Network Working Group
Internet-Draft
Intended status: Experimental
Expires: November 8, 2021

V. Ermagan
Google
D. Farinacci
lispers.net
D. Lewis
F. Maino
M. Portoles
Cisco Systems, Inc.
J. Skriver
Arista
C. White
Logicalelegance, Inc.
A. Lopez
A. Cabellos
UPC/BarcelonaTech

May 7, 2021

NAT traversal for LISP
draft-ermagan-lisp-nat-traversal-19

Abstract

This document describes a mechanism for IPv4 NAT traversal for LISP tunnel routers (xTR) and LISP Mobile Nodes (LISP-MN) behind a NAT device. A LISP device both detects the NAT and initializes its state. Forwarding to the LISP device through a NAT is enabled by the LISP Re-encapsulating Tunnel Router (RTR) network element, which acts as an anchor point in the data plane, forwarding traffic from unmodified LISP devices through the NAT.

Table of Contents

1. Introduction 3
2. Definition of Terms 3
3. Basic Overview 5
3.1. LISP NAT Traversal Overview 6
4. LISP RTR Message Details 7
4.1. Info-Request Message 7
4.2. LISP Info-Reply 9
4.3. LISP Map-Register Message 11
4.4. LISP Map-Notify 12
4.5. LISP Data-Map-Notify Message 13
5. Protocol Operations 15
5.1. xTR Processing 15
5.1.1. ETR Registration 16
5.1.2. Map-Request and Map-Reply Handling 17
5.1.3. xTR Sending and Receiving Data 18
5.2. Map-Server Processing 18
5.3. RTR Processing 19
5.3.1. RTR Data Forwarding 21
5.4. Multi-homed xTRs 22
5.5. Updating contents of EID-to-RLOC Mappings 23
5.6. Example 24
6. Security Considerations 27
6.1. Acknowledgments 27
7. IANA Considerations 27
8. Normative References 27
Authors' Addresses 28

1. Introduction

The Locator/ID Separation Protocol [I-D.ietf-lisp-rfc6830bis] [I-D.ietf-lisp-rfc6833bis]defines a set of functions for encapsulating routers to exchange information used to map from Endpoint Identifiers (EIDs) to routable Routing Locators (RLOCs). The assumption that the LISP Tunnel Routers are reachable at their RLOC breaks when a LISP device is behind a NAT. LISP relies on the xTR being able to receive traffic at its RLOC on destination port 4341. However nodes behind a NAT are only reachable through the NAT's public address and in most cases only after the appropriate mapping state is set up in the NAT. Depending on the type of the NAT device, this mapping state may be address and port dependent. In other words, the mapping state in the NAT device may be associated with the 5 tuple that forms a specific flow, preventing incoming traffic from any LISP router other than the one associated with the 5 tuple. A NAT traversal mechanism is needed to make the LISP device behind a NAT reachable.

This document briefly discusses available NAT traversal options, and then it introduces in detail a NAT traversal mechanism for LISP. Two new LISP control messages - LISP Info-Request and LISP Info-Reply - are introduced in order to detect whether a LISP device is behind a NAT, and discover the global IP address and global ephemeral port used by the NAT to forward LISP packets sent by the LISP device. A new LISP component, the LISP Re-encapsulating Tunnel Router (RTR), acts as a re-encapsulating LISP tunnel router [I-D.ietf-lisp-rfc6830bis] to pass traffic through the NAT, to and from the LISP device. A modification to how the LISP Map-Register messages are sent allows LISP device to initialize NAT state to use the RTR services. This mechanism addresses the scenario where the LISP device is behind the NAT, but the associated Map-Server [I-D.ietf-lisp-rfc6833bis] is on the public side of the NAT.

2. Definition of Terms

LISP Info-Request: A LISP control message sent by a LISP device to its Map-Server.

LISP Info-Reply: A LISP control message sent by a Map Server to a LISP device in response to an Info-Request control message.

LISP Re-encapsulating Tunnel Router (RTR): An RTR is a re-encapsulating LISP Router (see [I-D.ietf-lisp-rfc6830bis]). One function that an RTR provides is enabling a LISP device to traverse NATs.

LISP Data-Map-Notify: A LISP Map-Notify message encapsulated in a LISP data header.

LISP xTR-ID A 128-bit field that, together with a site-ID, can be appended at the end of a Map-Register or Map-Notify message. An xTR-ID is used as a unique identifier of the xTR that is sending the Map-Register and is especially useful for identifying multiple xTRs serving the same site/EID-prefix. A value of all zeros indicate the xTR-ID is unspecified.

LISP site-ID A 64-bit field that, together with a xTR-ID, can be appended at the end of a Map-Register or Map-Notify message. A site-ID is used as a unique identifier of a group of xTRs belonging to the same site. A value of 0 indicate the site-ID is unspecified.

NAT: "Network Address Translation is a method by which IP addresses are mapped from one address realm to another, providing transparent routing to end hosts". "Traditional NAT would allow hosts within a private network to transparently access hosts in the external network, in most cases. In a traditional NAT, sessions are uni-directional, outbound from the private network." --RFC 2663 [NAT]. Basic NAT and NAPT are two varieties of traditional NAT.

Basic NAT: "With Basic NAT, a block of external addresses are set aside for translating addresses of hosts in a private domain as they originate sessions to the external domain. For packets outbound from the private network, the source IP address and related fields such as IP, TCP, UDP and ICMP header checksums are translated. For inbound packets, the destination IP address and the checksums as listed above are translated." --RFC 2663[NAT].

NAPT: "NAPT extends the notion of translation one step further by also translating transport identifier (e.g., TCP and UDP port numbers, ICMP query identifiers). This allows the transport identifiers of a number of private hosts to be multiplexed into the transport identifiers of a single external address. NAPT allows a set of hosts to share a single external address. Note that NAPT can be combined with Basic NAT so that a pool of external addresses are used in conjunction with port translation." --RFC 2663[NAT]. Transport identifiers of the destination hosts are not modified by the NAPT.

In this document the general term NAT is used to refer to both Basic NAT and NAPT.

While this document specifies LISP NAT Traversal for LISP tunnel routers, a LISP-MN can also use the same procedure for NAT traversal. The modifications attributed to a LISP-Device, xTR, ETR, and ITR must be supported by a LISP-MN where applicable, in order to achieve NAT traversal for such a LISP node. A NAT traversal mechanism for LISP-MN is also proposed in [NAT-MN].

For definitions of other terms, notably Map-Request, Map-Reply, Ingress Tunnel Router (ITR), and Egress Tunnel Router (ETR), please consult the LISP specification [I-D.ietf-lisp-rfc6833bis].

3. Basic Overview

There are a variety of NAT devices and a variety of network topologies utilizing NAT devices in deployments. Most NAT devices deployed today are designed primarily around the client/server paradigm, where client machines inside a private network initiate connections to public servers with public IP addresses. As such, any protocol requiring a device or host in a private network behind a NAT to receive packets or accept sessions from destinations without first initiating a session or sending packets towards those destinations, will be challenged by deployed NAT devices.

NAT devices are loosely classified based on how restrictive they are. These classifications are essentially identifying the type of mapping state that the NAT device is requiring to allow incoming traffic. For instance, the mapping state may be end-point independent: once device A inside the private network sends traffic to a destination outside, a mapping state in the NAT is created that only includes information about device A, namely its IP address and perhaps its port number. Once this mapping is established in the NAT device, any external device with any IP address could send packets to device A. More restrictive NAT devices could include the 5 tuple information of the flow as part of the mapping state, in other words, the mapping state in the NAT is dependent upon Source IP and Port, as well as destination IP and port (symmetric NAT or Endpoint-dependent NAT). Such a NAT only allows traffic from the specified destination IP and port to reach the specified source device on the specified source port. Traffic with a different 5 tuple signature will not be allowed to pass. In general, in the case of less restrictive NATs it may be possible to eventually establish direct peer-to-peer connections, by means of various hole punching techniques and initial rendezvous servers. However, in the case of symmetric NATs or NATs with endpoint-address-and-port-dependent mappings, direct connection may prove impossible. In such cases a relay device is required that is in the public Network and can relay packets between the two endpoints.

Various methods have been designed to address NAT traversal challenges, mostly in the context of peer-to-peer applications and protocols. Among these, the Interactive Connectivity Establishment (ICE) [ICE] seems the most comprehensive, which defines a protocol that leverages other protocols such as Session Traversal Utilities for NAT(STUN) [STUN] and Traversal Using Relays around NAT (TURN) [TURN], as well as a rendezvous server to identify and exchange a list of potential transport (IP and Port) addresses between the two endpoints. All possible pairs of transport addresses are exhaustively tested to find the best possible option for communication, preferring direct connection to connections using a relay. In the case of most restrictive NATs, ICE leads to use of TURN servers as relay for the traffic. TURN requires a list of allowed peer IP addresses defined as permissions, before allowing a peer to use the relay server to reach a TURN client.

Common NAT traversal techniques such as ICE generally assume bi-directional traffic with the same 5 tuple. LISP, however, requires traffic to use destination UDP port 4341, without specifying the source port. As a result, LISP traffic is generally uni-directional. This means that, in the case of symmetric or endpoint-address-and-port-dependent mapping NATs, even when an outgoing mapping is established, still incoming traffic may not match the established mapping and will not be allowed to pass. As a result, while ICE may be used to traverse less restrictive NATs, use of standard TURN servers as relays to traverse symmetric NATs for LISP protocol is not possible. The rest of this document specifies a NAT traversal technique for the LISP protocol that enables LISP protocol to traverse multiple types of NATs including symmetric NATs.

3.1. LISP NAT Traversal Overview

There are two attributes of a LISP device behind a typical NAT that requires special consideration in LISP protocol behavior in order to make the device reachable. First, the RLOC assigned to the device is typically not globally unique nor globally routable. The NAT likely has a restrictive translation table and forwarding policy, requiring outbound packets to create state before the NAT accepts inbound packets. Second, LISP protocol requires an xTR to receive traffic on a specific UDP port 4341, so the random UDP port allocated by the NAT on its public side to associate with a xTR behind the NAT can not be used by other xTRs to send LISP traffic to. This section provides an overview of the LISP NAT traversal mechanism which deals with these conditions. The following sections specify the mechanism in more detail.

When a LISP device receives a new RLOC and wants to register it with the mapping system, it needs to first discover whether it is behind a

NAT. To do this, an ETR queries its Map-Server to discover the ETR's translated global RLOC and port via the two new LISP messages: Info-Request and Info-Reply. Once an ETR detects that it is behind a NAT, it uses a LISP Re-encapsulating Tunnel Router (RTR) entity as an anchor point for sending and receiving data plane traffic through the NAT device. The ETR registers the RTR RLOC(s) to its Map-Server using the RTR as a proxy for the Map-Register message. The ETR encapsulates the Map-Register message in a LISP ECM header destined to the RTR's RLOC. The RTR strips the LISP ECM header and sends it to the Map-Server. This initializes state in the NAT device so the ETR can receive traffic on port 4341 from the RTR. The ETR also registers the RTR RLOC as the RLOC where the ETR EID prefix is reachable. As a result, all packets destined to the ETR's EID will go to its RTR. The RTR will then re-encapsulate and forward the ETR's traffic via the existing NAT state to the ETR.

Outbound LISP data traffic from the xTR should also be encapsulated to the RTR, where the RTR de-capsulates the LISP packets, and then re-encapsulates them or forwards them natively depending on their destination.

In the next sections these procedures are discussed in more detail.

This document does not support different xTR-ID registering the same EID prefix and using the same set of RTRs. Future versions of this spec will explore this use-case.

4. LISP RTR Message Details

The main modifications in the LISP protocol to enable LISP NAT traversal via an RTR include: (1) two new messages used for NAT discovery (Info-Request and Info-Reply), and (2) encapsulation of two LISP control messages (Map-Register and Map-Notify) between the xTR and the RTR. Map-Register is encapsulated in an ECM header while Map-Notify is encapsulated in a LISP data header (Data-Map-Notify). This section describes the message formats and details of the Info-Request, Info-Reply, and Data-Map-Notify messages, as well as encapsulation details and minor changes to Map-Register and Map-Notify messages.

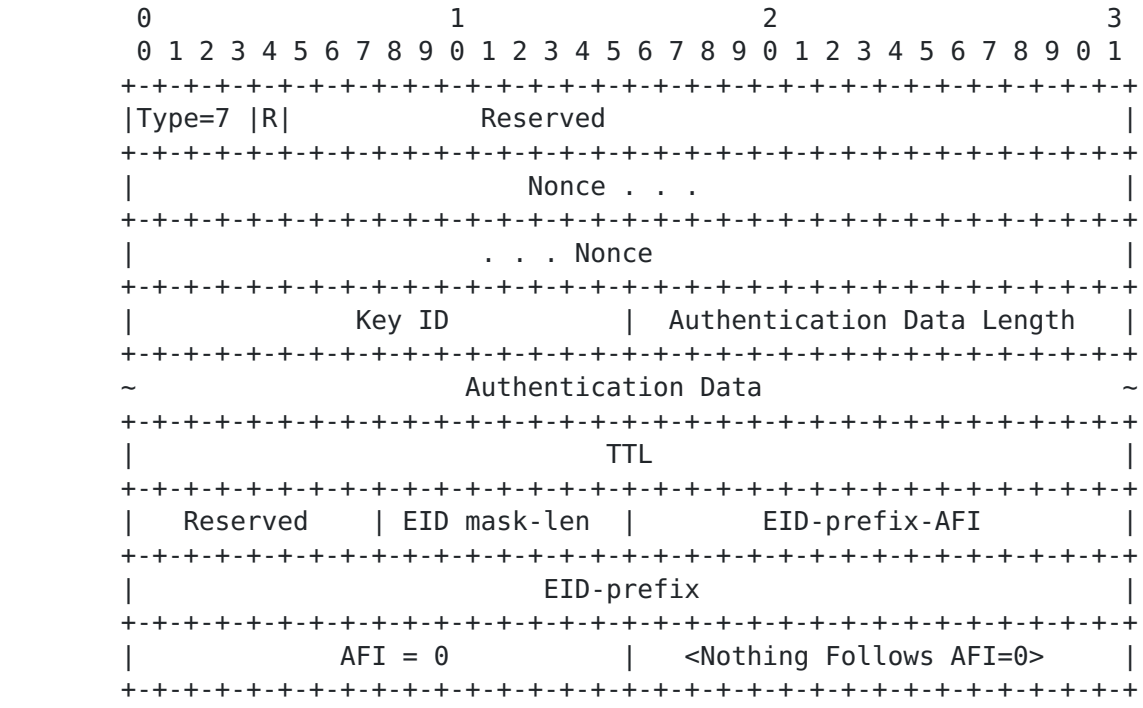
4.1. Info-Request Message

An ETR sends an Info-Request message to its Map-Server in order to

1. detect whether there is a NAT on the path to its Map-Server

2. obtain a list of RTR RLOCs that can be used for LISP data plane NAT traversal.

An Info-Request message is a LISP control message, its source port is chosen by the xTR and its destination port is set to 4342.



LISP Info-Request Message Format

Type: 7 (Info-Request)

R: R bit indicates this is a reply to an Info-Request (Info-Reply). R bit is set to 0 in an Info-Request. When R bit is set to 0, the AFI field (following the EID-prefix field) must be set to 0. When R bit is set to 1, the packet contents follow the format for an Info-Reply, as described below.

Reserved: Must be set to 0 on transmit and must be ignored on receipt.

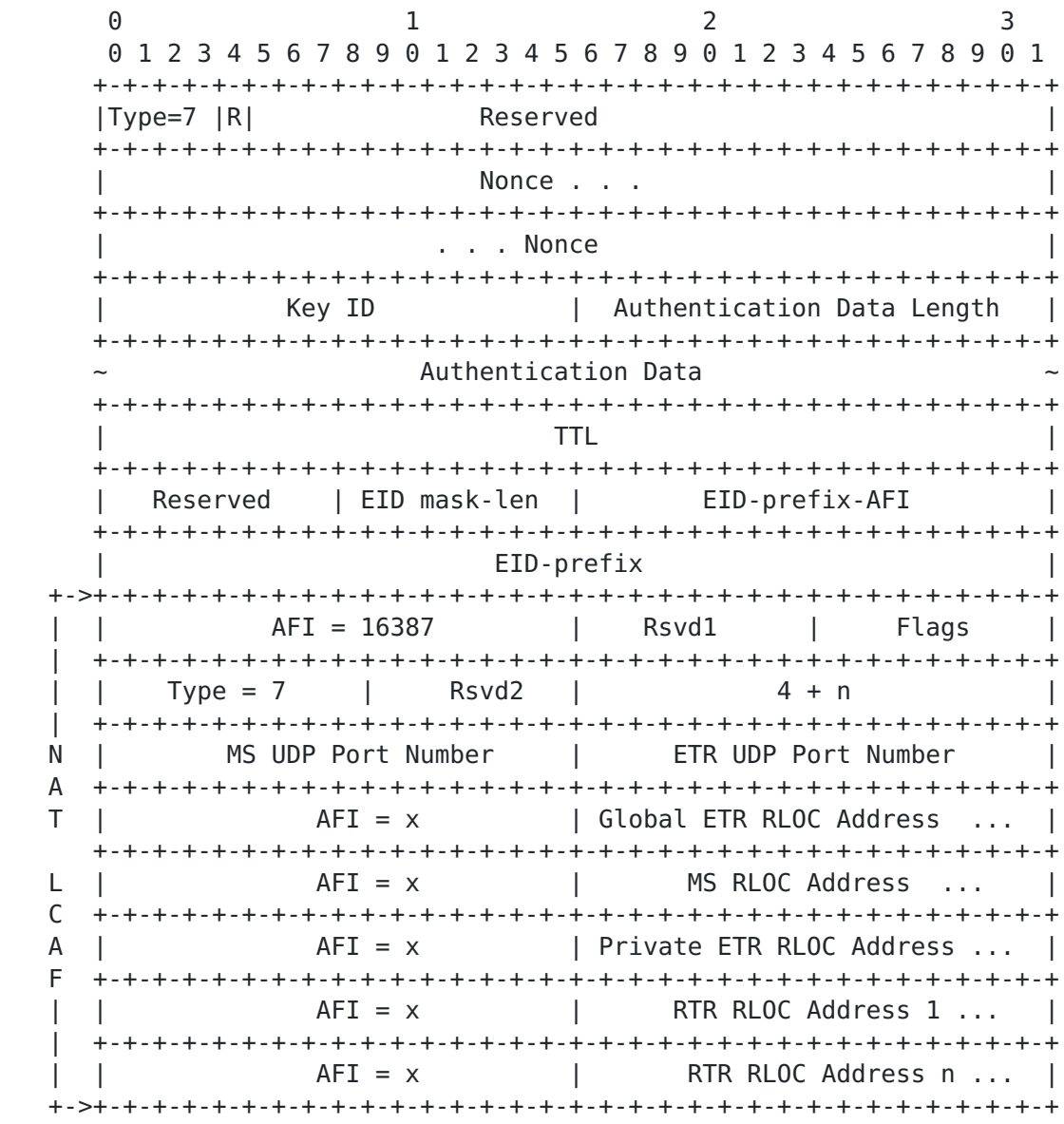
TTL: The time in minutes the recipient of the Info-Reply will store the RTR Information.

Nonce: An 8-byte random value created by the sender of the Info-Request. This nonce will be returned in the Info-Reply. The nonce SHOULD be generated by a properly seeded pseudo-random (or strong random) source. Future version of this document will discuss anti-replay mitigation mechanisms

Descriptions for other fields can be found in the Map-Register section of [I-D.ietf-lisp-rfc6833bis]. Field descriptions for the LCAF AFI = 0 can be found in the LISP LCAF RFC [LCAF] .

4.2. LISP Info-Reply

When a Map-Server receives an Info-Request message, it responds with an Info-Reply message. The Info-Reply message source port is 4342, and destination port is taken from the source port of the triggering Info-Request. Map-Server fills the NAT LCAF (LCAF Type = 7) fields according to their description. The Map-Server uses AFI=0 for the Private ETR RLOC Address field in the NAT LCAF.



LISP Info-Reply Message Format

Type: 7 , R = 1, (Info-Reply)

The format is similar to the Info-Request message. See Info-Request section for field descriptions. Field descriptions for the NAT LCAF section can be found in the LISP LCAF RFC [LCAF] .

4.3. LISP Map-Register Message

The third bit after the Type field in the Map-Register message is allocated as "I" bit. I bit indicates that a 128 bit xTR-ID and a 64 bit site-ID field are present at the end of the Map-Register message. If an xTR is configured with an xTR-ID or site-ID, it MUST set the I bit to 1 and include its xTR-ID and site-ID in the Map-Register messages it generates. If either the xTR-ID or site-ID is not configured an all zeros value is encoded for whichever ID that is not configured.

xTR-ID is a 128 bit field at the end of the Map-Register message, starting after the final Record in the message. The xTR-ID is used to identify the intended recipient xTR for a Map-Notify message, especially in the case where a site has more than one xTR. A value of all zeros indicate that an xTR-ID is not specified, though encoded in the message. This is useful in the case where a site-ID is specified, but no xTR-ID is configured. When a Map-Server receives a Map-Register with an xTR-ID specified (I bit set and xTR-ID has a non-zero value), it MUST copy the XTR-ID from the Map-Register to the associated Map-Notify message. When a Map-Server is sending an unsolicited Map-Notify to an xTR to notify the xTR of a change in locators, the Map-Server must include the xTR-ID for the intended recipient xTR, if it has one stored locally.

site-ID is a 64 bit field at the end of the Map-Register message, following the xTR-ID. site-ID is used by the Map-Server receiving the Map-Register message to identify which xTRs belong to the same site. A value of 0 indicate that a site-ID is not specified, though encoded in the message. When a Map-Server receives a Map-Register with a site-ID specified (I bit set and site-ID has non-zero value), it must copy the site-ID from the Map-Register to the associated Map-Notify message. When a Map-Server is sending an unsolicited Map-Notify to an xTR to notify the xTR of a change in locators, the Map-Server must include the site-ID for the intended recipient xTR, if it has one stored locally.

A LISP device that sends a Map-Register to an RTR must encapsulate the Map-Register message using an Encapsulated Control Message (ECM) [I-D.ietf-lisp-rfc6833bis]. The 6th bit in the ECM LISP header is allocated as the "R" bit. The R bit indicates that the encapsulated Map-Register is to be processed by an RTR. The 7th bit in the ECM header is allocated as the "N" bit. The N bit indicates that this Map-Register is being relayed by an RTR. When an RTR relays the ECM-ed Map-Register to a Map-Server, the N bit must be set to 1.

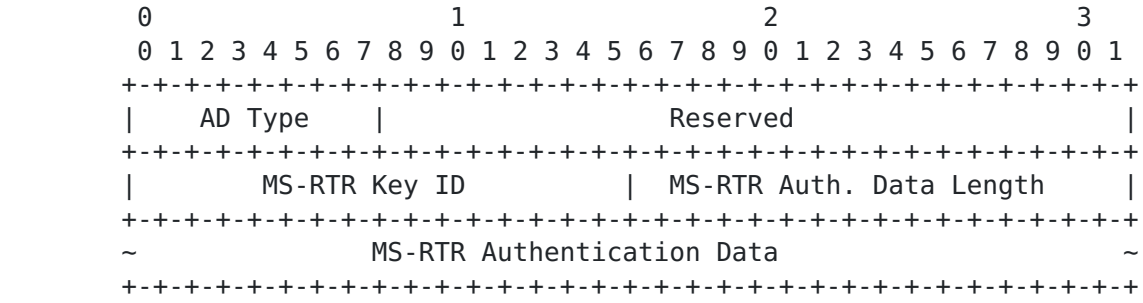
The outer header source RLOC of the ECM is set to the LISP device's local RLOC, and the outer header source port is set to 4341. The

outer header destination RLOC and port are set to RTR RLOC and 4342 respectively. The inner header source RLOC is set to LISP device's local RLOC, and the inner source port is picked at random. The inner header destination RLOC is set to the xTR's Map-Server RLOC, and inner header destination port is set to 4342.

4.4. LISP Map-Notify

The first bit after the Type field in a Map-Notify message is allocated as the "I" bit. I bit indicates that a 128 bit xTR-ID and 64 bit site-ID field is present at the end of the Map-Notify message, following the final Record in the Map-Notify (See Section 4.3 for details on xTR-ID and site-ID). A Map-Server MUST set the I bit in a Map-Notify and include the xTR-ID and/or site-ID of the intended recipient xTR if the associated Map-Register has an xTR-ID and/or site-ID specified, or when the Map-Server has previously cached an xTR-ID and/or site-ID for the destination xTR.

A LISP device that sends a Map-Notify to an RTR must encapsulate the Map-Notify message using an ECM. The 6th bit in the ECM LISP header, allocated as the "R" bit, must be set when the encapsulated Map-Notify is to be processed by an RTR. If the S bit is also set in the Map-Notify ECM header, it indicates that additional MS-RTR authentication data is included after the LISP header in the ECM. If the I bit is also set in the Map-Notify, the xTR-ID and site-ID fields are included in the Map-Notify. If a Map-Server receiving an ECM-ed Map-Register has a shared key associated with the sending RTR, it must generate a Map-Notify message with the S bit in the ECM header set to 1, and with the additional MS-RTR authentication related fields described below.



Changes to LISP Map-Notify Message

AD Type: 2 (RTR Authentication Data)

MS-RTR Key ID: A configured ID to find the configured Message Authentication Code (MAC) algorithm and key value used for the

authentication function. See [I-D.ietf-lisp-rfc6833bis] section 12.5 for code point assignments.

MS-RTR Authentication Data Length: The length in bytes of the MS-RTR Authentication Data field that follows this field. The length of the Authentication Data field is dependent on the Message Authentication Code (MAC) algorithm used. The length field allows a device that doesn't know the MAC algorithm to correctly parse the packet.

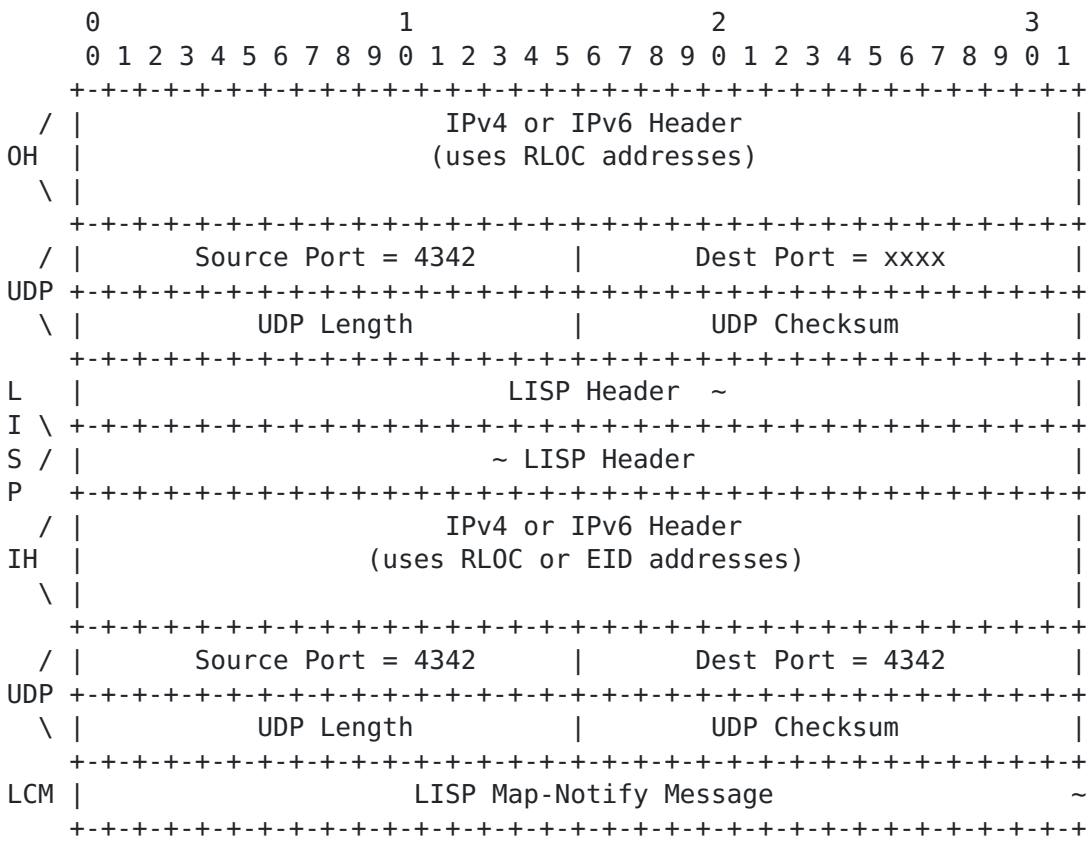
MS-RTR Authentication Data: The message digest used from the output of the Message Authentication Code (MAC) algorithm. The entire Map-Notify payload is authenticated. After the MAC is computed, it is placed in this field. Implementations of this specification MUST support HMAC-SHA-1-96 [RFC2404] and SHOULD support HMAC-SHA-256-128 [RFC6234].

For a full description of all fields in the Map-Notify message refer to Map-Notify section in [I-D.ietf-lisp-rfc6833bis].

The outer header source RLOC of the ECM is set to the xTR's Map-Server RLOC, and the outer header source port is set to 4342. The outer header destination RLOC and port are set to RTR's RLOC and 4342 respectively. The inner header source RLOC is set to the xTR's Map-Server RLOC, and the inner source port is set to 4342. The inner header destination RLOC is set to the LISP device's local RLOC copied from the Map-Register, and inner header destination port is set to 4342.

4.5. LISP Data-Map-Notify Message

When an RTR receives an ECM-ed Map-Notify message with R bit in the ECM header set to 1, it has to relay the Map-Notify payload to the registering LISP device. After removing the ECM header and processing the Map-Notify message as described in Section 5.3, the RTR encapsulates the Map-Notify in a LISP data header and sends it to the associated LISP device. This Map-Notify inside a LISP data header is referred to as a Data-Map-Notify message.



LISP Data-Map-Notify Message

In a Data-Map-Notify, the outer header source RLOC is set to the RTR's RLOC that was used in the associated Map-Register. This is previously cached by the RTR. The outer header source port is set to 4342. The outer header destination RLOC and port are filled based on the translated global RLOC and port of the registering LISP device previously stored locally at the RTR. The inner header source address is Map-Server's RLOC, and inner header source port is 4342. The inner header destination address is the LISP device's local RLOC. The inner header destination port is 4342.

Since a Data-Map-Notify is a control message encapsulated in a LISP data header, a special Instance ID is used as a signal for the xTR to trigger processing of the control packet inside the data header. The Instance ID value 0xFFFFFF is reserved for this purpose. The Instance ID field in a Data-Map-Notify must be set to 0xFFFFFF.

5. Protocol Operations

There are two main steps in the NAT traversal procedure. First, the ETR's translated global RLOC must be discovered. Second, the NAT translation table must be primed to accept incoming connections. At the same time, the RTR must be informed of the ETR's translated global RLOC including the translated ephemeral port number(s) at which the RTR can reach the LISP device.

5.1. xTR Processing

Upon receiving a new local RLOC, an ETR first has to detect whether the new RLOC is behind a NAT device. For this purpose the ETR sends an Info-Request message to its Map-Server in order to discover the ETR's translated global RLOC as it is visible to the Map-Server. The ETR uses its new local RLOC as the source RLOC of the message. The Map-Server, after authenticating the message, responds with an Info-Reply message. The Map-Server includes the source RLOC and port from the Info-Request message in the Global ETR RLOC Address and ETR UDP Port Number fields of the Info-Reply. The Map Server also includes the destination RLOC and port number of the Info-Request message in the MS RLOC Address and MS UDP Port Number fields of the Info-Reply. In addition, the Map-Server provides a list of RTR RLOCs that the ETR may use in case it needs NAT traversal services. The source port of the Info-Reply is set to 4342 and the destination port is copied from the source port of the triggering Info-Request message.

Upon receiving the Info-Reply message, the ETR compares the source RLOC and source port used for the Info-Request message with the Global ETR RLOC Address and ETR UDP Port Number fields of the Info-Reply message. If the two are not identical, the ETR concludes that its new local RLOC is behind a NAT and that it requires an RTR for NAT traversal services in order to be reachable at that RLOC. An ETR behind other statefull devices (e.g. statefull firewalls) may also use an RTR and the procedure specified here for traversing the statefull device. Detecting existence of such devices are beyond scope of this document.

It is worth noting that a STUN server can also be used to do NAT detection and to discover the NAT-translated public IP address and port number for the ETR behind NAT. If a STUN server is used, list of RTR devices that can be used by the xTR for NAT traversal must be provisioned to the xTR via other means which are outside the scope of this document.

If there is no NAT on the path identified by an info-Request and an Info-Reply, the ETR registers the associated RLOC with its Map-Server as described in [I-D.ietf-lisp-rfc6833bis].

A device that is not behind a NAT can still choose to use RTR as an anchor point. In some deployments, devices can roam from NATed connections to non-NATed connections. In such scenarios, the deployer may choose to always use an RTR, thus avoiding frequent notifications of RLOC changes. The main disadvantage of this policy is that RTRs may increase path length.

5.1.1. ETR Registration

Once an ETR has detected that it is behind a NAT, based on local policy the ETR selects one (or more) RTR(s) from the RTR RLOCs provided in the Info-Reply and initializes state in the NAT device in order to receive LISP data traffic on UDP port 4341 from the selected RTR. To do so, the ETR sends a Map-Register encapsulated in an ECM header to the selected RTR(s). The Map-Register message is created as specified in [I-D.ietf-lisp-rfc6833bis]. More specifically, the source RLOC of the Map-Register is set to ETR's local RLOC, while the destination RLOC is set to the ETR's Map-Server RLOC, and destination port is set to 4342. The ETR sets the P bit (proxy Map-Reply) and the M bit (want-Map-Notify) in Map-Register to 1, and it includes the selected RTR RLOC(s) as the locators in the Map-Register message. The ETR can also include its local RLOCs as locators in the Map-Register, including weight and priorities, while setting the R bit to 0 for each local RLOC. This can be used by the RTR for load balancing when forwarding data to a multi-homed xTR behind a NAT. The R bit is set to 1 for all RTR locators included in the Map-Register. The ETR must also set the I bit in the Map-Register message to 1 and include its xTR-ID and site-ID in the corresponding field. In the ECM header of this Map-Register the source RLOC is set to ETR's local RLOC and the source port is set to 4341, while the destination RLOC is the RTR's RLOC and the destination port is set to LISP control port 4342. The R bit in the ECM header is also set to 1, to indicate that this EDCM-ed Map-Register is to be processed by an RTR.

This ECM-ed Map-Register is then sent to the RTR. The RTR removes the ECM header, encapsulates the new Map-Register in a new ECM header with R bit set to 0, and sends it to the associated Map-Server. The RTR then encapsulates the corresponding Map-Notify message in a LISP data header (Data-Map-Notify) and sends it back to the xTR.

Upon receiving a Data-Map-Notify from the RTR, the ETR must strip the outer LISP data header, and process the inner Map-Notify message as described in [I-D.ietf-lisp-rfc6833bis]. Since outer header destination port in Data-Map-Notify is set to LISP data port 4341, the Instance ID 0xFFFFFF in the LISP header of the Data-Map-Notify is used by the ETR to detect and process the Data-Map-Notify as a control message encapsulated in a LISP data header. While processing

the Data-Map-Notify, the xTR also stores the RTR RLOC(s) as its data plane proxy for the interface/RLOC behind the NAT.

If the xTR is not multi-homed, or if all its interfaces are behind the NAT and will use the same RTR, then the xTR should map the EID prefix 0/0 to this RTR RLOC(s) in its map-cache. This results in the xTR encapsulating all LISP data plane traffic to this RTR, reducing the state created in the NAT. Note that not installing the default map-cache entry will lead to normal Map-Request and Map-Reply messages for EID mapping lookups which is only supported if the xTR has interfaces not behind NAT. If outgoing traffic is sent directly to destinations without passing through the RTR, this will result in additional state to be created in the NAT device.

At this point the registration and state initialization is complete and the xTR can use the RTR services. The state created in the NAT device based on the ECM-ed Map-Register and corresponding Data-Map-Notify is used by the xTR behind the NAT to send and receive LISP control packets to/from the RTR, as well as for receiving LISP data packets form the RTR.

If ETR receives a Data-Map-Notify with a xTR-ID specified, but the xTR-ID is not equal to its local xTR-ID, it must log this as an error. The ETR should discard such Data-Map-Notify message.

The ETR must periodically send ECM-ed Map-Register messages to its RTR in order to both refresh its registration to the RTR and the Map-Server, and as a keep alive in order to preserve the state in the NAT device. RFC 2663 [NAT] points out that the period for sending the keep alives can be set to default value of two minutes, however since shorter timeouts may exist in some NAT deployments, the interval for sending periodic ECM-ed Map-Registers must be configurable.

5.1.2. Map-Request and Map-Reply Handling

To avoid the creation of extra status in the NAT devices due to the Map-Replies send to requesting ITRs, the ETR set the proxy-bit of the Map-Register to one in order the Map-Server proxy-reply the ETR as described in [I-D.ietf-lisp-rfc6833bis]

When an ITR behind a NAT is encapsulating outbound LISP traffic, it can use its RTR RLOC as the locator for all destination EIDs that it wishes to send data to. As such, the ITR does not need to send Map-Requests for finding EID-to-RLOC mappings. However, if the ITR is multihomed and has at least one interface not behind NAT, it can choose to send Map-Requests. For this, the ITR specifies in the ITR-RLOC field of the Map-Request the list of RLOCs that are not behind NAT to receive the Map-Reply messages. It should be noted that

sending packets directly to destination RLOCs through the interface behind NAT will result in creating additional state in the NAT device. Also, it should be noted that outgoing packets use a direct path while the incoming packets are forwarded through an RTR.

For RLOC-probing, the periodic ECM-ed Map-Register and Data-Map-Notify messages between xTR and RTR can also serve the purpose of RLOC probes. However, if RLOC-probing is used, no changes are required to the RLOC-probing specification in [I-D.ietf-lisp-rfc6833bis], except that the LISP device behind a NAT only needs to probe the RTR's RLOC.

5.1.3. xTR Sending and Receiving Data

When a Map-Request for a LISP device behind a NAT is received by its Map-Server, the Map-Server responds with a Map-Reply including RTR's RLOC as the locator for the requested EID. As a result, all LISP data traffic destined for the ETR's EID behind the NAT is encapsulated to its RTR. The RTR re-encapsulates the LISP data packets to the ETR's translated global RLOC and port number so the data can pass through the NAT device and reach the ETR. As a result the ETR receives LISP data traffic with outer header destination port set to 4341 as specified in [I-D.ietf-lisp-rfc6830bis].

For sending outbound LISP data, an ITR behind a NAT SHOULD use the RTR RLOC as the locator for all EIDs that it wishes to send data to via the interface behind the NAT. The ITR then encapsulates the LISP traffic in a LISP data header with outer header destination set to RTR RLOC and outer header destination port set to 4341. This may create a secondary state in the NAT device. ITR SHOULD set the outer header source port in all egress LISP data packets to a random but static port number in order to avoid creating excessive state in the NAT device.

If the ITR and ETR of a site are not collocated, the RTR RLOC must be configured in the ITR via an out-of-band mechanism. Other procedures specified here would still apply.

5.2. Map-Server Processing

Upon receiving an Info-Request message a Map-Server first verifies the authenticity of the message. Next the Map-Server creates an Info-Reply message and copies the source RLOC and port number of the Info-Request message to the Global ETR RLOC Address and ETR UDP Port Number fields of the Info-Reply message. The Map-Server also includes a list of RTR RLOCs that the ETR may use for NAT traversal services. The Map-Server sends the Info-Reply message to the ETR, by setting the destination RLOC and port of the Info-Reply to the source

RLOC and port of the triggering Info-Request. The Map-Server sets the source port of the Info-Reply to 4342.

Upon receiving an ECM-ed Map-Register message with the N bit in the ECM header set to 1, the Map-Server removes the ECM header and if the M bit in the Map-Register is set, the Map-Server processes the Map-Register message and generates the resulting Map-Notify as described in [I-D.ietf-lisp-rfc6833bis]. The Map-Server encapsulates the Map-Notify in an ECM header and sets the R bit in the ECM header to 1. This indicates that the ECM-ed Map-Notify is to be processed by an RTR. If the Map-Server has a shared secret configured with the RTR sending the Map-Register, the Map-Server also sets the S bit in the ECM header of the Map-Notify and includes the MS-RTR authentication data after the ECM LISP header. See Security Considerations Section for more details. If the I bit is set in the Map-Register message, the Map-Server also locally stores the xTR-ID and site-ID from the Map-Register, and sets the I bit in the corresponding Map-Notify message and includes the same xTR-ID and site-ID in the Map-Notify. The ECM-ed Map-Notify is then sent to the RTR sending the corresponding Map-Register.

5.3. RTR Processing

Upon receiving an ECM-encapsulated Map-Register with the R bit set in the ECM header, the RTR creates a map-cache entry for the EID-prefix that was specified in the Map-Register message. The RTR stores the outer header source RLOC and outer header source port, the outer header destination RLOC (RTR's own RLOC), the inner header source RLOC (xTR's local RLOC), the xTR-ID, the weight and priority associated with the xTR's local RLOC that was used to send this Map-Register if present, and the nonce field of the Map-Register in this local map-cache entry. The RTR uses the inner header source address to identify which xTR local RLOC (R bit =0) was used by the xTR to send this Map-Register. The outer header source RLOC and outer header source port is the ETR's translated global RLOC and port number visible to the RTR. Once the registration process is complete, this map-cache entry can be used to send LISP data traffic to the ETR. The outer header destination RLOC is the RTR's RLOC used by the ETR. The RTR can later use these fields as the source RLOC for sending data-encapsulated control messages (Data-Map-Notify) back to the ETR. The nonce field is used for security purposes and is matched with the nonce field in the corresponding Map-Notify message. This map-cache entry is stored as an "unverified" mapping, until the corresponding Map-Notify message is received.

In the cases where the xTR has multiple RLOCs behind the NAT, and requires the RTR to load balance the traffic across those interfaces, the xTR must include the local RLOCs associated with each interface

behind the NAT with the R bit in the locator record set to 0 in the ECM-ed Map-Register sent to the RTR. The RTR uses the weight and priority policies of the RLOCs with R=0 in the Map-Register to load balance the traffic from the RTR to the xTR behind the NAT. The RTR compares the RLOCs with the R bit set to 0 in the Map-Register to the inner header source address of the Map-Register to find the matching RLOC that the xTR used to send the Map-Register from. The RTR associates the weight and priority policies of this local RLOC with the NAT-translated RLOC and xTR-ID for this map-cache entry. For all other local RLOCs included in the Map-Register, that the Map-Register is not originating from, the RTR only updates previously cached weight and priority policies if it already has those local RLOCs previously stored for that EID prefix and xTR-ID. In other words, the RTR only adds new local RLOCs and their weight and priority policies to its cache if the Map-Register is actually originating from that RLOC. The TTL for every map-cache is also only updated when a Map-Register is originating from the same RLOC. However, the weight and priorities of all previously cached local RLOCs will be updated by every Map-Register, whether it is originating from that RLOC or not. The xTR-ID is used to define the Merge domain for these RLOCs. In other words, a Map-Register originating from a unique xTR-ID will always overwrite previously stored policies for that xTR-ID. However it does not modify in any way the policies indicated by any other xTR-ID serving the same EID prefix. As a result, in the case of a renumbering or xTR reboot, the xTR uses its unique xTR-ID to send a new Map-Register, overwriting the previously stored policies for that xTR. Using this method the xTR can immediately remove any RLOCs from the RTR cache that are no longer active. In order to implement this, the RTR must compare the list of local RLOCs in the Map-Register (R=0) with the ones it has previously cached associated with the same xTR-ID. If there is any RLOC previously cached that does not appear in the newly received Map-Register, the RTR must remove that RLOC together with the associated translated RLOC and associated policies, because removal of a local (behind-the-NAT) RLOC also invalidates the NAT-ed address associated with it. .

After filling the local map-cache entry, the RTR strips the outer header and extracts the Map-Register message, encapsulated in a new ECM header with the R bit set to 0, and N bit set to 1, and sends the ECM-ed Map-Register to destination Map-Server.

Map-Server responds with a ECM-ed Map-Notify message to the RTR.

Upon receiving an ECM-ed Map-Notify message with R bit set to 1 in the ECM header, if the S bit in ECM header is set to 1, RTR uses the MS-RTR Key ID to verify the MS-RTR Authentication Data included after the ECM header. If the MS-RTR authentication fails, the RTR must drop the packet. Once the authenticity of the message is verified,

RTR can confirm that the Map-Register message for the ETR with the matching xTR-ID was accepted by the Map-Server. At this point the RTR can change the state of the associated map-cache entry to verified for the duration of the Map-Register TTL.

The RTR then uses the information in the associated map-cache entry to create a Data-Map-Notify message according to the following procedure: The RTR encapsulates the Map-Notify in a LISP data header, where the outer header destination RLOC and port number are set to the ETR's translated global RLOC and port number. If more than one ETR translated RLOC and port exists in the map-cache entry for the same EID prefix specified in the Map-Notify, the RTR can use the xTR-ID from the Map-Notify to identify which ETR is the correct destination for the Data-Map-Notify. The RTR sets the outer header source RLOC to RTR's RLOC from the map-cache entry and the outer header source port is set to 4342. The RTR also sets the Instance ID field in the LISP header of the Data-Map-Notify to 0xFFFFFF. The RTR then sends the Data-Map-Notify to the ETR.

If the S bit is set to 0 in the ECM header of the Map-Notify, and the RTR has a shared key configured locally with the sending Map-Server, the RTR must drop the packet. If the S bit is set to 0, and the RTR does not have a shared key configured with the associated Map-Server, according to local policy, the RTR may drop the packet. If the Map-Notify with S bit set to 0 is processed, the RTR must match the nonce field from this Map-Notify with the nonce stored in the local map-cache entry with the matching xTR-ID. If the nonces do not match, the RTR must drop the packet.

An RTR receiving an unsolicited Map-Notify for a registered EID should check if any of the RTR's RLOCs is present in the received record. If it does, the RTR rewrites the inner header destination RLOC of the Map-Notify message to ETR's local RLOC obtained from the associated map-cache entry of the EID. Then the RTR encapsulates and forwards the Map-Notify in a LISP data packet as explained above. If the record of the received Map-Notify doesn't contain any locator of the RTR it should drop it and request the new mapping to the mapping system. If the map reply record doesn't contain any RTR's locator, the map-cache entry for the EID is replaced otherwise, the Map-Reply is silently dropped.

5.3.1. RTR Data Forwarding

For all LISP data packets encapsulated to RTR's RLOC and outer header destination port 4341, the RTR first verifies whether the source or destination EID is a previously registered EID. If so, the RTR must process the packet according to the following. If the destination or

source EID is not a registered EID, the RTR can drop or process the packets based on local policy.

In the case where the destination EID is a previously registered EID, the RTR must strip the LISP data header and re-encapsulate the packet in a new LISP data header. The outer header RLOCs and UDP ports are then filled based on the matching map-cache entry for the associated destination EID prefix. The RTR uses the RTR RLOC from the map-cache entry as the outer header source RLOC. The outer header source port is set to 4342. The RTR sets the outer header destination RLOC and outer header destination port based on the ETR translated global RLOC and port stored in the map-cache entry. Then the RTR forwards the LISP data packet.

In the case where the source EID is a previously registered EID, the RTR process the packet as if it is a Proxy ETR (PETR). The RTR must strip the LISP data header, and process the packet based on its inner header destination address. The packet may be forwarded natively, it may be LISP encapsulated to the destination ETR, or it may trigger the RTR to send a LISP Map-Request.

5.4. Multi-homed xTRs

In the case where an xTR has multiple interfaces and RLOCs, info-Requests can be sent per each interface and NAT discovery is done per each interface. NAT traversal is accomplished by following state and processes described above per each interface/RLOC. In other words, if multiple interfaces of an xTR are behind a NAT, the ECM-ed Map-Register messages should be sent via each xTR interface behind NAT if the xTR desires to receive traffic via that interface. This is required to establish the state in the NAT device for that interface. The M bit (want Map-Notify) must be set in ECM-ed Map-Register messages sent from at least one of xTR interfaces behind the NAT. If additional interfaces behind the NAT are using the same RTR for NAT traversal, no Map-Notify processing is required for such interfaces and M bit in Map-Register can be set to 0 for these to reduce processing on the RTR and the Map-Server.

The RLOCs included in Map-Register messages when the xTR has multiple interfaces SHOULD be the union of the locators (behind NAT or not) resulting from the process defined above per each RLOC of the xTR, according to the specifics of that interface (whether it is behind the NAT or not).

In cases where some xTR interfaces are behind NAT while others are not, ECM-ed Map-Register messages should be sent via interfaces behind the NAT through the selected RTRs. xTR can receive traffic via both types of interfaces by including the associated RLOCs (as well

as the RTR RLOCs) in its ECM-ed Map-Register messages. The xTR can choose to store RTR RLOCs in a default map-cache entry to forward all the traffic through the RTR. If the xTR decides to populate its map-cache, the xTR may configure the RTR as a proxy of the interface behind NAT instead of sending the traffic directly to avoid generate new state in the NAT device.

5.5. Updating contents of EID-to-RLOC Mappings

When an interface of a LISP device is configured with a new RLOC, it needs to discover whether it is behind NAT by sending an Info-Request message. If the modified interface was previously behind NAT and the new RLOC is also behind NAT using the same RTRs association, the cache of the RTR is updated through the ECM-ed Map-Register send by the ETR. Otherwise, the change of the EID-to-RLOC mapping needs to be notified to remote ITRs/PiTRs.

LISP defines several mechanisms to update mappings in the Data-Plane [I-D.ietf-lisp-rfc6830bis] and in the control plane [I-D.ietf-lisp-rfc6833bis]. Also an additional Control-Plane mechanism based on the Publish/subscribe paradigm is specified in [I-D.ietf-lisp-pubsub]. It is recommended to use [I-D.ietf-lisp-pubsub] to notify mapping changes as it is a faster mechanism and it avoids the temporal increase of the path length associated with an ETR desassociating of an RTR.

In the case of Solicit-Map-Request mechanism, when an RTR receives a Soliciting Map-Request for a cached EID-Prefix, the RTR should request for the new mapping to the mapping system. If the Map-Reply record contains any RTR's locator, the RTR silently discards the Map-Reply; otherwise, the status of the map-cache entry used to punch the NAT of the EID-Prefix would be lost. If the Map-Reply record doesn't contain any RTR's locator, the RTR must update the cache entry.

A Map-Server receiving a new registration for an EID should send unsolicited Map-Notify to the departure ETRs as specified in section 4.2.3 of [I-D.ietf-lisp-eid-mobility]. An RTR receiving an unsolicited Map-Notify for a registered EID should process it as described in Section 5.3.

An RTR that replace a registered cache entry through a Map-Request / Map-Reply must forward traffic to the EID as described in [I-D.ietf-lisp-rfc6833bis] until the map-cache entry expires. Optionally, the RTR can send Data Driven SMRs to ITRs sending traffic to the removed EID through the RTR as described in section 4.2.4 of [I-D.ietf-lisp-eid-mobility].

5.6. Example

What follows is an example of an ETR initiating a registration of a new RLOC to its Map-Server, when there is a NAT device on the path between the ETR and the Map-Server.

In this example, the ETR (site1-ETR) is configured with the local RLOC of 192.168.1.2. The NAT's global (external) addresses are from 2.0.0.1/24 prefix. The Map-Server is at 3.0.0.1. And one potential RTR has an IP address of 1.0.0.1. The site1-ETR has an EID Prefix of 128.1.0.0/16.

An example of the registration process follows:

1. The Site1-ETR receives the private IP address, 192.168.1.2 as its RLOC via DHCP.

2. The Site1-ETR sends an Info-Request message with the destination RLOC of the Map-Server, 3.0.0.1, and source RLOC of 192.168.1.2. This packet has the destination port set to 4342 and the source port is set to (for example) 5001.

3. The NAT device translates the source IP from 192.168.1.2 to 2.0.0.1, and source port to (for example) 20001 global ephemeral source port.

4. The Map-Server receives and responds to this Info-Request with an Info-Reply message. This Info-Reply has the destination address set to ETR's translated address of 2.0.0.1 and the source address is the Map-Server's RLOC, namely 3.0.0.1. The destination port is 20001 and the source port is 4342. Map-Server includes a copy of the source address and port of the Info-Request message (2.0.0.1:20001), and a list of RTR RLOCs including RTR RLOC 1.0.0.1 in the Info-Reply contents.

5. The NAT translates the Info-Reply packet's destination IP from 2.0.0.1 to 192.168.1.2, and translates the destination port from 20001 to 5001, and forwards the Info-Reply to site1-ETR at 192.168.1.2.

6. The Site1-ETR detects that it is behind a NAT by comparing its local RLOC (192.168.1.2) with the Global ETR RLOC Address in the Info-Reply (2.0.0.2) . Then site1-ETR picks the RTR 1.0.0.1 from the list of RTR RLOCs in the Info-Reply. ETR stores the RTR RLOC in a default map-cache entry to periodically send ECM-ed Map-Registers to.

7. The ETR sends an ECM encapsulated Map-Register to RTR at 1.0.0.1. The outer header source RLOC of this Map-Register is set to 192.168.1.2 and the outer header source port is set to 4341. The outer header destination RLOC and port are set to RTR RLOC at 1.0.0.1 and 4342 respectively. The R bit in ECM header is set to 1. The inner header destination RLOC is set to ETR's Map-Server 3.0.0.1, and the inner header destination port is set to 4342. The inner header source RLOC is set to ETR's local RLOC 192.168.1.2 and the source port is set to (for example) 5002. In the Map-Register message the RTR RLOC 1.0.0.1 appears as the locator set for the ETR's EID prefix (128.1.0.0/16). In this example ETR also sets the Proxy bit in the Map-Register to 1, and sets I bit to 1, and includes its xTR-ID and site-ID in the Map-Register.

8. The NAT translates the source RLOC in the ECM header of the Map-Register, by changing it from 192.168.1.2 to 2.0.0.2, and translates the source port in the ECM header from 4341 to (for example) 20002, and forwards the Map-Register to RTR.

9. The RTR receives the Map-Register and creates a map-cache entry with the ETR's xTR-ID, EID prefix, and the source RLOC and port of the ECM header of the Map-Register as the locator (128.1.0.0/16 is mapped to 2.0.0.2:20002). RTR also caches the inner header source RLOC of the Map-Register namely 192.168.1.2, and the outer header destination RLOC of the ECM header in the Map-Register (this would be RTR's RLOC 1.0.0.1) to use for sending back a Data-Map-Notify. RTR then removes the outer header, adds a new ECM header with R=0, and N=1, and forwards the Map-Register to the destination Map-Server.

10. The Map-Server receives the ECM-ed Map-Register with N bit set to 1, removes the ECM header, and processes it according to [I-D.ietf-lisp-rfc6833bis]. Since Map-Server has a shared secret with the sending RTR, after registering the ETR, Map-Server responds with a ECM-ed Map-Notify with the R bit and S bit both set to 1 in the ECM header and including the MS-RTR authentication data. Since the I bit is set in the Map-Register, the Map-Server also sets the I bit in the Map-Notify and copies the xTR-ID and site-ID from the Map-Register to the Map-Notify. The source address of this Map-Notify is set to 3.0.0.1. The destination is copied from the local source address of the Map Register (192.168.1.2), and both source and destination ports are set to 4342.

11. The RTR receives the ECM-ed Map-Notify and verifies the MS-RTR authentication data. The RTR data-encapsulates the Map-Notify and sends the resulting Data-Map-Notify to site1-ETR with a

matching xTR-ID. The outer header source RLOC and port of the Data-Map-Notify are set to 1.0.0.1:4342. The outer header destination RLOC and port are retrieved from previously cached map-cache entry in step 9 namely 2.0.0.2:20002. RTR sets the Instance ID in the LISP header to 0xFFFFFF. At this point RTR marks ETR's EID prefix as "Registered" status and forwards the Data-Map-Notify to ETR.

12. The NAT device translates the destination RLOC and port of the Data-Map-Notify to 192.168.1.2:4341 and forwards the packet to ETR.

13. The Site1-ETR receives the packet with a destination port 4341, and processes the packet as a control packet after observing the Instance ID value 0xFFFFFF in the LISP header. At this point ETR's registration to the RTR is complete.

Assume a requesting ITR in a second LISP (site2-ITR) site has an RLOC of 74.0.0.1. The following is an example process of an EID behind site2-ITR sending a data packet to an EID behind the site1-ETR:

1. The ITR sends a Map-Request which arrives via the LISP mapping system to the ETR's Map Server.

2. The Map-Server sends a Map-Reply on behalf of the ETR, using the RTR's RLOC (1.0.0.1) in the Map-Reply's Locator Set.

3. The ITR encapsulates a LISP data packet with ITR's local RLOC (74.0.0.1) as the source RLOC and the RTR as the destination RLOC (1.0.0.1) in the outer header.

4. The RTR decapsulates the packet, evaluates the inner header against its map-cache and then re-encapsulates the packet. The new outer header's source RLOC is the RTR's RLOC 1.0.0.1 and the new outer header's destination RLOC is the Global NAT address 2.0.0.2. The destination port of the packet is set to 20002 (discovered above during the registration phase) and the source port is 4342.

5. The NAT translates the LISP data packet's destination IP from to 2.0.0.2 to 192.168.1.2, and translates the destination port from 20002 to 4341, and forwards the LISP data packet to the ETR at 192.168.1.2.

6. For the reverse path the ITR uses its local map-cache entry with the RTR RLOC as the default locator and encapsulates the LISP data packets using RTR RLOC, and 4341 as destination RLOC and port. The ITR must pick a random source port to use for all

outbound LISP data traffic in order to avoid creating excessive state in the NAT.

6. Security Considerations

By having the RTR relay the ECM-ed Map-Register message from an ETR to its Map-Server, the RTR can restrict access to the RTR services, only to those ETRs that are registered with a given Map-Server. To do so, the RTR and the Map-Server may be configured with a shared key that is used to authenticate the origin and to protect the integrity of the Map-Notify messages sent by the Map Server to the RTR. This prevents an on-path attacker from impersonating the Map-Server to the RTR, and allows the RTR to cryptographically verify that the ETR is properly registered with the Map-Server.

Having the RTR re-encapsulate traffic only when the source or the destination are registered EIDs, protects against the adverse use of an RTR for EID spoofing.

Upon receiving a Data-Map-Notify, an xTR can authenticate the origin of the Map-Notify message using the key that the ETR shares with the Map-Server. This enables the ETR to verify that the ECM-ed Map-Register was indeed forwarded by the RTR to the Map-Server, and was accepted by the Map-Server.

6.1. Acknowledgments

The authors would like to thank Noel Chiappa, Alberto Rodriguez Natal, Lorand Jakab, Albert Cabellos, Dominik Klein, Matthias Hartmann, and Michael Menth for their previous work, feedback and helpful suggestions.

7. IANA Considerations

This document does not request any IANA actions.

8. Normative References

[I-D.ietf-lisp-eid-mobility]
Comeras, M. P., Ashtaputre, V., Moreno, V., Maino, F., and D. Farinacci, "LISP L2/L3 EID Mobility Using a Unified Control Plane", draft-ietf-lisp-eid-mobility-07 (work in progress), January 2021.

[I-D.ietf-lisp-pubsub]
Rodriguez-Natal, A., Ermagan, V., Cabellos, A., Barkai, S., and M. Boucadair, "Publish/Subscribe Functionality for LISP", draft-ietf-lisp-pubsub-08 (work in progress), February 2021.

[I-D.ietf-lisp-rfc6830bis]
Farinacci, D., Fuller, V., Meyer, D., Lewis, D., and A. Cabellos, "The Locator/ID Separation Protocol (LISP)", draft-ietf-lisp-rfc6830bis-36 (work in progress), November 2020.

[I-D.ietf-lisp-rfc6833bis]
Farinacci, D., Maino, F., Fuller, V., and A. Cabellos-Aparicio, "Locator/ID Separation Protocol (LISP) Control-Plane", draft-ietf-lisp-rfc6833bis-30 (work in progress), November 2020.

[ICE] Rosenberg, J., "Interactive Connectivity Establishment (ICE)", RFC rfc5245, October 2008.

[LCAF] Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format (LCAF)", RFC 8060, December 2015.

[NAT] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

[NAT-MN] Klein, D., Hartmann, M., and M. Menth, "NAT traversal for LISP mobile node, In Proceedings of the Re-Architecting the Internet Workshop (ReARCH '10).", 2010.

[STUN] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC rfc5389, October 2008.

[TURN] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN)", RFC rfc5766, April 2010.

Authors' Addresses

Vina Ermagan
Google

Email: ermagan@gmail.com

Dino Farinacci
lispers.net

Email: farinacci@gmail.com

Darrel Lewis
Cisco Systems, Inc.

Email: darlewis@cisco.com

Fabio Maino
Cisco Systems, Inc.

Email: fmaino@cisco.com

Marc Portoles Comeras
Cisco Systems, Inc.

Email: mportole@cisco.com

Jesper Skriver
Arista

Email: jesper@skriver.dk

Chris White
Logicalelegance, Inc.

Email: chris@logicalelegance.com

Albert Lopez
UPC/BarcelonaTech

Email: alopez@ac.upc.edu

Albert Cabellos
UPC/BarcelonaTech

Email: acabello@ac.upc.edu