Internet Engineering Task Force (IETF)
Request for Comments: 6834
Category: Experimental
ISSN: 2070-1721

L. Iannone
Telecom ParisTech
D. Saucez
INRIA Sophia Antipolis
O. Bonaventure
Universite catholique de Louvain

January 2013

Locator/ID Separation Protocol (LISP) Map-Versioning

Abstract

This document describes the LISP (Locator/ID Separation Protocol) Map-Versioning mechanism, which provides in-packet information about Endpoint ID to Routing Locator (EID-to-RLOC) mappings used to encapsulate LISP data packets. The proposed approach is based on associating a version number to EID-to-RLOC mappings and the transport of such a version number in the LISP-specific header of LISP-encapsulated packets. LISP Map-Versioning is particularly useful to inform communicating Ingress Tunnel Routers (ITRs) and Egress Tunnel Routers (ETRs) about modifications of the mappings used to encapsulate packets. The mechanism is transparent to implementations not supporting this feature, since in the LISP-specific header and in the Map Records, bits used for Map-Versioning can be safely ignored by ITRs and ETRs that do not support the mechanism.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for examination, experimental implementation, and evaluation.

This document defines an Experimental Protocol for the Internet community. This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6834.

Table of Contents

1. Introduction

This document describes the Map-Versioning mechanism used to provide information on changes in the EID-to-RLOC (Endpoint ID to Routing Locator) mappings used in the LISP (Locator/ID Separation Protocol [RFC6830]) context to perform packet encapsulation. The mechanism is totally transparent to xTRs (Ingress and Egress Tunnel Routers) not supporting such functionality. It is not meant to replace any existing LISP mechanisms but rather to extend them by providing new functionalities. If for any unforeseen reason a normative conflict between this document and the LISP main specifications is found, the latter ([RFC6830]) has precedence over this document.

The basic mechanism is to associate a Map-Version number to each LISP EID-to-RLOC mapping and transport such a version number in the LISP-specific header. When a mapping changes, a new version number is assigned to the updated mapping. A change in an EID-to-RLOC mapping can be a change in the RLOCs set, by adding or removing one or more RLOCs, but it can also be a change in the priority or weight of one or more RLOCs.

When Map-Versioning is used, LISP-encapsulated data packets contain the version number of the two mappings used to select the RLOCs in the outer header (i.e., both source and destination). These version numbers are encoded in the 24 low-order bits of the first longword of the LISP header and indicated by a specific bit in the flags (first 8 high-order bits of the first longword of the LISP header). Note that not all packets need to carry version numbers.

When an ITR (Ingress Tunnel Router) encapsulates a data packet, with a LISP header containing the Map-Version numbers, it puts in the LISP-specific header two version numbers:

1. The version number assigned to the mapping (contained in the EID-to-RLOC Database) used to select the source RLOC.

2. The version number assigned to the mapping (contained in the EID-to-RLOC Cache) used to select the destination RLOC.

This operation is two-fold. On the one hand, it enables the ETR (Egress Tunnel Router) receiving the packet to know if the ITR has the latest version number that any ETR at the destination EID site has provided to the ITR in a Map-Reply. If this is not the case, the ETR can send to the ITR a Map-Request containing the updated mapping or solicit a Map-Request from the ITR (both cases are already defined in [RFC6830]). In this way, the ITR can update its EID-to-RLOC Cache. On the other hand, it enables an ETR receiving such a packet

to know if it has in its EID-to-RLOC Cache the latest mapping for the source EID (in the case of bidirectional traffic). If this is not the case, a Map-Request can be sent.

Issues and concerns about the deployment of LISP for Internet traffic are discussed in [RFC6830]. Section 11 provides additional issues and concerns raised by this document. In particular, Section 11.1 provides details about the ETRs' synchronization issue in the context of Map-Versioning.

2. Requirements Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Definitions of Terms

This document uses terms already defined in the main LISP specification [RFC6830]. Here, we define the terms that are specific to the Map-Versioning mechanism. Throughout the whole document, Big Endian bit ordering is used.

Map-Version number: An unsigned 12-bit integer is assigned to an EID-to-RLOC mapping, not including the value 0 (0x000).

Null Map-Version: The 12-bit null value of 0 (0x000) is not used as a Map-Version number. It is used to signal that no Map-Version number is assigned to the EID-to-RLOC mapping.

Source Map-Version number: This Map-Version number of the EID-to-RLOC mapping is used to select the source address (RLOC) of the outer IP header of LISP-encapsulated packets.

Destination Map-Version number: This Map-Version number of the EID-to-RLOC mapping is used to select the destination address (RLOC) of the outer IP header of LISP-encapsulated packets.

4. EID-to-RLOC Map-Version Number

The EID-to-RLOC Map-Version number consists of an unsigned 12-bit integer. The version number is assigned on a per-mapping basis, meaning that different mappings have a different version number, which is also updated independently. An update in the version number (i.e., a newer version) consists of incrementing by one the older version number. Appendix A contains a rough estimation of the wrap-around time for the Map-Version number.

The space of version numbers has a circular order where half of the version numbers are greater (i.e., newer) than the current Map-Version number and the other half of the version numbers are smaller (i.e., older) than the current Map-Version number. In a more formal way, assuming that we have two version numbers V1 and V2 and that the numbers are expressed in N bits, the following steps MUST be performed (in the same order as shown below) to strictly define their order:

1. V1 = V2 : The Map-Version numbers are the same.

2. V2 > V1 : if and only if

   V2 > V1 AND (V2 - V1) <= 2**(N-1)

   OR

   V1 > V2 AND (V1 - V2) > 2**(N-1)

3. V1 > V2 : otherwise.

Using 12 bits, as defined in this document, and assuming a Map-Version value of 69, Map-Version numbers in the range [70; 69 + 2048] are greater than 69, while Map-Version numbers in the range [69 + 2049; (69 + 4096) mod 4096] are smaller than 69.

Map-Version numbers are assigned to mappings by configuration. The initial Map-Version number of a new EID-to-RLOC mapping SHOULD be assigned randomly, but it MUST NOT be set to the Null Map-Version value (0x000), because the Null Map-Version number has a special meaning (see Section 4.1).

Upon reboot, an ETR will use mappings configured in its EID-to-RLOC Database. If those mappings have a Map-Version number, it will be used according to the mechanisms described in this document. ETRs MUST NOT automatically generate and assign Map-Version numbers to mappings in the EID-to-RLOC Database.

4.1. The Null Map-Version

The value 0x000 (zero) is not a valid Map-Version number indicating the version of the EID-to-RLOC mapping. Such a value is used for special purposes and is named the Null Map-Version number.

The Null Map-Version MAY appear in the LISP-specific header as either a Source Map-Version number (cf. Section 5.2) or a Destination Map-Version number (cf. Section 5.1). When the Source Map-Version number is set to the Null Map-Version value, it means that no map

version information is conveyed for the source site. This means that if a mapping exists for the source EID in the EID-to-RLOC Cache, then the ETR MUST NOT compare the received Null Map-Version with the content of the EID-to-RLOC Cache. When the Destination Map-Version number is set to the Null Map-Version value, it means that no map version information is conveyed for the destination site. This means that the ETR MUST NOT compare the value with the Map-Version number of the mapping for the destination EID present in the EID-to-RLOC Database.

The other use of the Null Map-Version number is in the Map Records, which are part of the Map-Request, Map-Reply, and Map-Register messages (defined in [RFC6830]). Map Records that have a Null Map-Version number indicate that there is no Map-Version number associated with the mapping. This means that LISP-encapsulated packets destined to the EID-Prefix referred to by the Map Record MUST either not contain any Map-Version numbers (V-bit set to 0) or, if they contain Map-Version numbers (V-bit set to 1), then the destination Map-Version number MUST be set to the Null Map-Version number. Any value different from zero means that Map-Versioning is supported and MAY be used.

The fact that the 0 value has a special meaning for the Map-Version number implies that, when updating a Map-Version number because of a change in the mapping, if the next value is 0, then the Map-Version number MUST be incremented by 2 (i.e., set to 1, which is the next valid value).

5. Dealing with Map-Version Numbers

The main idea of using Map-Version numbers is that whenever there is a change in the mapping (e.g., adding/removing RLOCs, a change in the weights due to Traffic Engineering policies, or a change in the priorities) or a LISP site realizes that one or more of its own RLOCs are not reachable anymore from a local perspective (e.g., through IGP, or policy changes) the LISP site updates the mapping, also assigning a new Map-Version number.

To each mapping, a version number is associated and changes each time the mapping is changed. Note that Map-Versioning does not introduce new problems concerning the coordination of different ETRs of a domain. Indeed, ETRs belonging to the same LISP site must return for a specific EID-Prefix the same mapping, including the same Map-Version number. In principle, this is orthogonal to whether or not Map-Versioning is used. The synchronization problem and its implication on the traffic are out of the scope of this document (see Section 11).

In order to announce in a data-driven fashion that the mapping has been updated, Map-Version numbers used to create the outer IP header of the LISP-encapsulated packet are embedded in the LISP-specific header. This means that the header needs to contain two Map-Version numbers:

- The Source Map-Version number of the EID-to-RLOC mapping in the EID-to-RLOC Database used to select the source RLOC.

- The Destination Map-Version number of the EID-to-RLOC mapping in the EID-to-RLOC Cache used to select the destination RLOC.

By embedding both the Source Map-Version number and the Destination Map-Version number, an ETR receiving a LISP packet with Map-Version numbers can perform the following checks:

1. The ITR that has sent the packet has an up-to-date mapping in its EID-to-RLOC Cache for the destination EID and is performing encapsulation correctly.

2. In the case of bidirectional traffic, the mapping in the local ETR EID-to-RLOC Cache for the source EID is up to date.

If one or both of the above conditions do not hold, the ETR can send a Map-Request either to make the ITR aware that a new mapping is available (see Section 5.1) or to update the mapping in the local EID-to-RLOC Cache (see Section 5.2).

5.1. Handling Destination Map-Version Number

When an ETR receives a packet, the Destination Map-Version number relates to the mapping for the destination EID for which the ETR is an RLOC. This mapping is part of the ETR EID-to-RLOC Database. Since the ETR is authoritative for the mapping, it has the correct and up-to-date Destination Map-Version number. A check on this version number can be done, where the following cases can arise:

1. The packet arrives with the same Destination Map-Version number stored in the EID-to-RLOC Database. This is the regular case. The ITR sending the packet has in its EID-to-RLOC Cache an up-to-date mapping. No further actions are needed.

2. The packet arrives with a Destination Map-Version number greater (i.e., newer) than the one stored in the EID-to-RLOC Database. Since the ETR is authoritative on the mapping, meaning that the Map-Version number of its mapping is the correct one, this implies that someone is not behaving correctly with respect to

the specifications. In this case, the packet carries a version number that is not valid; otherwise, the ETR would have the same number, and the packet SHOULD be silently dropped.

3. The packets arrive with a Destination Map-Version number smaller (i.e., older) than the one stored in the EID-to-RLOC Database. This means that the ITR sending the packet has an old mapping in its EID-to-RLOC Cache containing stale information. The ETR MAY choose to normally process the encapsulated datagram according to [RFC6830]; however, the ITR sending the packet has to be informed that a newer mapping is available. This is done with a Map-Request message sent back to the ITR. The Map-Request will either trigger a Map-Request back using the Solicit-Map-Request (SMR) bit or it will piggyback the newer mapping. These are not new mechanisms; how to use the SMR bit or how to piggyback mappings in Map-Request messages is already described in [RFC6830], while their security is discussed in [LISP-THREATS]. These Map-Request messages should be rate-limited (rate-limitation policies are also described in [RFC6830]). The feature introduced by Map-Version numbers is the possibility of blocking traffic not using the latest mapping. Indeed, after a certain number of retries, if the Destination Map-Version number in the packets is not updated, the ETR MAY drop packets with a stale Map-Version number while strongly reducing the rate of Map-Request messages. This is because either the ITR is refusing to use the mapping for which the ETR is authoritative, or (worse) it might be some form of attack. Another case might be that the control plane is experiencing transient failures, so the Map-Requests cannot reach that ITR. By continually sending Map-Requests at a very low rate, it is possible to recover from this situation.

The rule in the third case MAY be more restrictive. If the mapping has been the same for a period of time as long as the Time to Live (TTL) (defined in [RFC6830]) of the previous version of the mapping, all packets arriving with an old Map-Version SHOULD be silently dropped right away without issuing any Map-Request. Such action is permitted because if the new mapping with the updated version number has been unchanged for at least the same time as the TTL of the older mapping, all the entries in the EID-to-RLOC Caches of ITRs must have expired. Hence, all ITRs sending traffic should have refreshed the mapping according to [RFC6830]. If packets with old Map-Version numbers are still received, then either someone has not respected the TTL or it is a form of spoof/attack. In both cases, this is not valid behavior with respect to the specifications and the packet SHOULD be silently dropped.

LISP-encapsulated packets with the V-bit set, when the original mapping in the EID-to-RLOC Database has the version number set to the Null Map-Version value, MAY be silently dropped. As explained in Section 4.1, if an EID-to-RLOC mapping has a Null Map-Version, it means that ITRs, using the mapping for encapsulation, MUST NOT use a Map-Version number in the LISP-specific header.

For LISP-encapsulated packets with the V-bit set, when the original mapping in the EID-to-RLOC Database has the version number set to a value different from the Null Map-Version value, a Destination Map-Version number equal to the Null Map-Version value means that the Destination Map-Version number MUST be ignored.

5.2. Handling Source Map-Version Number

When an ETR receives a packet, the Source Map-Version number relates to the mapping for the source EID for which the ITR that sent the packet is authoritative. If the ETR has an entry in its EID-to-RLOC Cache for the source EID, then a check can be performed and the following cases can arise:

1. The packet arrives with the same Source Map-Version number as that stored in the EID-to-RLOC Cache. This is the correct regular case. The ITR has in its EID-to-RLOC Cache an up-to-date copy of the mapping. No further actions are needed.

2. The packet arrives with a Source Map-Version number greater (i.e., newer) than the one stored in the local EID-to-RLOC Cache. This means that the ETR has in its EID-to-RLOC Cache a mapping that is stale and needs to be updated. A Map-Request SHOULD be sent to get the new mapping for the source EID. This is a normal Map-Request message sent through the mapping system and MUST respect the specifications in [RFC6830], including rate-limitation policies.

3. The packet arrives with a Source Map-Version number smaller (i.e., older) than the one stored in the local EID-to-RLOC Cache. Such a case is not valid with respect to the specifications. Indeed, if the mapping is already present in the EID-to-RLOC Cache, this means that an explicit Map-Request has been sent and a Map-Reply has been received from an authoritative source. Assuming that the mapping system is not corrupted, the Map-Version in the EID-to-RLOC Cache is the correct one, while the one carried by the packet is stale. In this situation, the packet MAY be silently dropped.

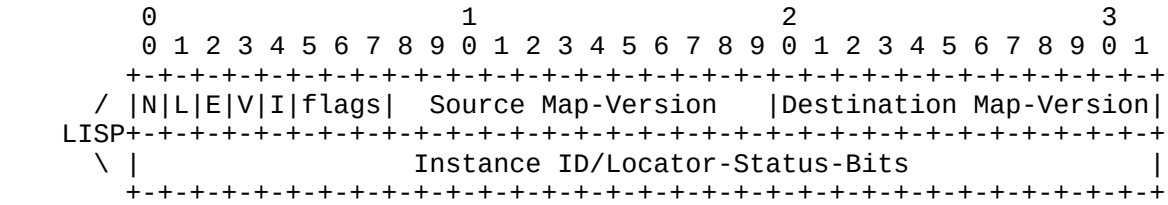
If the ETR does not have an entry in the EID-to-RLOC Cache for the source EID (e.g., in the case of unidirectional traffic), then the Source Map-Version number can be safely ignored.

For LISP-encapsulated packets with the V-bit set, if the Source Map-Version number is the Null Map-Version value, it means that the Source Map-Version number MUST be ignored.

6. LISP Header and Map-Version Numbers

In order for the versioning approach to work, the LISP-specific header has to carry both the Source Map-Version number and Destination Map-Version number. This is done by setting the V-bit in the LISP-specific header as defined in [RFC6830] Section 5.3. When the V-bit is set, the low-order 24 bits of the first longword are used to transport both the source and destination Map-Version numbers. In particular, the first 12 bits are used for the Source Map-Version number and the second 12 bits for the Destination Map-Version number.

Below is an example of a LISP header carrying version numbers in the case of IPv4-in-IPv4 encapsulation. The same setting can be used for any other case (IPv4-in-IPv6, IPv6-in-IPv4, and IPv6-in-IPv6).



Source Map-Version number (12 bits): Map-Version of the mapping used by the ITR to select the RLOC present in the 'Source Routing Locator' field. Section 5.2 describes how to set this value on transmission and handle it on reception.

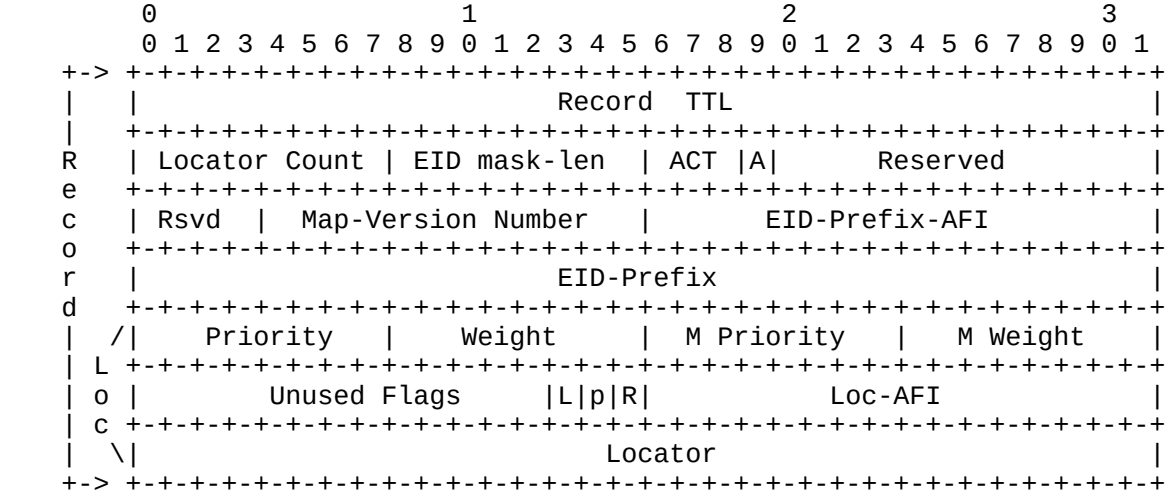
Destination Map-Version number (12 bits): Map-Version of the mapping used by the ITR to select the RLOC present in the 'Destination Routing Locator' field. Section 5.1 describes how to set this value on transmission and handle it on reception.

This document only specifies how to use the low-order 24 bits of the first longword of the LISP-specific header when the V-bit is set to 1. All other cases, including the bit fields of the rest of the LISP-specific header and the whole LISP packet format, are specified in [RFC6830]. Not all of the LISP-encapsulated packets need to carry

version numbers. When Map-Version numbers are carried in these packets, the V-bit MUST be set to 1. All permissible combinations of the flags when the V-bit is set to 1 are described in [RFC6830].

7. Map Record and Map-Version

To accommodate the proposed mechanism, the Map Records that are transported in Map-Request/Map-Reply/Map-Register messages need to carry the Map-Version number as well. For this purpose, the 12 bits before the 'EID-Prefix-AFI' field in the Record that describes a mapping are used. This is defined in Section 6.1.4 of [RFC6830] and reported here as an example.



Map-Version Number: Map-Version of the mapping contained in the Record. As explained in Section 4.1, this field can be zero (0), meaning that no Map-Version is associated to the mapping; hence, packets that are LISP encapsulated using this mapping MUST NOT contain Map-Version numbers in the LISP-specific header, and the V-bit MUST be set to 0.

This packet format works perfectly with xTRs that do not support Map-Versioning, since they can simply ignore those bits.

8. Benefits and Case Studies for Map-Versioning

In the following sections, we provide more discussion on various aspects and uses of Map-Versioning. Security observations are grouped in Section 10.

8.1. Map-Versioning and Unidirectional Traffic

When using Map-Versioning, the LISP-specific header carries two Map-Version numbers, for both source and destination mappings. This can raise the question on what will happen in the case of unidirectional flows, for instance, in the case presented in Figure 1, since the LISP specification does not mandate that the ETR have a mapping for the source EID.

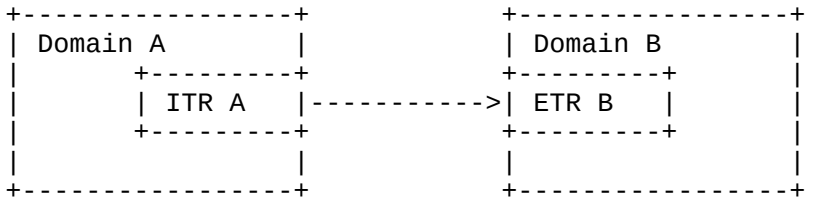


Figure 1: Unidirectional Traffic between LISP Domains

In the case of the ITR, the ITR is able to put both the source and destination version number in the LISP header, since the Source Map-Version number is in the ITR's database, while the Destination Map-Version number is in the ITR's cache.

In the case of the ETR, the ETR simply checks only the Destination Map-Version number in the same way as that described in Section 5, ignoring the Source Map-Version number.

8.2. Map-Versioning and Interworking

Map-Versioning is compatible with the LISP interworking between LISP and non-LISP sites as defined in [RFC6832]. LISP interworking defines three techniques to make LISP sites and non-LISP sites, namely Proxy-ITR, LISP-NAT, and Proxy-ETR. The following text describes how Map-Versioning relates to these three mechanisms.

8.2.1. Map-Versioning and Proxy-ITRs

The purpose of the Proxy-ITR (PITR) is to encapsulate traffic originating in a non-LISP site in order to deliver the packet to one of the ETRs of the LISP site (cf. Figure 2). This case is very similar to the unidirectional traffic case described in Section 8.1; hence, similar rules apply.

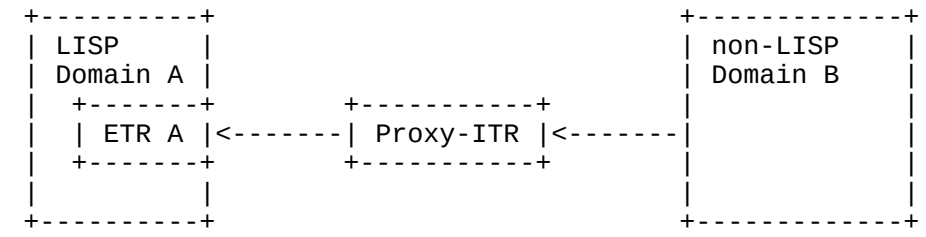


Figure 2: Unidirectional Traffic from Non-LISP Domain to LISP Domain

The main difference is that a Proxy-ITR does not have any mapping, since it just encapsulates packets arriving from the non-LISP site and thus cannot provide a Source Map-Version. In this case, the Proxy-ITR will just put the Null Map-Version value as the Source Map-Version number, while the receiving ETR will ignore the field.

With this setup, LISP Domain A is able to check whether or not the PITR is using the latest mapping. If this is not the case, the mapping for LISP Domain A on the PITR can be updated using one of the mechanisms defined in [RFC6830] and [RFC6832].

8.2.2. Map-Versioning and LISP-NAT

The LISP-NAT mechanism is based on address translation from non-routable EIDs to routable EIDs and does not involve any form of encapsulation. As such, Map-Versioning does not apply in this case.

8.2.3. Map-Versioning and Proxy-ETRs

The purpose of the Proxy-ETR (PETR) is to decapsulate traffic originating in a LISP site in order to deliver the packet to the non-LISP site (cf. Figure 3). One of the main reasons to deploy PETRs is to bypass uRPF (Unicast Reverse Path Forwarding) checks on the provider edge.

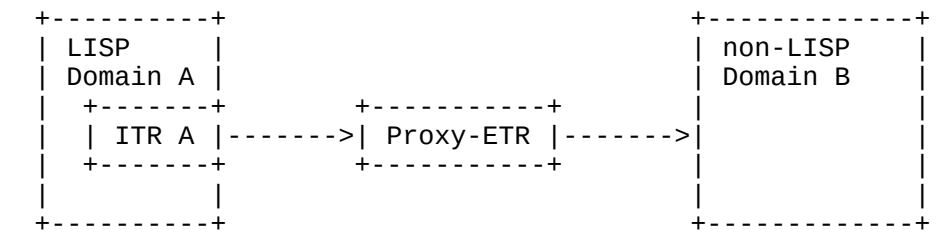


Figure 3: Unidirectional Traffic from LISP Domain to Non-LISP Domain

A Proxy-ETR does not have any mapping, since it just decapsulates packets arriving from the LISP site. In this case, the ITR will just put the Null Map-Version value as the Destination Map-Version number, while the receiving Proxy-ETR will ignore the field.

With this setup, the Proxy-ETR is able to check whether or not the mapping has changed. If this is the case, the mapping for LISP Domain A on the PETR can be updated using one of the mechanisms defined in [RFC6830] and [RFC6832].

8.3. RLOC Shutdown/Withdraw

Map-Versioning can also be used to perform a graceful shutdown or withdraw of a specific RLOC. This is achieved by simply issuing a new mapping, with an updated Map-Version number where the specific RLOC to be shut down is withdrawn or announced as unreachable (via the R-bit in the Map Record; see [RFC6830]), but without actually turning it off.

Once no more traffic is received by the RLOC, it can be shut down gracefully, because all sites actively using the mapping have updated it.

It should be pointed out that for frequent up/down changes such a mechanism should not be used, since this can generate excessive load on the mapping system.

8.4. Map-Version for Lightweight LISP Implementation

The use of Map-Versioning can help in developing a lightweight implementation of LISP. However, this comes with the price of not supporting the Locator-Status-Bit, which is useful in some contexts.

In the current LISP specifications, the set of RLOCs must always be maintained ordered and consistent with the content of the Locator-Status-Bits (see Section 6.5 of [RFC6830]). With Map-Versioning, such types of mechanisms can be avoided. When a new RLOC is added to a mapping, it is not necessary to "append" new Locators to the existing ones as explained in Section 6.5 of [RFC6830]. A new mapping with a new Map-Version number will be issued, and since the old Locators are still valid, the transition will occur with no disruptions. The same applies for the case where an RLOC is withdrawn. There is no need to maintain holes in the list of Locators, as is the case when using Locator-Status-Bits, for sites that are not using the RLOC that has been withdrawn; in this case, the transition will occur with no disruptions.

All of these operations, as already stated, do not need to maintain any consistency among Locator-Status-Bits and in the way that the RLOCs are stored in the EID-to-RLOC Cache.

Further, Map-Versioning can be used as a substitute for the "clock sweep" operation described in Section 6.6.1 of [RFC6830]. Indeed, every LISP site communicating to a specific LISP site that has updated the mapping will be informed of the available new mapping in a data-driven manner.

Note that what is proposed in this section is just an example and MUST NOT be considered as specifications for a lightweight LISP implementation. If the IETF decides to undertake such work, it will be documented elsewhere.

9. Incremental Deployment and Implementation Status

Map-Versioning can be incrementally deployed without any negative impact on existing LISP elements (e.g., xTRs, Map-Servers, Proxy-ITRs, etc.). Any LISP element that does not support Map-Versioning can safely ignore Map-Version numbers carried in the LISP header. Further, there is no need of any specific mechanism to discover whether or not an xTR supports Map-Versioning. This information is already included in the Map Record.

Map-Versioning is currently implemented in OpenLISP [OPENLISP].

Note that the reference document for LISP implementations and interoperability tests remains [RFC6830].

10. Security Considerations

Map-Versioning does not introduce any security issues concerning both the data plane and the control plane. On the contrary, as described below, if Map-Versioning may also be used to update mappings in the case of change in the reachability information (i.e., instead of the Locator-Status-Bits), it is possible to reduce the effects of some DoS or spoofing attacks that can happen in an untrusted environment.

Robustness of the Map-Versioning mechanism leverages on a trusted Mapping Distribution System. A thorough security analysis of LISP is documented in [LISP-THREATS].

10.1. Map-Versioning against Traffic Disruption

An attacker can try to disrupt ongoing communications by creating LISP-encapsulated packets with wrong Locator-Status-Bits. If the xTR blindly trusts the Locator-Status-Bits, it will change the encapsulation accordingly, which can result in traffic disruption.

This does not happen in the case of Map-Versioning. As described in Section 5, upon a version number change the xTR first issues a Map-Request. The assumption is that the mapping distribution system is sufficiently secure that Map-Request and Map-Reply messages and their content can be trusted. Security issues concerning specific mapping distribution systems are out of the scope of this document. In the case of Map-Versioning, the attacker should "guess" a valid version number that triggers a Map-Request as described in Section 5; otherwise, the packet is simply dropped. Nevertheless, guessing a version number that generates a Map-Request is easy; hence, it is important to follow the rate-limitation policies described in [RFC6830] in order to avoid DoS attacks.

Note that a similar level of security can be obtained with Locator-Status-Bits by simply making it mandatory to verify any change through a Map-Request. However, in this case Locator-Status-Bits lose their meaning, because it does not matter anymore which specific bits have changed; the xTR will query the mapping system and trust the content of the received Map-Reply. Furthermore, there is no way to perform filtering as in Map-Versioning in order to drop packets that do not carry a valid Map-Version number. In the case of Locator-Status-Bits, any random change can trigger a Map-Request (unless rate limitation is enabled, which raises another type of attack as discussed in Section 10.2).

10.2. Map-Versioning against Reachability Information DoS

Attackers can try to trigger a large amount of Map-Requests by simply forging packets with random Map-Versions or random Locator-Status-Bits. In both cases, the Map-Requests are rate-limited as described in [RFC6830]. However, in contrast to the Locator-Status-Bit, where there is no filtering possible, in the case of Map-Versioning it is possible to filter invalid version numbers before triggering a Map-Request, thus helping to reduce the effects of DoS attacks. In other words, the use of Map-Versioning enables a fine control on when to update a mapping or when to notify someone that a mapping has been updated.

It is clear that Map-Versioning does not protect against DoS and DDoS attacks, where an xTR loses processing power when doing checks on the LISP header of packets sent by attackers. This is independent of Map-Versioning and is the same for Locator-Status-Bits.

11. Open Issues and Considerations

There are a number of implications of the use of Map-Versioning that are not yet completely explored. Among these are:

- Performance of the convergence time when an EID-to-RLOC mapping changes, i.e., how much time is needed to update mappings in the EID-to-RLOC Cache of the ITRs currently sending traffic to ETRs for the EID whose mapping has been changed.

- Support for ETR synchronization. The implications that a temporary lack of synchronization may have on the traffic are yet to be fully explored. Details on how to maintain synchronization are presented in Section 6.6 of [RFC6830]. Section 11.1 discusses the issue in further detail with respect to the Map-Versioning mechanism.

The authors expect that experimentation will help assess the performance and limitations of the Map-Versioning mechanism. Issues and concerns about the deployment of LISP for Internet traffic are discussed in [RFC6830].

11.1. Lack of Synchronization among ETRs

Even without Map-Versioning, LISP ([RFC6830]) requires ETRs to announce the same mapping for the same EID-Prefix to a requester. The implications that a temporary lack of synchronization may have on the traffic are yet to be fully explored.

Map-Versioning does not require additional synchronization mechanisms as compared to the normal functioning of LISP without Map-Versioning. Clearly, all the ETRs have to reply with the same Map-Version number; otherwise, there can be an inconsistency that creates additional control traffic, instabilities, and traffic disruptions. It is the same without Map-Versioning, with ETRs that have to reply with the same mapping; otherwise, the same problems can arise.

There are two ways Map-Versioning is helpful with respect to the synchronization problem. On the one hand, assigning version numbers to mappings helps in debugging, since quick checks on the consistency of the mappings on different ETRs can be done by looking at the Map-Version number. On the other hand, Map-Versioning can be used to control the traffic toward ETRs that announce the latest mapping.

As an example, let's consider the topology of Figure 4 where ITR A.1 of Domain A is sending unidirectional traffic to Domain B, while A.2 of Domain A exchanges bidirectional traffic with Domain B. In particular, ITR A.2 sends traffic to ETR B, and ETR A.2 receives traffic from ITR B.

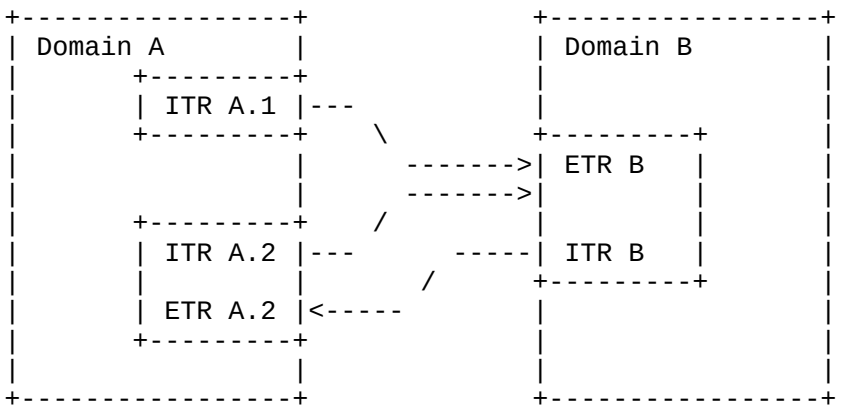


Figure 4: Example Topology

Obviously, in the case of Map-Versioning, both ITR A.1 and ITR A.2 of Domain A must use the same value; otherwise, the ETR of Domain B will start to send Map-Requests.

The same problem can, however, arise without Map-Versioning, for instance, if the two ITRs of Domain A send different Locator-Status-Bits. In this case, either the traffic is disrupted if ETR B trusts the Locator-Status-Bits, or if ETR B does not trust the Locator-Status-Bits it will start sending Map-Requests to confirm each change in reachability.

So far, LISP does not provide any specific synchronization mechanism but assumes that synchronization is provided by configuring the different xTRs consistently (see Section 6.6 in [RFC6830]). The same applies for Map-Versioning. If in the future any synchronization mechanism is provided, Map-Versioning will take advantage of it automatically, since it is included in the Record format, as described in Section 7.

12. Acknowledgments

The authors would like to thank Alia Atlas, Jesper Skriver, Pierre Francois, Noel Chiappa, and Dino Farinacci for their comments and review.

This work has been partially supported by the INFSO-ICT-216372 TRILOGY Project (http://www.trilogy-project.org).

13. References

13.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, January 2013.

[RFC6832] Lewis, D., Meyer, D., Farinacci, D., and V. Fuller, "Interworking between Locator/ID Separation Protocol (LISP) and Non-LISP Sites", RFC 6832, January 2013.

13.2. Informative References

[LISP-THREATS] Saucez, D., Iannone, L., and O. Bonaventure, "LISP Threats Analysis", Work in Progress, October 2012.

[OPENLISP] Iannone, L., Saucez, D., and O. Bonaventure, "Implementing the Locator/ID Separation Protocol: Design and experience", Computer Networks Vol. 55, Number 4, Pages 948-958, March 2011.

Appendix A. Estimation of Time before Map-Version Wrap-Around

This section proposes an estimation of the wrap-around time for the 12-bit size of the Map-Version number.

Using a granularity of seconds and assuming as worst case that a new version is issued each second, it takes slightly more than 1 hour before the version wraps around. Note that the granularity of seconds is in line with the rate-limitation policy for Map-Request messages, as proposed in the LISP main specifications ([RFC6830]).

Alternatively, a granularity of minutes can also be used, as for the TTL of the Map-Reply ([RFC6830]). In this case, the worst-case scenario is when a new version is issued every minute, leading to a much longer time before wrap-around. In particular, when using 12 bits, the wrap-around time is almost 3 days.

For general information, Figure 5 below provides a rough estimation of the time before wrap-around in the worst-case scenario, considering different sizes (length in bits) of the Map-Version number and different time granularities.

Since even in the case of a high mapping change rate (1 per second) the wrap-around time using 12 bits is far larger than any reasonable Round-Trip Time (RTT), there is no risk of race conditions.

Version Number Size (bits)	Time before Wrap-Around	
	Granularity: Minutes (mapping changes every 1 minute)	Granularity: Seconds (mapping changes every 1 second)
32	8171 years	136 years
30	2042 years	34 years
24	31 years	194 days
16	45 days	18 hours
15	22 days	9 hours
14	11 days	4 hours
13	5.6 days	2.2 hours
12	2.8 days	1.1 hours

Figure 5: Estimation of Time before Wrap-Around

Authors' Addresses

Luigi Iannone
Telecom ParisTech

EMail: luigi.iannone@telecom-paristech.fr

Damien Saucez
INRIA Sophia Antipolis
2004 route des Lucioles - BP 93
Sophia Antipolis
France

EMail: damien.saucez@inria.fr

Olivier Bonaventure
Universite catholique de Louvain
Place St. Barbe 2
Louvain-la-Neuve
Belgium

EMail: olivier.bonaventure@uclouvain.be