

NAT Traversal for LISP Mobile Node

Dominik Klein, Matthias Hartmann
University of Würzburg, Germany
{dominik.klein,matthias.hartmann}
@uni-wuerzburg.de

Michael Menth
University of Tübingen, Germany
menth@uni-tuebingen.de

ABSTRACT

The *Locator/ID Separation Protocol* (LISP) is currently being developed and standardized in the IETF aiming to solve the Internet's routing scaling problem. It separates global routing in the Internet from local routing in end-user networks (so-called LISP-domains). It also provides additional benefits like simplified multihoming or the avoidance of network renumbering. The basic LISP architecture does not support mobility. Recently, the mobility extension *LISP Mobile Node* (LISP-MN) was presented. It describes a mechanism that enables LISP mobile nodes to roam into LISP and non-LISP networks while being reachable under the same identifier address. Currently, LISP-MN does not support networks that use network address translation (NAT). In this paper, we present a NAT traversal mechanism for LISP mobile nodes and a slight adaptation which is also applicable to stationary LISP domains behind a NAT.

1. INTRODUCTION

The Internet has become the nervous system of most of today's business and private communications, and the number of systems and networks connected to the Internet is rising at an increasing pace. For reliable interconnections, many networks have multiple points of attachment to several ISPs. This requires provider-independent IP addresses, which must be routable in the default-free zone (DFZ) in the Internet, and add additional entries in the BGP routing tables. The growing size of these routing tables will sooner or later cause scalability and flexibility problems.

To solve these issues, several new naming, addressing, and routing schemes are currently under discussion in the IETF and IRTF [9]. Most of the approaches are based on the locator / identifier (Loc/ID) split [11]. It uses special identifier addresses (IDs) to denote end-hosts or services. These IDs are not routable in the DFZ. Instead, a routable locator address (Loc) is added to packets to send them over the Internet. The current Loc for an ID is returned by a special mapping service that stores an ID-to-Loc-mapping for each ID in the Internet. In this way, Loc/ID split decouples the combined identification and location function of today's IP addresses. This provides benefits like stable provider-independent addressing or multihoming, without increasing the size of BGP routing tables.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACM ReArch 2010, November 30, 2010, Philadelphia, USA.
Copyright 2010 ACM 978-1-4503-0469-6/10/11 ...\$5.00.

The Loc/ID split approach that currently draws most attention is the *Locator/ID Separation Protocol* (LISP) [2]. LISP separates local routing in edge networks from global routing in the DFZ. Its underlying philosophy is to reuse as much as possible of current technology and protocols. Neither end-hosts, nor routers in the DFZ need to be upgraded. Only special LISP gateways are required to encapsulate packets addressed to endpoint identifiers (EIDs) with the appropriate routing locators (RLOCs).

The basic LISP architecture does not support mobility of end-hosts. Recently, the extension *LISP Mobile Node* (LISP-MN) [1] was presented. It describes a mechanism that enables mobile nodes (MNs) to roam into LISP and non-LISP networks while being reachable under the same EID-address. Each MN acts as its own LISP gateway, and uses the globally routable address it obtained in the current network as RLOC. If a MN connects to a network that uses NAT, it only receives a private address that is not routable in the DFZ. This address is unsuitable as RLOC and thus, LISP-MN does not work with NAT boxes. In the following, we assume that a NAT box not only translates IP addresses but also ports (NAPT).

In this paper we present an extension to LISP-MN which allows NAT traversal for MNs by utilizing special NAT traversal routers (NTRs). While this mechanism is primarily designed for MNs, a slight adaptation also enables several LISP domains behind a NAT.

The paper is structured as follows. Section 2 explains the basic LISP mechanisms and the extension for supporting mobility. In Section 3 we explain our approach for supporting MNs behind NAT gateways and how it is also applicable to stationary LISP domains. Section 4 then presents a modification of our NAT traversal mechanism to increase its robustness and Section 5 describes related work and explains how it differs from our work. Finally, Section 6 concludes this work.

2. LISP

In this section, we first give an overview of the basic operation of LISP and introduce its interworking techniques used to enable communication with the non-LISP Internet. We describe mechanisms, which are required to support LISP gateways behind a NAT. Then, we briefly describe the mobility extension of LISP and explain, why an additional traversal mechanism is needed to support MNs behind a NAT.

2.1 Basic Operation

The *Locator/ID Separation Protocol* (LISP) [2] is an implementation of the Loc/ID split. The IP address range is divided into two different subsets. Endpoint identifiers (EIDs) identify end-hosts on a global scale and are used to forward packets locally inside LISP domains. LISP domains are edge networks that are connected via LISP gateways to the core of the Internet, where globally routable

addresses are used to forward packets. The globally routable addresses of LISP gateways are called routing locators (RLOCs).

The communication between LISP nodes inside the same LISP domain does not change due to LISP. However, the communication between LISP nodes in different domains requires tunneling between the different LISP gateways. The gateways either act as ingress tunnel router (ITR) or as egress tunnel router (ETR). ITRs tunnel packets to other LISP gateways which then act as ETRs. Figure 1 shows a packet flow sequence for the communication between two LISP clients located in different LISP domains. ITR A receives packets addressed to EID 2 from an end-host in its own LISP domain. It keeps the inner header (IH) untouched and adds a UDP header addressed to the default LISP port 4341 and an outer LISP header (OH) with its RLOC (RLOC A) as source and RLOC B as destination address so that the packets are globally routable.

This procedure requires a mapping lookup to learn the appropriate RLOC (RLOC B) for the destination EID (EID 2). LISP does not mandate a specific mapping service but instead introduces map servers (MS) and map resolvers [3]. These two entities form an interface which facilitates the operation of LISP with different mapping systems. ETRs register the EID-to-RLOC mapping for all attached LISP nodes at their associated map server on the default LISP signaling port 4342. The EID-to-RLOC registration process and security mechanisms to protect the registration against for example spoofing attacks are described in Section 4.2 of [3].

Map servers listen on port 4342 and once the map server receives the registration, it distributes the mapping information within the mapping service so that map resolvers find the authoritative map server for a specific EID. ITRs query map resolvers for the RLOC of a specific EID. The map resolver initiates a map-request which is forwarded via the mapping service to the authoritative map server. Again, port 4342 is used for the map-request message. The map server responds with a map-reply message which contains the valid locator set for the queried EID. Map resolvers and map servers can either be deployed in separate nodes or inside ITRs and ETRs. The current most prominent mapping system is *LISP Alternative Topology* (LISP+ALT) [4].

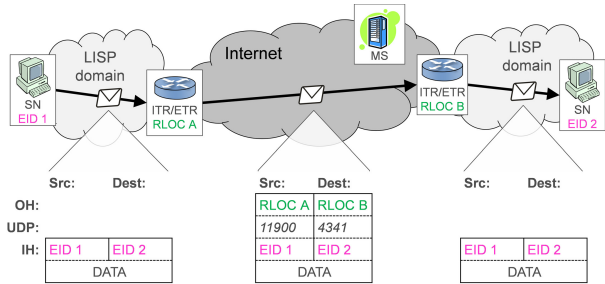


Figure 1: Packet flow sequence with LISP.

2.2 LISP Interworking

To communicate with nodes in the non-LISP Internet, additional interworking mechanisms are required [7]. In the following, we describe two different mechanisms proposed in the LISP interworking draft (LISP-IW). The first one integrates NAT functionality inside LISP gateways and the second one relies on special proxy LISP gateways.

2.2.1 LISP Gateways with NAT Functionality

The LISP-NAT interworking mechanism integrates NAT functionality inside LISP gateways to enable communications from LISP domains to non-LISP domains. LISP-NAT must not be mistaken

for a NAT traversal mechanism for LISP or LISP mobile node and hence, we briefly describe the LISP-NAT interworking mechanism to avoid any misunderstanding in this context.

LISP gateways get a pool of globally routable RLOC addresses and use a free RLOC address from this address pool to translate the locally routable source EID of outgoing packets (see Figure 2(a)). Due to the globally routable source address, return packets can be delivered directly to the source LISP gateway. The gateway has sufficient state to translate the destination RLOC address of return packets back to the EID of the source LISP client (see Figure 2(b)).

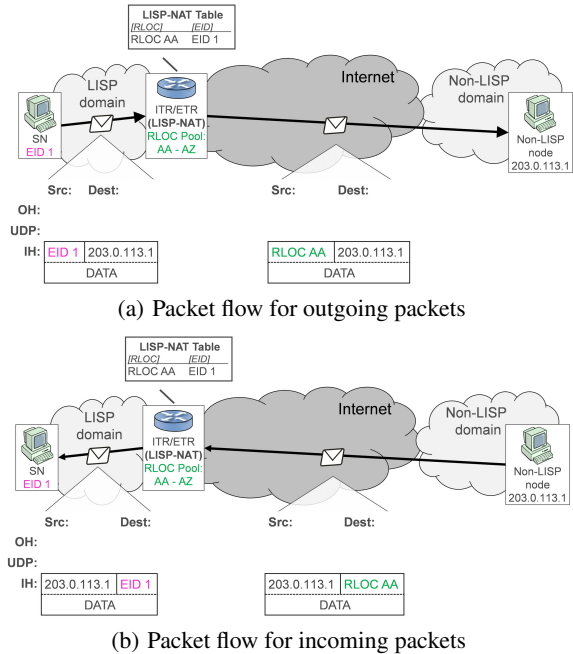


Figure 2: LISP-NAT packet flow sequence.

With LISP-NAT, LISP domains have full control over the interworking mechanism because additional boxes are not required outside the LISP domain. However, the number of available RLOC addresses in the pool limits the number of LISP clients in a LISP domain which can communicate at a time with non-LISP nodes. Also the communication initiation is only possible from LISP domains to non-LISP domains. LISP-NAT solely offers a mechanism for interworking and must not be mistaken for a NAT traversal mechanism.

2.2.2 Proxy LISP Gateways

Normal IP nodes usually resolve the DNS name of a LISP node into an EID and use it as destination address. However, EIDs are not globally routable and thus the border router of the non-LISP domain discards the packets because of missing forwarding entries for EIDs. To solve this problem, LISP-IW proposes additional boxes called proxy-ITRs (PITRs), which are located outside edge domains and advertise highly aggregated EID prefixes into BGP. This way, packets addressed to EIDs become globally routable and are forwarded to one of the PITRs. PITRs perform the same traffic processing as ordinary ITRs, i.e., they query the mapping system for an RLOC of the destination EID and encapsulate packets towards the returned RLOC (see Figure 3(a)).

In the reverse direction, LISP packets destined to non-LISP nodes are not encapsulated by ITRs and the EID remains in the source address field of outgoing packets. Since the EID is not part of the upstream providers address range, such packets might be dropped

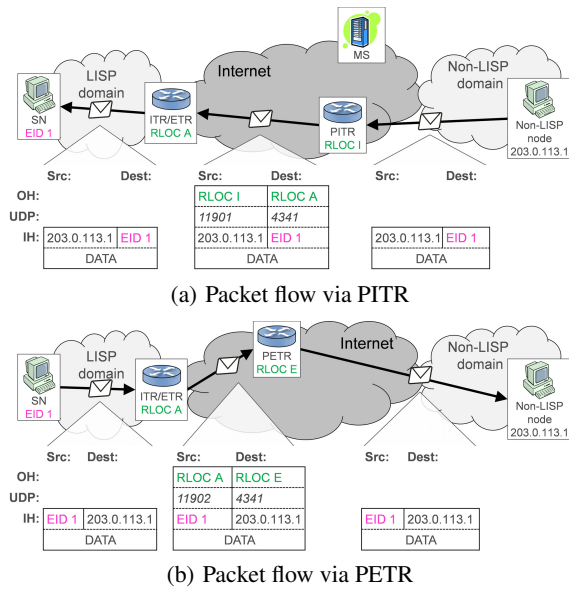


Figure 3: LISP interworking packet flow sequence.

when the provider does source address filtering to ensure that outgoing packets carry only addresses from its own address range. In this case, LISP uses proxy-ETRs (PETERs). LISP gateways encapsulate packets destined to non-LISP nodes and send them to a pre-configured PETER outside their own domain (see Figure 3(b)). The PETER decapsulates the packet and sends it to the destination node in the non-LISP Internet. This way, LISP bypasses the source address filtering of upstream providers. PETERs can also be used to connect LISP domains which use a different IP version than their upstream provider.

In contrast to LISP-NAT, proxy LISP gateways enable communication initiation from non-LISP domains to LISP domains. Hence, we assume in the following that proxy LISP gateways are used for interworking between LISP and non-LISP domains.

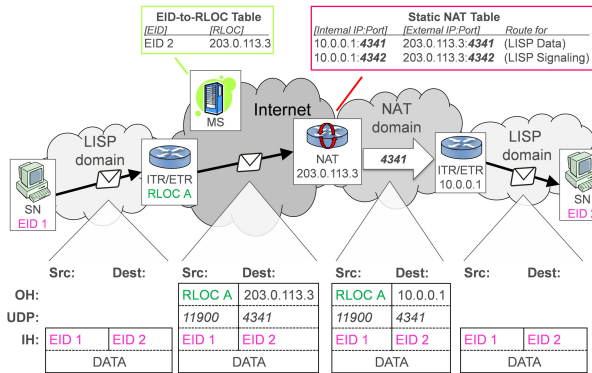


Figure 4: LISP gateway behind a NAT.

2.3 LISP Gateways behind a NAT

The ETR functionality of LISP gateways is by definition addressable by globally routable RLOCs. However, during the early deployment phase, it may be necessary to deploy LISP gateways behind a NAT box. This requires that the external address of the NAT device is registered in the mapping service, and that two static routes are installed in the NAT.

ITRs send LISP data messages over UDP/IP tunnels, and always use port 4341 as destination port and a random source port. Outgoing LISP data messages can be sent through the NAT but response packets are discarded at the NAT as the destination port does not match the previously used source port. Thus, it is necessary to install a static route in the NAT that forwards the external port 4341 to the internal port 4341 of the ETR. All incoming packets that arrive on port 4341 at the NAT are then statically forwarded to port 4341 at the ETR (see Figure 4). The static route enables the ETR behind a NAT to receive LISP data traffic. However, only one static route can be installed for a specific port at a NAT gateway and hence, only one ETR can be deployed behind a NAT by means of a static route.

When ETRs behind a NAT also implement map server functionality, they must be able to receive map-requests, which arrive at port 4342. Hence, this requires an additional static route (see Figure 4).

Theoretically, static routes could support the deployment of an ETR behind a NAT, but it would be limited to a single ETR.

2.4 LISP Mobile Node

LISP Mobile Node (LISP-MN) [1] introduces mobility and allows LISP nodes to roam into other domains. Mobile nodes (MNs) possess an upgraded stack and act as light-weight LISP domains. They implement ITR/ETR functionality and are configured with the address of a map server that controls the EID-to-RLOC mappings for the MN. MNs register their currently valid locator at their configured map server and refresh this information by sending periodic map register messages.

MNs also implement map resolver functionality and send signaling traffic to their configured map server without encapsulation. In contrast, data traffic is always encapsulated. The MN thus requires a PETER for communications with non-LISP nodes. Therefore, the map server also acts as PETER if the MN communicates with non-LISP nodes.

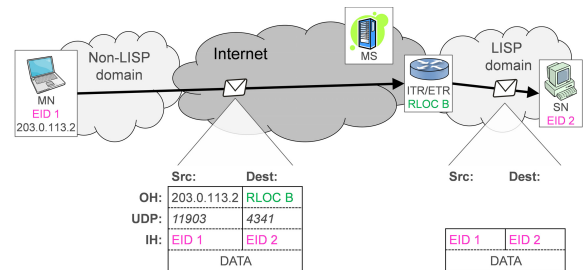


Figure 5: MN in non-LISP domain.

Figure 5 shows an example where a MN roams into a non-LISP domain and initiates a communication with a remote LISP stationary node (SN) in a LISP domain. The MN receives the globally routable non-LISP care-of-address 203.0.113.2 upon roaming into the new domain. It registers this address as RLOC along with its own EID 1 at its associated map server. This address can be obtained from the map server by ITR B as destination locator upon sending a packet to the MN.

A companion document [10] to this work describes further communication examples and gives a detailed description of LISP-MN.

2.5 Mobile Nodes behind a NAT

When mobile nodes are behind a NAT and implement only the proposed LISP-MN architecture, they can send traffic to other nodes, but cannot receive traffic from them. When roaming into the NAT domain, they receive a private care-of-address, and register it at

their associated map server. When sending traffic, the MN queries the mapping system on destination port 4342 without LISP encapsulation so that the map-reply is able to return to the MN. The MN encapsulates data packets towards the obtained RLOC using the care-of-address as source address. The address is modified in the headers of outgoing packets when crossing the NAT. When an ITR tries to send a packet to a MN behind a NAT, the mapping system returns a private address as RLOC so that packets cannot be forwarded correctly after encapsulation and get dropped. Thus, the packets never reach the MN behind the NAT. This issue is described in the LISP-MN draft (see Section 9.1 in [1]), but up to now, there is no publicly available document describing a solution for that problem. In the next section we present a NAT traversal mechanism that solves the problem.

3. NAT TRAVERSAL FOR LISP-MN

We propose a NAT traversal mechanism for LISP MNs behind NATs so that they can receive traffic. We first give an overview of the concept, then we explain how MNs behind a NAT register at NAT traversal routers (NTRs), and how NTRs relay packets destined to registered MNs. We then describe the applicability of our NAT traversal mechanism to stationary LISP domains behind a NAT. Finally, we discuss some deployment considerations and security concerns.

3.1 Overview

The NAT traversal functionality is collocated with the same box that also implements the map server and the PETR for the MN. In the remainder of this document, we call a modified map server that implements the NAT traversal mechanism a NAT Traversal Router (NTR). When a MN roams into a network, it obtains a care-of-address and registers it as RLOC for its EID at its preconfigured NTR. If the NTR recognizes that the MN is behind a NAT, the IP address of the NTR is registered as RLOC for the EID of the MN in the mapping system. Thus, when traffic is sent to MNs behind a NAT, (P)ITRs tunnel it to NTRs instead of to the care-of-address of the MNs. The NTR has sufficient information to relay that traffic to the MNs and the traffic traverses the NAT due to the context established during the registration. This essentially constitutes a tunnel between the NTR and the MN which is used to bypass the NAT gateway.

Due to the tunnel between the NTR and the MN, our NAT traversal mechanism works with every type of NAT, even with symmetric NATs, and is able to cope with several layers of NAT gateways.

3.2 Registration Process

When a MN roams into a network, it receives a care-of-address from the local DHCP service and sends a map-register message to the map server using destination port 4342 without any LISP encapsulation. In contrast to the current behavior in LISP-MN, our NAT traversal proposal requires that source port 4341 is used (the reason is explained later in Section 3.3). The collocated NTR compares the reported care-of-address with the source address of the register message. If they are the same, the MN is not behind a NAT and the address is registered as RLOC for the EID of the MN in the mapping system. If the two addresses differ, the newly proposed NAT traversal concept for MNs behind NATs is used. We explain it using the packet flow sequence in Figure 6. A MN with EID 1 has roamed into a private network and obtained the care-of-address 10.0.0.1. It sends a register message containing this address to port 4342 at the NTR with RLOC N. The intermediate NAT gateway translates the source IP:port 10.0.0.1:4341 into 203.0.113.3:11341 and stores this as context for outgoing packets with destination IP:port RLOC N:4342. The NTR detects that the

care-of-address 10.0.0.1 differs from the source address of the register message (203.0.113.3) and, therefore, it stores its own IP address (RLOC N) as RLOC for EID 1 in the mapping system. In addition, the NTR records the source address and port of the register message (203.0.113.3:11341) with the EID (EID 1) in an EID-to-IP:port table. The NTR requires this IP:port to relay packets to the MN behind the NAT. The private address (10.0.0.1) is not stored at the NTR and only used by the NTR to detect whether the source address of the register message differs from the registered care-of-address.

To make the mapping system robust against stale information, an expiration timer is associated with registered EID-to-RLOC mappings. The same may be applied to the EID-to-IP:port table inside the NTR. However, in this context, the expiration timer should be set to small value so that the established context in the NAT gateway is also refreshed in time.

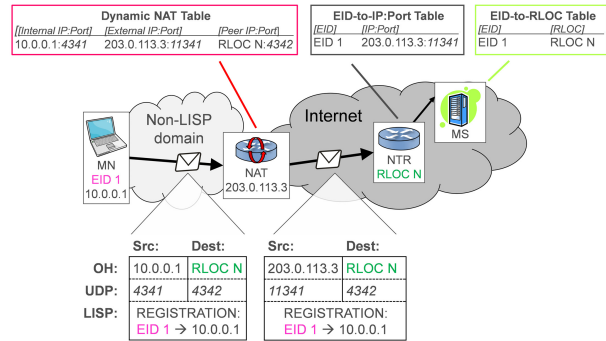


Figure 6: Registration process.

3.3 Relaying Process

When traffic is sent to MNs behind a NAT, (P)ITRs tunnel it to the NTR at which the MNs have registered. This is depicted in Figure 7. An NTR relays such traffic as follows. It strips off the LISP and UDP header, uses the destination EID (EID 1) in the IH of the packet to look up the IP:port 203.0.113.3:11341 in the EID-to-IP:port table, and encapsulates the packets to this IP:port combination using its own IP address and port 4342 as source IP:port combination (RLOC N:4342). The NAT gateway recognizes the destination IP:port and translates it accordingly which is 10.0.0.1:4341 in our example. Eventually, the translated packet reaches the MN on the correct port 4341 for incoming LISP-encapsulated traffic. The correct port number is achieved by requiring MNs to send map-register messages to the map server using source port 4341. Regarding the behavior of a MN, this constitutes the only difference between our proposal and the original LISP MN architecture.

Choosing another source port for the registration process would require that the LISP MN has to listen on that port for LISP data traffic in case it is behind a NAT. By using the LISP data port 4341, we avoid this issue and the MN has not to be aware of the NAT.

3.4 Applicability to Stationary LISP Domains

A LISP gateway behind a NAT can be made reachable from the outside by a static route from the NAT gateway to the LISP gateway (see Section 2.3). However, this works only for a single LISP gateway per NAT gateway. A slight adaptation of our proposed NAT traversal mechanisms allows to operate a large number of LISP gateways behind a NAT which might be a significant advantage.

The EID ranges for stationary LISP nodes are configured with the LISP gateway. The LISP gateway registers all configured EID ranges with the NTR and the NTR registers its own RLOC in the map server for these EID ranges. As a consequence, the station-

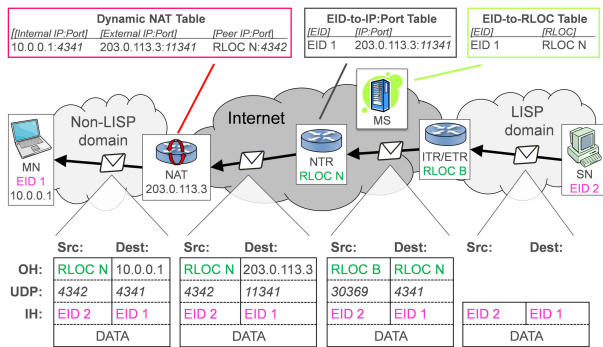


Figure 7: Incoming flow for a MN behind a NAT.

any LISP nodes in the LISP domain behind the NAT are reachable from the outside through the NTR which forwards incoming traffic to the respective LISP gateway. For LISP domains receiving high data rates, care must be taken since all incoming traffic is relayed over the NTR. The number of supportable users in these LISP domains is limited only by the number of simultaneous outgoing connections that can be supported by the NAT device.

3.5 Deployment Considerations

In the description of our NAT traversal mechanism, we assumed that the NTR is collocated with the map server of the MN. However, it is also possible to run the NTR functionality in a separate box. This may be important when the map server provider does not want to relay high data rates. The NTR in this case relays signaling traffic between the MN and its map server and data traffic between communication partners of the MN and the MN itself. The NTR infrastructure is then completely decoupled from the map server infrastructure.

3.6 Security Concerns

The presented NAT traversal allows nodes in the Internet to contact MNs behind a NAT gateway which is the intention of the proposal. If the NAT is used as part of a firewall, external nodes can easily circumvent this security feature and contact MNs. This is a general concern of all NAT traversal mechanisms. Moreover, any type of traffic can reach the MN behind a NAT/firewall because of tunneling. This may be improved by making the NAT/firewall aware of this mechanism using deep packet inspection for incoming LISP traffic.

4. MODIFIED NAT TRAVERSAL

In this section, we introduce a modified version of the NAT traversal mechanism that requires modifications to the MN host stack. We first give a short motivation why this may be interesting and then explain the differences to the basic mechanism.

4.1 Motivation

The NAT traversal mechanism described in Section 3 added new functionality to map servers and did not introduce additional complexity at MNs. While this may be important for low power nodes, a more sophisticated version of a NAT traversal mechanism may be interesting for MNs with sufficient resources. The mechanism described in the following requires updates to the MN stack but avoids sending registration messages that carry a private IP address. This avoids a possible problem with NAT gateways that utilize an application layer gateway (ALG) to modify private IP addresses inside the payload of outgoing packets. Under these conditions, the previously described NTR would not be able to determine whether the MN is behind a NAT.

4.2 Modified Registration Process

In the modified mechanism, an upgraded MN first sends a message to its preconfigured NTR and asks for its external address. The NTR responds with the address seen in the source address field of the message from the upgraded MN. The MN receives the response message and compares the returned external address with its own care-of-address. If the external address matches its care-of-address, the MN infers that it is not behind a NAT and contacts its preconfigured map server to register its care-of-address as current locator. Otherwise, both addresses differ and the MN concludes that it is behind a NAT and uses the proposed NAT traversal.

This improved version of the NAT traversal mechanism avoids problems with NAT devices that use an ALG to modify private addresses inside the packet payload but requires changes to the MN stack. However, the implementations of the MN stack are not yet deployed and hence the proposed modification may be an interesting tradeoff between complexity and increased robustness of our proposed NAT traversal mechanism.

5. RELATED WORK

We briefly review existing work about NAT traversal mechanisms in general and explain why they do not work with LISP-MN. Then we sketch other mobility extension for locator/identifier separation protocols, point out the differences to our work, and finally, we describe other protocols which use similar techniques.

5.1 Existing NAT Traversal Mechanisms

With Session Traversal Utilities for NAT (STUN) [13], an application A behind a NAT uses a so-called STUN server before the NAT to learn its external IP:port combination. To that end, application A sends a request to the STUN server which responds with the IP:port pair it has seen in the request message. Then, application A uses the same source port to register the obtained IP:port pair at a well-known rendez-vous point that may be found via DNS. When another application B wants to contact application A, it tells the rendez-vous point to trigger application A so that it sends a UDP packet to application B using a specific destination IP:port pair. This establishes a context for application B in the NAT behind which application A is located. Then, application B can contact application A using the external IP:port combination of A which was previously determined by the STUN server. This is known as UDP hole punching. This method is not applicable for LISP-MN because it would require changes to the specifications of ITRs and the mapping service.

STUN works only with port-restricted and simpler NATs where an outgoing source IP:port combination depends only on the original source IP:port pair. In particular, such a NAT box may reuse the same outgoing source IP:port pair for communication with different destination IP:port combinations. In contrast, symmetric NATs cannot use the same outgoing IP:port combination for different destination IP:port combinations. Therefore, STUN cannot be used for these kind of NATs.

Traversal Using Relays around NAT (TURN) [8] provides relay extensions to STUN. A node behind a NAT registers with a TURN server which effectively sets up a tunnel between them. The TURN server also provides a globally reachable IP:port combination for an application behind a NAT. This address may be announced to other peers that may want to communicate with the node behind the NAT. When communicating with them, traffic to remote peers is tunneled through the NAT box to the TURN server which relays it to the remote peers. In the reverse direction, remote peers send traffic to the TURN server which relays the traffic to the node behind the NAT using address translation. The application of TURN to LISP-

MN would also require major changes to the specification of ITRs and the mapping service.

Interactive Connectivity Establishment (ICE) [12] is another protocol for NAT traversal utilizing STUN and TURN. It discovers and tests the most suitable address pair for two communication peers behind NAT boxes. Hence, ICE has the same disadvantages and is also not applicable as traversal mechanism for LISP-MN.

5.2 Other Mobility Extensions

A different mobility mechanism besides LISP-MN that can be used with Loc/ID split mechanisms is the *Translating Tunnel Router* (TTR) mobility architecture [15]. It has primarily been developed for IVIP [14], but can also be used for LISP and similar architectures. When a mobile node roams into a network, it connects either to a nearby or to a preset TTR and establishes a bidirectional tunnel. The TTR is then registered as ETR for the MN. It receives all encapsulated packets for the MN and sends them through the pre-established tunnel to the mobile node. A MN can establish multiple tunnels to different TTRs and uses the most suitable TTR to send packets to other nodes. During a roaming event for example, the MN retains a tunnel to its previous TTR in addition to the tunnel to its new TTR to keep existing transport connections alive.

Since the tunnel is established by the MN and incoming connection requests do not need to be supported, private addressing and NAT do not cause problems. The major difference of the TTR architecture is that the tunnel to the TTR is always used for outgoing and incoming packets although, for example, both mobile communication partners are in the same domain. In contrast, with LISP-MN, a LISP MN can send outgoing packets to stationary and mobile LISP nodes in other domains and to mobile LISP nodes in the same domain without any path stretch. Extensions to LISP-MN have been proposed to eliminate existing triangle routing for a few other networking scenarios [10].

5.3 Protocols with Similar Techniques

Teredo [5] is a mechanism to provide nodes with only private IPv4 addresses behind a NAT with globally reachable IPv6 addresses. We briefly review it as it has similar features as the NAT traversal mechanisms presented in this paper when looking at IPv6 addresses as EIDs and IPv4 addresses as RLOCs. A host H with a private IPv4 address behind a NAT request a globally reachable IPv6 address from a so-called Teredo server. This Teredo server returns an IPv6 address that contains a Teredo-specific prefix and the external IPv4 IP:port combination of host H. In contrast to LISP EIDs, Teredo addresses are not persistent because they change whenever a node reconnects to the Teredo server and requests a new Teredo address. The request for a Teredo address has established a context in the NAT between the host H and the Teredo server so that the Teredo server can contact host H when needed. This is similar to the context between a MN and an NTR in the proposed NAT traversal method. Packets that are sent to the IPv6 address of host H are routed towards special Teredo relays (comparable to regular ITRs), according to the Teredo prefix. The relay is both connected to the IPv6 and the IPv4 Internet. It sends a trigger-packet over the Teredo server to host H, which then opens a new IPv6-over-IPv4-tunnel to the Teredo relay. From then on, the relay node is able to relay IPv6 packets through the NAT box to host H. Thus, initial packets of a communication experience a significant delay until a context in the NAT box has been established for the relay node.

6. CONCLUSION

In this paper, we presented a NAT traversal mechanism for LISP mobile nodes. When LISP mobile nodes roam into a network be-

hind a NAT, they obtain just private and only locally routable care-of-address. Our proposed mechanism makes the LISP mobile node globally reachable and allows transparent communication with other nodes in the Internet in spite of NAT. While the primary motivation for the presented NAT traversal was to make mobile nodes behind NATs reachable in the Internet, it can also be used to make several LISP domains reachable behind a NAT. Recently, we implemented the NAT traversal mechanism for LISP mobile nodes in the Omnet simulation framework to demonstrate its viability in multiple communication scenarios [6].

7. ACKNOWLEDGEMENTS

The authors would like to thank Dino Farinacci, Gaetan Feige, Joel Halpern, Michael Höfling, Luigi Iannone, David Meyer, Jesper Skriver, and Robin Whittle for insightful comments and fruitful discussions and Prof. Tran-Gia for the stimulating environment and the support through G-Lab.

8. REFERENCES

- [1] D. Farinacci, V. Fuller, D. Lewis, and D. Meyer. LISP Mobile Node. draft-meyer-lisp-mn-01, Feb. 2010.
- [2] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis. Locator/ID Separation Protocol (LISP). draft-ietf-lisp-06, Jan. 2010.
- [3] V. Fuller and D. Farinacci. LISP Map Server. draft-ietf-lisp-ms-04, Oct. 2009.
- [4] V. Fuller, D. Farinacci, D. Meyer, and D. Lewis. LISP Alternative Topology (LISP+ALT). draft-ietf-lisp-alt-03, Mar. 2010.
- [5] C. Huitema. RFC4380: Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs), Feb. 2006.
- [6] D. Klein, M. Hartmann, M. Höfling, and M. Menth. Demo: Improvements to LISP Mobile Node Including NAT Traversal. *EuroView*, Aug. 2010.
- [7] D. Lewis, D. Meyer, D. Farinacci, and V. Fuller. Interworking LISP with IPv4 and IPv6. draft-ietf-lisp-interworking-00, May 2009.
- [8] R. Mahy, P. Matthews, and J. Rosenberg. RFC5766: Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN), Mar. 2010.
- [9] M. Menth, M. Hartmann, D. Klein, and P. Tran-Gia. Future Internet Routing: Motivation and Design Issues. *it - Information Technology*, 5(6), Dec. 2008.
- [10] M. Menth, D. Klein, and M. Hartman. Improvements to LISP Mobile Node. In *Proceedings of the 22nd International Teletraffic Conference (ITC 22)*, Amsterdam, The Netherlands, Sept. 2010.
- [11] D. Meyer, L. Zhang, and K. Fall. RFC4984: Report from the IAB Workshop on Routing and Addressing, Sept. 2007.
- [12] J. Rosenberg. RFC5245: Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols, Feb. 2010.
- [13] J. Rosenberg, R. Mahy, P. Matthews, and D. Wing. RFC5389: Session Traversal Utilities for NAT (STUN), Oct. 2008.
- [14] R. Whittle. Ivip (Internet Vastly Improved Plumbing) Architecture. draft-whittle-ivip-arch-04, Mar. 2010.
- [15] R. Whittle and S. Russert. TTR Mobility Extensions for Core-Edge Separation Solutions to the Internet's Routing Scale Problem. Technical report, Rosanna, Vic, Australia, Aug. 2008.