-MN. To that end, we consider MNs in LISP or non-LISP domains that communicate with MNs or non-LISP nodes in non-LISP domains or with SNs or MNs in the same or another LISP domain.

1) A MN in a non-LISP domain communicates with another MN in a non-LISP domain: The MN addresses a packet towards the EID of the other MN, and encapsulates and sends the packet towards the globally routable RLOC for this EID. The same procedure applies for the reverse direction.

2) A MN in a non-LISP domain communicates with a SN in a LISP domain: The MN addresses a packet towards the EID of the SN and encapsulates the packet towards the RLOC for the SN's EID. The packet is forwarded to the ETR of the destination LISP domain where it is decapsulated and then forwarded to the SN (see Figure 2(a)). In the reverse direction, the SN addresses a packet towards the EID of the MN and the packet is forwarded to a default ITR. The ITR encapsulates the packet towards the RLOC for the MN's EID and sends it to the MN.

3) A MN in a non-LISP domain communicates with a MN in a LISP domain: The MN in the non-LISP domain addresses a packet towards the EID of the MN in the LISP domain. It encapsulates the packet towards the LLOC of the other MN's EID and sends it. The packet is carried towards a PITR which encapsulates the packet again towards the RLOC of the ETR of the destination LISP domain and sends it. The packet is carried to that ETR which decapsulates the packet, and then forwards it to the MN in the LISP domain (see Figure 2(b)). In the reverse direction, the MN in the LISP domain addresses the packet towards the EID of the MN in the non-LISP domain. It encapsulates it towards the other MN's RLOC and the packet is forwarded to a LISP gateway. After a mapping lookup which is negative, the LISP gateway tunnels the packet to its configured PETR to hide the LLOC as source address. The PETR decapsulates the packet and forwards it to the MN in the non-LISP domain. In both cases, triangle routing occurs due to the use of a PITR and a PETR.

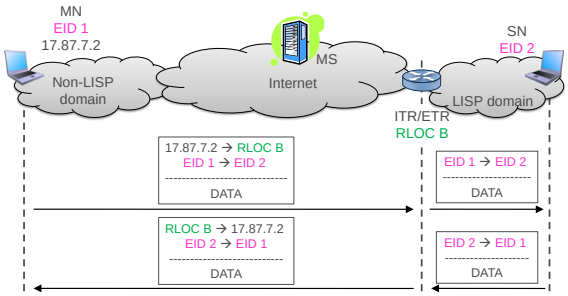
4) A MN in a non-LISP domain communicates with a non-LISP node: The MN in the non-LISP domain addresses a packet towards the IP address of a non-LISP node. As there is no locator for that address, it encapsulates the packet towards the RLOC of its configured PETR and sends the packet. The PETR just strips off the outer header and the packet is forwarded to the non-LISP node (see Figure 2(c)). In the reverse direction, the non-LISP node addresses a packet towards the EID of the MN and sends it. The packet is carried to a PITR. The PITR encapsulates it to the RLOC of the MN and sends it to that node. In both directions we observe triangle routing via the PETR or the PITR.

5) A MN in a LISP domain communicates with a non-LISP node: The MN in the LISP domain addresses a packet towards the IP address of a non-LISP node. As there is no locator for that address, it encapsulates the packet towards the RLOC F of its configured PETR and sends the packet. The ITR receives the packet and sees the LLOC in the source field. Therefore, it also encapsulates the packet towards the RLOC E of its configured PETR and sends it. The packet is first carried to PETR E which decapsulates it, then to PETR F which decapsulates it again, and eventually the non-encapsulated packet is carried to the non-LISP node (see Figure 2(d)). In the reverse direction, the non-LISP node addresses a packet towards the EID of the MN. The packet is forwarded to a PITR. The PITR first encapsulates the packet towards the LLOC of the MN and then towards the RLOC for that LLOC. The packet is carried to the ETR which decapsulates it and passes it on to the MN in the LISP domain. In the forward direction we observe "quadrangle" routing via the PETRs of the ITR and the PETR of the MN while in the reverse directions we observe triangle routing via the PITR.

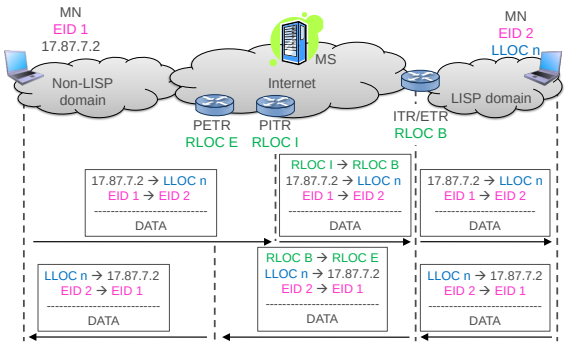
6) A MN in a LISP domain communicates with a SN in another LISP domain: The MN addresses a packet towards the SN's EID and encapsulates the packet to the SN's RLOC. The packet is sent to the ITR. The ITR sees an LLOC in the source field and tunnels the packet to its PETR. The PETR decapsulates the packet and forwards it to the ETR of the destination domain. The ETR decapsulates the packet and it is forwarded to the SN (see Figure 2(e)). In the reverse direction, the SN addresses a packet towards the EID of the MN and forwards it to its default ITR. The ITR first encapsulates the packet towards the LLOC for the MN's EID and then to the RLOC for this LLOC. The packet is carried to the ETR of the MN's domain, which strips off the outer encapsulation header and sends the packet to the MN.

7) A MN in a LISP domain communicates with a MN in another LISP domain: The MN addresses a packet towards the EID of the other MN and encapsulates it towards the LLOC of the other MN. The packet is carried to the ITR which queries the RLOC for the LLOC, encapsulates the packet accordingly, and sends it. The ETR decapsulates the packet and forwards it to the destination node (see Figure 2(f)). The reverse direction works likewise.

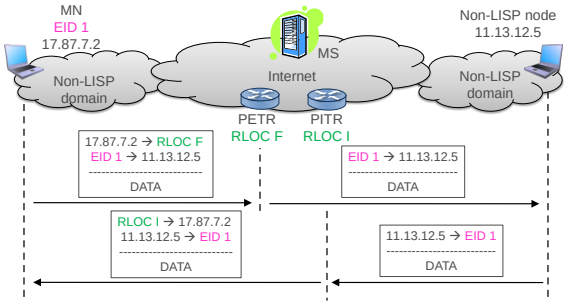
8) A MN in a LISP domain communicates with a SN in the same LISP domain: The MN addresses a packet towards



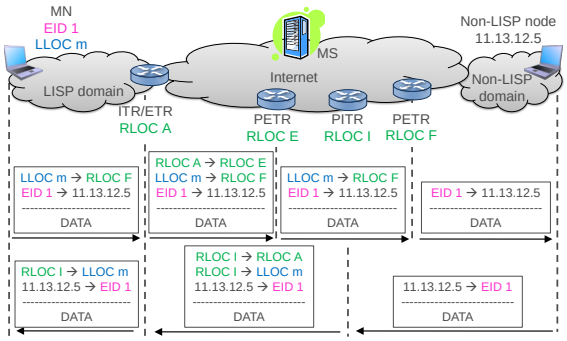
(a) Scenario 2: A MN in a non-LISP domain communicates with a SN in a LISP domain.



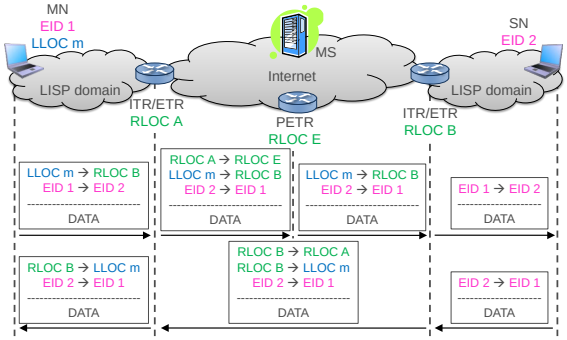
(b) Scenario 3: A MN in a non-LISP domain communicates with a MN in a LISP domain.



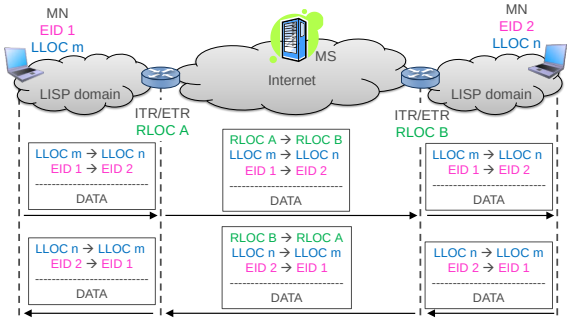
(c) Scenario 4: A MN in a non-LISP domain communicates with a non-LISP node.



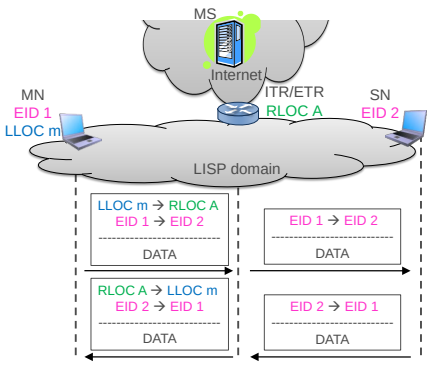
(d) Scenario 5: A MN in a LISP domain communicates with a non-LISP node.



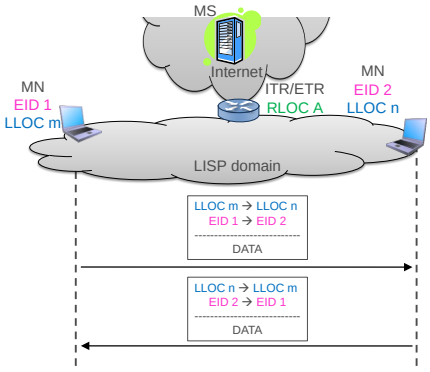
(e) Scenario 6: A MN in a LISP domain communicates with a SN in another LISP domain.



(f) Scenario 7: A MN in a LISP domain communicates with a MN in another LISP domain.



(g) Scenario 8: A MN in a LISP domain communicates with a SN in the same LISP domain.



(h) Scenario 9: A MN in a LISP domain communicates with a MN in the same LISP domain.

Fig. 2. Packet encapsulation and forwarding with LISP-MN when MNs are involved.

the SN's EID and encapsulates it towards the RLOC for that EID. The packet is sent to the ETR with that RLOC. This ETR decapsulates the packet and forwards it to the SN (see Figure 2(g)). If the SN is multihomed, it might have another ETR. The MN should ensure to choose the ETR that it has in common with the SN. In the reverse direction, the SN addresses a packet towards the MN's EID. The packet is forwarded to the default LISP gateway which first encapsulates it towards the LLOC of the MN and then to the RLOC for that LLOC. The packet is sent to the ETR with that RLOC which is possibly but not necessarily the same LISP gateway and the ETR decapsulates the outer decapsulation header. If the default LISP gateway is the same node as this ETR, the second encapsulation and the decapsulation can be omitted which is shown in the figure. Eventually the packet is forwarded to the MN. In the forward direction we observe triangle routing via the ETR of the LISP domain. In the reverse direction we witness either triangle routing via the default ITR or even "quadrangle" routing via the default ITR and the chosen ETR.

9) A MN in a LISP domain communicates with a MN in the same LISP domain: The MN addresses a packet towards the EID of the other MN in the same LISP domain. It encapsulates the packet towards the LLOC for that EID and sends it. The packet is carried directly to the corresponding MN (see Figure 2(h)). The reverse direction works likewise.

B. Disadvantages of LISP-MN

We summarize the observed disadvantages of LISP-MN. LISP-MN requires double encapsulation by ITRs or PITRs when they receive traffic from SNs or non-LISP nodes towards MNs in LISP domains. Hence, two mapping lookups are needed. As the (P)ITR cannot know a priori whether a second lookup returns another RLOC, it must perform two mapping lookups for all packets unless the first mapping lookup already returns a negative map-reply. We explain why this is an undesired feature. When ITRs or PITRs receive packets destined to EIDs or RLOCs for which they do not have mappings (including negative map-replies) in their local cache, these packets need to be queued (or discarded, or relayed via other nodes to the destination) until the requested mappings are retrieved from the mapping system. The basic LISP architecture requires only a single mapping lookup while LISP-MN mandates a second mapping lookup. This increases the time until queued packets can be sent. That raises the buffer overflow probability in ITRs and PITRs and extends the delay for first packets of a communication.

Under some conditions, PETRs and PITRs are needed as intermediate boxes for decapsulation and encapsulation (see scenarios 3 – 6). As PETRs and PITRs cause path stretch, their use should be avoided if possible. While the configured PETR of an ITR may be close to the ITR, the configured PETR of a MN may be far away from the MN and its corresponding node. Therefore, avoiding the use of PETRs is especially attractive for MNs.

When MNs communicate with SNs in the same LISP domain, they communicate via the ETR of the LISP domain. This leads to triangle routing (see scenario 8) and causes path stretch. Since the traffic is detoured within the same LISP domain, the absolute path stretch is probably not very large. Nevertheless, it is not a desired property.

When traffic is sent towards a MN residing in a LISP domain, it carries two encapsulation headers until it reaches the ETR of the destination LISP domain (see scenarios 3, 5 – 7). This may cause issues with maximum transfer units (MTUs) and lead to packet fragmentation. Therefore, it is also an undesired property.

IV. Improvements to LISP-MN

We suggest improvements to LISP-MN that avoid the disadvantages listed in Section III-B under many conditions. We introduce our improvements gradually to facilitate the adoption of a subset of the presented methods.

A. Filter Check and Direct Communication

A MN tunnels packets to its PETR to hide non-routable source addresses from the provider edge router which performs source address filtering. A recent study observed that in 31% of the investigated scenarios, ISPs did not block spoofed source addresses [9]. In such networks, tunneling traffic to the PETR is not necessary. This provides some optimization potential to reduce path stretch and latency.

We propose that when a MN roams into a new network, it should find out whether outgoing packets sent with its EID as source address are blocked by the ISP. To that end, the MN may ping the RLOC of its configured PETR without encapsulating packets. If the PETR responds, the provider edge router does not filter packets having the EID of the MN as source address. Therefore, the MN can communicate directly with non-LISP nodes. That means, when the MN later queries the mapping system with the destination address of an outgoing packet and the map-reply is negative, the MN can send the packet directly without tunneling it to its PETR. This improvement can reduce the path stretch for scenario 4 and scenario 5. However, this mechanism is only effective if the ISP of the network hosting the MN does not perform source address filtering.

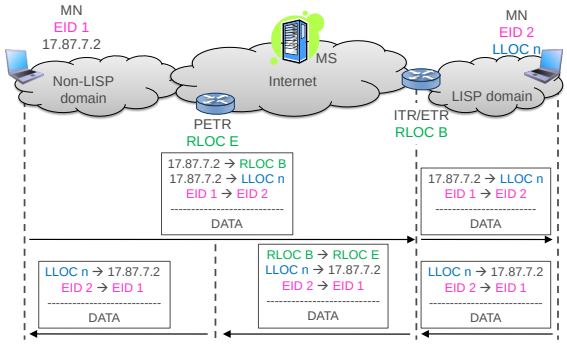
B. Location-Aware MNs

We propose that the MN should find out whether it is currently hosted in a LISP domain or in a non-LISP domain. To that end, it queries the mapping system with its assigned care-of-address. If a negative map-reply is returned, the MN is in a non-LISP domain and the care-of-address is an RLOC; otherwise, when an RLOC is returned, the MN is in a LISP domain, and the care-of-address is an LLOC. We call a MN having that information location-aware.

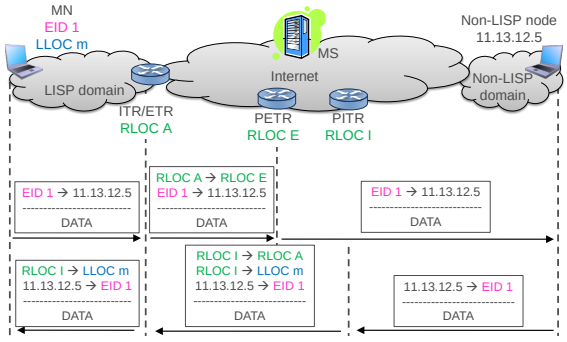
A location-aware MN in a non-LISP domain can avoid that its sent traffic is routed via a PITR if it communicates with a MN in a LISP domain (see scenario 3 in Figure 2(b)). To that end, the MN always performs a double mapping lookup and double encapsulation if possible just like an ITR or PITR. Figure 3(a) shows that traffic is then carried directly from the

MN in the non-LISP domain to the ETR of the destination LISP domain so that triangle routing via a PITR is avoided.

Furthermore, a location-aware MN in a LISP domain can avoid that its sent traffic is routed over the PETR of the MN if it communicates with a non-LISP node (see scenario 5 in Figure 2(d)) although the ISP of the LISP domain performs source address filtering. To that end, the MN sends packets directly instead of encapsulating them to its PETR when the mapping lookup for the destination address of a packet returns a negative map-reply. Figure 3(b) shows that the ITR then tunnels the traffic to its PETR which then forwards it directly to the non-LISP destination node. This avoids triangle routing via the PETR of the MN, which would otherwise most likely add a significant path stretch. The remaining path stretch through triangle routing over the PETR is likely to be small because the involved PETR is located near the ITR.



(a) Scenario 3: A MN in a non-LISP domain communicates with a MN in a LISP domain. Comparison to Figure 2(b): MNs in non-LISP domains perform double mapping lookups and double encapsulation if needed.



(b) Scenario 5: A MN in a LISP domain communicates with a non-LISP node. Comparison to Figure 2(d): MNs in LISP domains send packets destined to non-LISP nodes without encapsulation.

Fig. 3. Location-aware MNs avoid the use of a PITR and a second PETR.

C. Locator Types

In Section II-C, we introduced LLOCs and RLOCs as two different locator types for the sake of simpler readability, but LISP-MN does not take advantage of this differentiation. We now suggest that the locator type is stored as accompanying information together with the mapping in the mapping system, returned in map-replies, and stored in map-caches.

The globally reachable IP addresses of LISP gateways are registered in the mapping system as RLOCs for EIDs and care-of-addresses inside that LISP domain. When a MN roams into a new network, it obtains a new care-of-address (e.g., by DHCP) under which it is then reachable. It registers this address in the mapping system as a locator for its EID. We propose that it also stores the locator type as "LLOC" in the mapping system. Then, it queries the mapping system with that address. If a negative map-reply is returned, the MN is in a non-LISP domain and the care-of-address is an RLOC. Therefore, the MN changes the locator type for its EID-to-locator mapping in the mapping system to "RLOC". If RLOCs are returned for the requested care-of-address, the MN is in a LISP domain and the care-of-address is in fact an LLOC so that nothing needs to be changed.

ITRs, PITRs, and MNs take advantage of the locator type information in the mappings. If they encapsulate packets towards RLOCs, they can send them immediately without querying the mapping system again. This avoids unnecessary double mapping lookups by ITRs, PITRs, and MNs when the destination address of a packet is not a MN in a LISP domain.

A MN-bit was proposed in [6, Sect. 8] in the context of multicast. It indicates in the mapping whether the node for which the locator is returned is a MN. This is similar to the locator type but not the same because MNs in non-LISP domains do not have LLOCs. To save extra bits in the mappings, the MN-bit may be used instead of the locator type. This avoids double lookups by ITRs, PITRs, and MNs when the destination of a packet is a SN, but if the destination is a MN in a non-LISP domain, the second unnecessary lookup cannot be avoided.

D. Local Mapping System

We propose a local mapping system that helps a MN to send traffic to a SN in the same LISP domain without triangle routing over the SN's ETR. The proposal requires location-aware MNs and locator types (see Sections IV-B and IV-C).

Each LISP gateway knows the routable EIDs of all SNs in its domain, e.g., by configuration. Furthermore, it keeps a local EID-to-LLOC table for all MNs in its domain. It returns these mappings when queried by registered MNs. This constitutes the local mapping system. We explain how the EID-to-LLOC table is populated. If a MN roams into a LISP domain, it receives an LLOC, queries the global mapping system with that LLOC, and records the returned RLOCs as configured LISP gateways. Then, it registers its EID together with its obtained LLOC at all configured LISP gateways. The LISP gateways use soft state to store this information in their EID-to-LLOC tables so that stale information is purged after short time when MNs have left the LISP domain without logging off properly.

When a MN in a LISP domain wants to send an outgoing packet, it first queries one of its configured LISP gateways with the destination address of the packet for an EID-to-LLOC mapping. The gateway returns one of the following three responses: (1) the EID belongs to a SN in the same domain, (2)

the EID belongs to a MN in the same domain and the LLOC is delivered, or (3) a node with the requested destination address is not in the same domain. In the first case, the MN sends the packet without encapsulation. In the second case, the MN encapsulates the packet towards that LLOC and sends it. In the third case, the MN queries the global mapping system with the destination address of the packet, encapsulates the packet if the map-reply was positive, and sends the packet. In this third case, the lookup latency of first packets is increased by the query to the LISP gateway. However, this adds only little delay because the LISP gateway is near and extra delay is not very critical in this particular situation as the packet waits at its source node. As a result of this optimization, the MN sends traffic directly to another SN in the same domain and triangle routing via the SN's ETR is avoided (see Figure 2(g)).

Also the behavior of ITRs should be changed. Before querying the global mapping system, they should query their local mapping system when a packet causing a cache miss is to be sent. If the map-reply from the local mapping system is positive, the packet is encapsulated with the returned LLOC and sent. This avoids potential quadrangle routing when a SN sends traffic to a MN in the same LISP domain (see Section III-A.8).

E. Avoiding Double Encapsulation Headers

Packets addressed to MNs in LISP domains have two encapsulation headers. The outer encapsulation header carries the destination RLOC which is used for forwarding in the Internet except for the destination domain. The inner encapsulation header carries the destination LLOC which is used for forwarding in the destination domain. We propose a mechanism to avoid this double encapsulation. In the Internet except for the destination domain, packets addressed to MNs in LISP domains are encapsulated with the destination RLOC in the destination field. In the destination domain, they are encapsulated with the destination LLOC in the destination field. Our proposal prerequisites the local mapping system presented in Section IV-D.

We change the behavior of the MN slightly. If it queries the global mapping system and receives an LLOC, it queries the global mapping system again with the obtained LLOC to get its RLOC. Then it encapsulates the packet only to the RLOC.

When an ITR receives a packet, it performs the modified steps according to the behavior described in Algorithm 1. PITRs work in the same way but cannot consult a local mapping system and check for registered LLOCs.

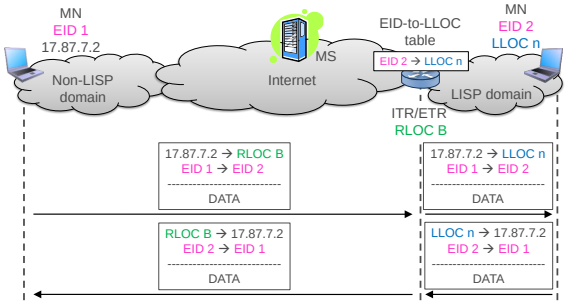
When an ETR receives a packet destined to itself, it decapsulates it. Then, it queries the local mapping system for an LLOC of the destination address. If an LLOC is returned, the ETR encapsulates the packet towards that LLOC. Eventually, the ETR forwards the packet into its LISP domain.

These changes have no impact on the encapsulation and forwarding structure in scenarios 1, 2, 4, 8, and 9 which do not suffer from double encapsulation headers. Scenarios 3, 5, 6, and 7 still suffer from double encapsulation either in the forward or reverse direction even if the changes of the previous sections are applied. Figures 4(a)–4(d) show the encapsulation and forwarding structure with the proposed changes of this section. We compare the structure with and without our new mechanism and explain how double encapsulation is avoided.

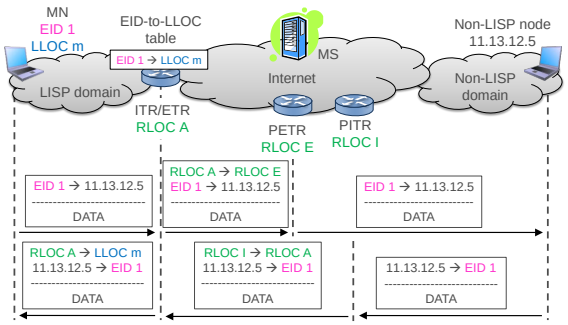
We consider scenario 3 and compare Figures 4(a) and 3(a). On the forward path, double encapsulation is avoided by adding the encapsulation header with the destination LLOC only at the ETR. On the reverse path, an additional encapsulation header is avoided because the ITR substitutes the source LLOC by its own RLOC and avoids thereby tunneling packets to its PETR. We consider scenario 5 and compare Figures 4(b) and 3(b). On the reverse path, double encapsulation is avoided by adding the encapsulation header with the destination LLOC only at the ETR instead at the PITR. We consider scenario 6 and compare Figures 4(c) and 2(e). On the forward path, an additional encapsulation header is avoided because the ITR substitutes the source LLOC by its own RLOC and avoids thereby tunneling packets to its PETR. On the reverse path, double encapsulation is avoided by adding the encapsulation header with the destination LLOC only at the ETR instead at ITR. We consider scenario 7 and compare Figures 4(d) and 2(f). On the forward and on the reverse path, an additional encapsulation header is avoided because the ITR substitutes the source LLOC by its own RLOC and avoids thereby tunneling packets towards its PETR. Moreover, double encapsulation is avoided by adding the encapsulation header

Algorithm 1 Modified ITR forwarding behavior.

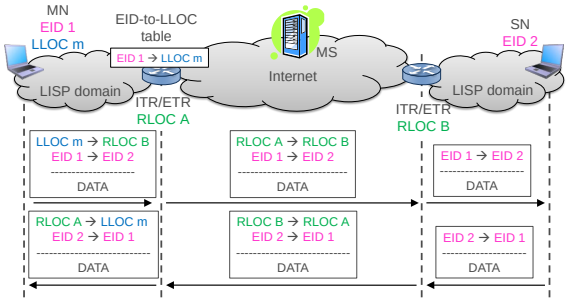
```
Input: packet with srcAddress and destAddress
Loc1 ← queryLocalMS(destAddress)
if Loc1 ≠ Null then
  //Loc1 is LLOC
  //similar to reverse direction in Fig. 2(g)
  encapsulate packet to Loc1, send it
else
  Loc2 ← queryGlobalMS(destAddress)
  if Loc2 == Null then
    //neg. map-reply ⇒ destAddress is routable
    if srcAddress is LLOC in ITR's domain then
      //see reverse direction in Fig. 4(a), forward
      //direction in Fig. 4(c), and Fig. 4(d)
      substitute srcAddress with ITR's RLOC, send packet
    else
      //srcAddress is EID, see Fig. 4(b)
      encapsulate packet to PETR, send it
    end if
  else if typeof(Loc2) == LLOC then
    //see reverse direction in Fig. 4(c)
    Loc3 ← queryGlobalMS(Loc2)
    encapsulate packet to Loc3, send it
  else
    //Loc2 is already RLOC, see Fig. 1(a)
    encapsulate packet to Loc2, send it
  end if
end if
```



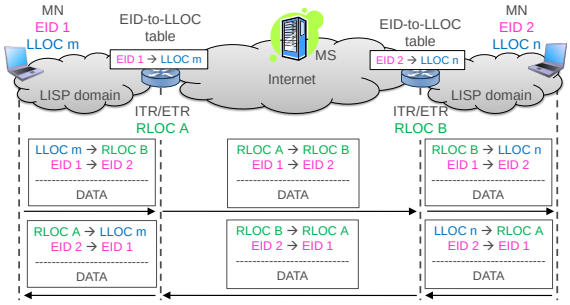
(a) Scenario 3: A MN in a non-LISP domain communicates with a MN in a LISP domain.



(b) Scenario 5: A MN in a LISP domain communicates with a non-LISP node.



(c) Scenario 6: A MN in a LISP domain communicates with a SN in another LISP domain.



(d) Scenario 7: A MN in a LISP domain communicates with a MN in another LISP domain.

Fig. 4. Packet encapsulation and forwarding when double encapsulation is avoided.

with the destination LLOC only at the ETR instead at the ITR.

We have shown that double encapsulation can be avoided for communication with MNs. In addition, the use of PETRs is minimized and thereby path stretch is reduced. However, our proposal has some disadvantages. When ITRs change the source address of a packet, this corresponds to combined decapsulation and encapsulation which might be difficult to implement. ETRs need to add encapsulation headers whereas without our additions they just used to decapsulate traffic. Whenever the global mapping system is queried and returns an LLOC, the LLOC is no longer added as destination address to packets which seems inefficient. This unnecessary indirection in the mapping system may be avoided by registering only RLOCs for both SNs and MNs.

V. Conclusion

We have reviewed the operation of the Locator/Identifier Separation Protocol (LISP) including interworking techniques. We made the encapsulation and forwarding structure of the currently discussed LISP Mobile Node architecture [6] (LISP-MN) explicit and pointed out its shortcomings. Those are: unnecessary mapping lookups, path stretch through routing over PITRs and PETRs, as well as double encapsulation headers. We proposed improvements which can avoid these disadvantages under many conditions. These changes can be introduced gradually. Which of them are applied to LISP-MN is a tradeoff between implementation complexity and optimality of the resulting routing. We believe that the improvements in Sections IV-A – IV-C are simple and relevant in practice, the improvements in Section IV-D are interesting for LISP-domains hosting a large number of mobile nodes, while the ideas in Section IV-E may be only of academic interest.

Acknowledgements

The authors would like to thank Dino Farinacci, David Meyer, Phuoc Tran-Gia, Steve Uhlig, and Vince Fuller for insightful comments and in particular Darrel Lewis for his thorough feedback regarding the illustration and analysis of LISP-MN.

References

[1] X. Meng, Z. Xu, B. Zhang, G. Huston, S. Lu, and L. Zhang, "IPv4 Address Allocation and the BGP Routing Table Evolution," *ACM SIGCOMM Computer Communications Review*, vol. 35, no. 1, pp. 71 – 80, Jan. 2005.

[2] D. Meyer, L. Zhang, and K. Fall, "RFC4984: Report from the IAB Workshop on Routing and Addressing," Sep. 2007.

[3] B. Quoitin, L. Iannone, C. de Launois, and O. Bonaventure, "Evaluating the Benefits of the Locator/Identifier Separation," in *ACM International Workshop on Mobility in the Evolving Internet Architecture (MobiArch)*, Kyoto, Japan, Aug. 2007.

[4] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, "Locator/ID Separation Protocol (LISP)," http://tools.ietf.org/html/draft-ietf-lisp-06, Jan. 2010.

[5] D. Lewis, D. Meyer, D. Farinacci, and V. Fuller, "Interworking LISP with IPv4 and IPv6," draft-ietf-lisp-interworking-01, Feb. 2010.

[6] D. Farinacci, V. Fuller, D. Lewis, and D. Meyer, "LISP Mobile Node," http://tools.ietf.org/html/draft-meyer-lisp-mn-01, Feb. 2010.

[7] C. Perkins, "RFC3344: IP Mobility Support for IP4," Aug. 2002.

[8] L. Iannone and O. Bonaventure, "On the Cost of Caching Locator/ID Mappings," in *ACM Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, Dec. 2007.

[9] R. Beverly, A. Berger, Y. Hyun, and k claffy, "Understanding the Efficacy of Deployed Internet Source Address Validation Filtering," in *ACM Internet Measurements Conference (IMC)*, Nov. 2009.