

Network Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: September 12, 2011

L. Iannone  
TU Berlin - Deutsche Telekom  
Laboratories AG  
D. Saucez  
O. Bonaventure  
Universite catholique de Louvain  
March 11, 2011

**LISP Map-Versioning**  
**draft-ietf-lisp-map-versioning-01.txt**

**Abstract**

This document describes the LISP Map-Versioning mechanism, which provides in-packet information about 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 transport such a version number in the LISP specific header of LISP-encapsulated packets. LISP Map-Versioning is particularly useful to inform communicating xTRs about modifications of the mappings used to encapsulate packets. The mechanism is transparent to legacy implementations, since in the LISP-specific header and in the Map Records, bits used for Map-Versioning can be safely ignored by xTRs that do not support the mechanism.

**Status of this Memo**

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on September 12, 2011.

## Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the BSD License.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">4</a>
<a href="#">2.</a>	Requirements Notation . . . . .	<a href="#">5</a>
<a href="#">3.</a>	Definitions of Terms . . . . .	<a href="#">5</a>
<a href="#">4.</a>	EID-to-RLOC Map-Version number . . . . .	<a href="#">5</a>
<a href="#">4.1.</a>	The Null Map-Version . . . . .	<a href="#">6</a>
<a href="#">5.</a>	Dealing with Map-Version numbers . . . . .	<a href="#">7</a>
<a href="#">5.1.</a>	Handling Destination Map-Version number . . . . .	<a href="#">7</a>
<a href="#">5.2.</a>	Handling Source Map-Version number . . . . .	<a href="#">9</a>
<a href="#">6.</a>	LISP header and Map-Version numbers . . . . .	<a href="#">10</a>
<a href="#">7.</a>	Map Record and Map-Version . . . . .	<a href="#">11</a>
<a href="#">8.</a>	Benefits and case studies for Map-Versioning . . . . .	<a href="#">12</a>
<a href="#">8.1.</a>	Synchronization of different xTRs . . . . .	<a href="#">12</a>
<a href="#">8.2.</a>	Map-Versioning and unidirectional traffic . . . . .	<a href="#">13</a>
<a href="#">8.3.</a>	Map-Versioning and interworking . . . . .	<a href="#">13</a>
<a href="#">8.3.1.</a>	Map-Versioning and Proxy-ITRs . . . . .	<a href="#">14</a>
<a href="#">8.3.2.</a>	Map-Versioning and LISP-NAT . . . . .	<a href="#">14</a>
<a href="#">8.3.3.</a>	Map-Versioning and Proxy-ETRs . . . . .	<a href="#">14</a>
<a href="#">8.4.</a>	RLOC shutdown/withdraw . . . . .	<a href="#">15</a>
<a href="#">8.5.</a>	Map-Version for lightweight LISP implementation . . . . .	<a href="#">15</a>
<a href="#">9.</a>	Incremental deployment and implementation status . . . . .	<a href="#">16</a>
<a href="#">10.</a>	Security Considerations . . . . .	<a href="#">16</a>
<a href="#">10.1.</a>	Map-Versioning against traffic disruption . . . . .	<a href="#">17</a>
<a href="#">10.2.</a>	Map-Versioning against reachability information DoS . . . . .	<a href="#">17</a>
<a href="#">11.</a>	IANA Considerations . . . . .	<a href="#">18</a>
<a href="#">12.</a>	Acknowledgements . . . . .	<a href="#">18</a>
<a href="#">13.</a>	References . . . . .	<a href="#">18</a>
<a href="#">13.1.</a>	Normative References . . . . .	<a href="#">18</a>
<a href="#">13.2.</a>	Informative References . . . . .	<a href="#">18</a>
<a href="#">Appendix A.</a>	Estimation of time before Map-Version wrap-around . . . . .	<a href="#">19</a>
<a href="#">Appendix B.</a>	Document Change Log . . . . .	<a href="#">20</a>



Authors' Addresses . . . . .	<a href="#">21</a>
------------------------------	--------------------

## 1. Introduction

This document describes the Map-Versioning mechanism used to provide information on changes in the EID-to-RLOC mappings used in the LISP ([\[I-D.ietf-lisp\]](#)) context to perform packet encapsulation. The mechanism is totally transparent to xTRs not supporting such functionality. It is not meant to replace any existing LISP mechanism, but rather to complete them providing new functionalities. 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 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 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 it is not the case the ETR can send to the ITR a Map-Request containing the updated mapping or soliciting a Map-Request from the ITR (both cases are already defined in [\[I-D.ietf-lisp\]](#)). In this way the ITR can update its 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 case of bidirectional traffic). If it is not the case a Map-Request can be send.



## **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**

The present document uses terms already defined in main LISP specification [[I-D.ietf-lisp](#)]. Hereafter are defined only the terms that are specific to the Map-Versioning mechanism.

Map-Version number: An unsigned 12-bits assigned to an EID-to-RLLOC mapping, not including the value 0 (0x000).

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

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

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

## **4. EID-to-RLLOC Map-Version number**

The EID-to-RLLOC Map-Version number consists in an unsigned 12-bits integer. The version number is assigned in a per-mapping fashion, meaning that different mappings will have assigned a different version number, which is also updated independently. An update in the version number (i.e., a newer version) consist in 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 is greater than the current Map-Version number and the other half is smaller than current Map-Version number. In a more formal way, assuming we have two version numbers V1 and V2 and that the numbers are expressed on N bits, the following three cases may happen:





$V1 = V2$  : This is the exact match case.

$V1 < V2$  : True if and only if  $V1 < V2 < (V1 + 2^{N-1})$ .

$V1 > V2$  : True if and only if  $V1 > V2 > (V1 - 2^{N-1})$ .

Using 12 bits, as defined in this document, and assuming a Map-Version value of 69, Map-Version numbers in the range [70; 69 + 2047] are greater than 69, while Map-Version numbers in the range [69 + 2048; (69 + 4095) mod 4096] are smaller than 69.

The initial Map-Version number of a new EID-to-RLOC mapping SHOULD be randomly generated. However, it MUST NOT be set to the Null Map-Version value (0x000), because it has a special meaning (see [Section 4.1](#)).

#### **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 Source Map-Version number (cf. [Section 5.2](#)) or 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 [\[I-D.ietf-lisp\]](#)). 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 the Map Record refers to, MUST NOT contain Map-Version numbers (i.e., V bit MUST always be 0). In other words, the Null Map-Version number signals to the ITR using the mapping that the Map-Versioning is not supported, or even if supported it MUST NOT be used for that specific EID-Prefix. 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 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 TE policies, or a change in the priorities) or an ISP 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 ISP updates the mapping also assigning a new Map-Version number.

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:

- o The Source Map-Version number of the EID-to-RLOC mapping in the EID-to-RLOC Database used to select the source RLOC.
- o 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 Source Map-Version number and 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 cache for the destination EID and is performing encapsulation correctly.
2. In 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 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 a



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 packets arrive 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, this means that someone is not behaving correctly w.r.t. the specifications, thus the packet carries a not valid version number and 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 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 SMR bit or it will piggyback the newer mapping. These are not new mechanisms; how to SMR or piggyback mappings in Map-Request messages is already described in [[I-D.ietf-lisp](#)], while their security is discussed in [[I-D.saucez-lisp-security](#)]. These Map-Request messages should be rate limited (rate limitation policies are also described in [[I-D.ietf-lisp](#)]). The feature introduced by Map-Version numbers is the possibility of blocking traffic from ITRs 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 silently drop packets with a stale Map-Version number. This 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.

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 TTL (defined in [[I-D.ietf-lisp](#)]) 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. The reason that allows such action is the fact that 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 caches of ITRs must have expired. Hence, all ITRs sending traffic should have refreshed the mapping according to [[I-D.ietf-lisp](#)]. If packets with old Map-



Version number 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 w.r.t. 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 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 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 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 stored in the EID-to-RLOC Cache. This is the correct regular case. The ITR has in its 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 ETR has in its 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 [[I-D.ietf-lisp](#)], 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 w.r.t. 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 anyhow, the Map-Version in the EID-to-RLOC Cache is the correct one and 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 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 Source Map-Version number and Destination Map-Version number. This is done by setting the V-bit in the LISP specific header. When the V-bit is set the low-order 24-bits of the first longword (which usually contains the nonce) are used to transport both source and destination Map-Version numbers. In particular the first 12 bits are used for Source Map-Version number and the second 12 bits for the Destination Map-Version number.

Hereafter is the example of 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).

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
/ |N|L|E|V|I|flags| Source Map-Version |Destination Map-Version|
LISP+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
\ |                               Instance ID/Locator Status Bits |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

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. How to set on transmission and handle on reception this value is described in [Section 5.2](#).

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. How to set on transmission and handle on reception this value is described in [Section 5.1](#).

The present document just 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



## 7. Map Record and Map-Version

This packet format works perfectly with xTRs that do not support Map-Versioning, since they can simply ignore those bits. Furthermore, existing and future mapping distribution protocol (e.g., ALT [[I-D.ietf-lisp-alt](#)]) are able to carry version numbers without needing any modification. The same applies to the LISP Map Server ([[I-D.ietf-lisp-ms](#)]), which will still work without any change since reserved bits are simply ignored.



## 8. Benefits and case studies for Map-Versioning

In the following sections we provide more discussion on various aspects and use of the Map-Versioning. Security observations are instead grouped in [Section 10](#).

### 8.1. Synchronization of different xTRs

Map-Versioning does not require additional synchronization mechanism 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, 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.

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

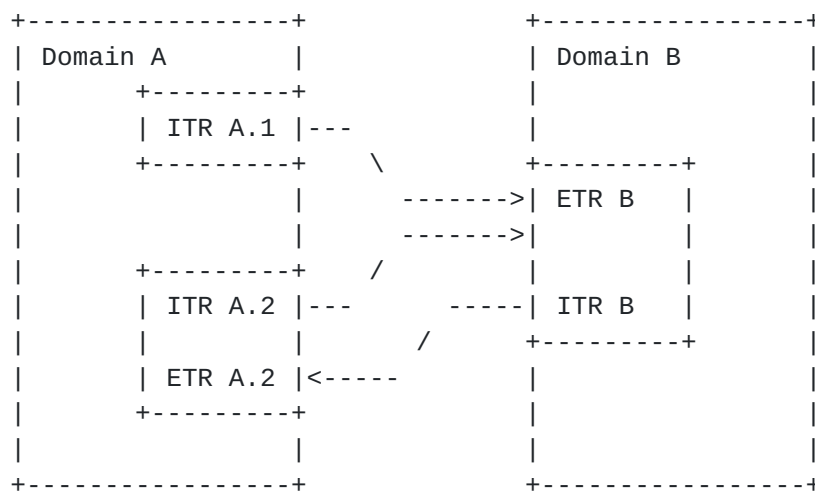


Figure 1

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 Loc Status Bits. In this case either the traffic is disrupted, if the ETR B trusts the Locator Status Bits, or if ETR B does not trusts the Locator Status



Bits it will start sending Map-Requests to confirm the each change in the reachability.

So far, LISP does not provide any specific synchronization mechanism, but assumes that synchronization is provided by configuring the different xTRs consistently. 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](#).

### **8.2. 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, like for instance in the case presented in Figure 2, since LISP specification do not mandate for ETR to have a mapping for the source EID.

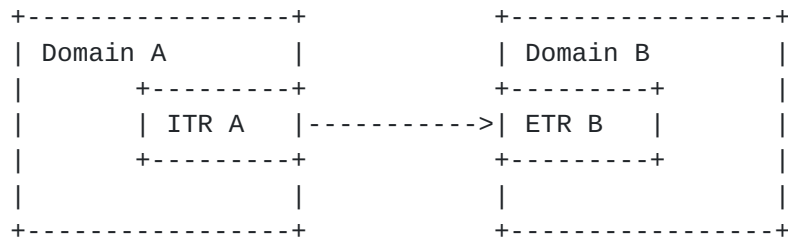


Figure 2

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

For what concerns the ETR, it simply checks only the Destination Map-Version number in the same way as described in [Section 5](#), ignoring the Source Map-Version number.

### **8.3. Map-Versioning and interworking**

Map-Versioning is compatible with the LISP interworking between LISP and non-LISP sites as defined in [[I-D.ietf-lisp-interworking](#)]. LISP interworking defines three techniques to make LISP sites and non-LISP sites, namely Proxy-ITR, LISP-NAT, and Proxy-ETR. Hereafter it is described how Map-Versioning relates to these three mechanisms.





### 8.3.1. 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 3). This case is very similar to the unidirectional traffic case described in [Section 8.2](#), hence similar rules apply.

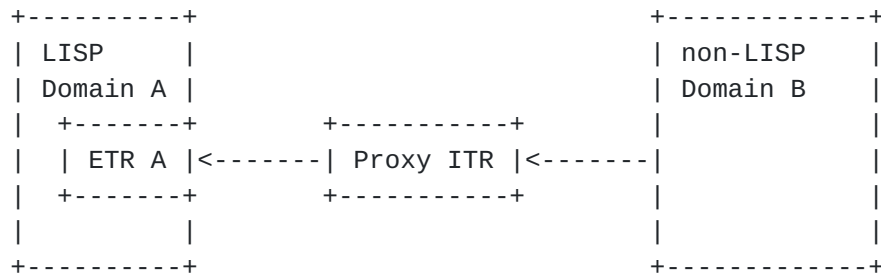


Figure 3

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

With this setup the 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 [\[I-D.ietf-lisp\]](#) and [\[I-D.ietf-lisp-interworking\]](#).

### 8.3.2. 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.

### 8.3.3. 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 4). One of the main reasons of deploy PETRs is to bypass uRPF (Unicast Reverse Path Forwarding) checks on the provider edge.





Figure 4

A Proxy-ETR does not have any mapping, since it just decapsulate packets arriving from LISP site. In this case, the ITR will just put the Null Map-Version value as 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 [[I-D.ietf-lisp](#)] and [[I-D.ietf-lisp-interworking](#)].

#### 8.4. RLOC shutdown/withdraw

Map-Versioning can be even 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 (R bit in the Map Record, see [[I-D.ietf-lisp](#)]), but without actually turning it off.

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

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.

#### 8.5. Map-Version for lightweight LISP implementation

The use of Map-Versioning can help in developing a lightweight implementation of LISP. This comes with the price of not supporting Loc-Status-Bit, which are 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 Loc Status Bits (see section 6.5 of [[I-D.ietf-lisp](#)]). With Map-Versioning such type 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 [[I-D.ietf-lisp](#)]. A new mapping with a new Map-Version number will be issued, and since the old locators are still valid the transition will be with no disruptions. The same applies for the case a 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 the transition will be with no disruptions.

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

Further, Map-Version can be used to substitute the "clock sweep" operation described in Section 6.5.1 of [[I-D.ietf-lisp](#)]. 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 proposed in the present section is just a case study and MUST NOT be considered as specification for a lightweight LISP implementation.

## **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 them. Further, there is no need of any specific mechanism to discover if an xTR supports or not Map-Versioning. This information is already included in the Map Record.

Map-Versioning is currently implemented in OpenLISP [[I-D.iannone-openlisp-implementation](#)].

Note that the reference document for LISP implementation and interoperability tests remains [[I-D.ietf-lisp](#)].

## **10. Security Considerations**

Map-Versioning does not introduce any new security issue concerning both the data-plane and the control-plane. On the contrary, as described in the following, if Map-Versioning may be used also to update mappings in 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.



A thorough security analysis of LISP is documented in [\[I-D.saucez-lisp-security\]](#).

### **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 system 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 limitations policies described in [\[I-D.ietf-lisp\]](#) in order to avoid DoS attacks.

Note that a similar level of security can be obtained with Loc Status Bits, by simply making 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 has 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 the 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 raise another type of attack discussed in [Section 10.2](#)).

### **10.2. Map-Versioning against reachability information DoS**

Attackers can try to trigger a large amount of Map-Request by simply forging packets with random Map-Version or random Locator Status Bits. In both cases the Map-Requests are rate limited as described in [\[I-D.ietf-lisp\]](#). However, differently from Locator Status Bit where there is no filtering possible, in the case of Map-Versioning is possible to filter not valid version numbers before triggering a Map-Request, thus helping in reducing 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 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 doing checks on the LISP header of packets sent by attackers. This is independent from Map-Versioning and is the same for Loc Status Bits.

## **11. IANA Considerations**

This document has no actions for IANA.

## **12. Acknowledgements**

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

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

## **13. References**

### **13.1. Normative References**

- [I-D.ietf-lisp]  
Farinacci, D., Fuller, V., Meyer, D., and D. Lewis,  
"Locator/ID Separation Protocol (LISP)",  
[draft-ietf-lisp-10](#) (work in progress), March 2011.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate  
Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

### **13.2. Informative References**

- [I-D.iannone-openlisp-implementation]  
Iannone, L., Saucez, D., and O. Bonaventure, "OpenLISP  
Implementation Report",  
[draft-iannone-openlisp-implementation-01](#) (work in  
progress), July 2008.
- [I-D.ietf-lisp-alt]  
Fuller, V., Farinacci, D., Meyer, D., and D. Lewis, "LISP  
Alternative Topology (LISP+ALT)", [draft-ietf-lisp-alt-06](#)  
(work in progress), March 2011.
- [I-D.ietf-lisp-interworking]  
Lewis, D., Meyer, D., Farinacci, D., and V. Fuller,  
"Interworking LISP with IPv4 and IPv6",



[draft-ietf-lisp-interworking-02](#) (work in progress),  
March 2011.

[I-D.ietf-lisp-ms]

Fuller, V. and D. Farinacci, "LISP Map Server",  
[draft-ietf-lisp-ms-07](#) (work in progress), March 2011.

[I-D.saucez-lisp-security]

Saucez, D., Iannone, L., and O. Bonaventure, "LISP  
Security Threats", [draft-saucez-lisp-security-03](#) (work in  
progress), March 2011.

#### **[Appendix A](#). Estimation of time before Map-Version wrap-around**

The present section proposes an estimation of the wrap-around time for the proposed 12 bits size for 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 ([\[I-D.ietf-lisp\]](#)). Alternatively a granularity of minutes can also be used, as for the TTL of the Map-Reply ([\[I-D.ietf-lisp\]](#)). In this case the worst 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, hereafter there is a table with a rough estimation of the time before wrap-around in the worst-case scenario, considering different sizes (bits length) of the Map-Version number and different time granularity.



Version Number		Time before wrap around			
Size (bits)					
		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

**Appendix B. Document Change Log**

- o Version 01 Posted March 2011.
  - \* Changed the wording from "Map-Version number 0" to "Null Map-Version".
  - \* Clarification of the use of the Null Map-Version value as Source Map-Version Number and Destination Map-Version Number.
  - \* Extended the section describing Map-Versioning and LISP Interworking co-existence.
  - \* Reduce packet format description to avoid double definitions with the main specs.
- o Version 00 Posted September 2010.
  - \* Added Section "Definitions of Terms".
  - \* Editorial polishing of all sections.
  - \* Added clarifications in section "Dealing with Map-Version numbers" for the case of the special Map-Version number 0.
  - \* Rename of [draft-iannone-mapping-versioning-02.txt](#).



Authors' Addresses

Luigi Iannone  
TU Berlin - Deutsche Telekom Laboratories AG  
Ernst-Reuter Platz 7  
Berlin  
Germany

Email: [luigi@net.t-labs.tu-berlin.de](mailto:luigi@net.t-labs.tu-berlin.de)

Damien Saucez  
Universite catholique de Louvain  
Place St. Barbe 2  
Louvain la Neuve  
Belgium

Email: [damien.saucez@uclouvain.be](mailto:damien.saucez@uclouvain.be)

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

Email: [olivier.bonaventure@uclouvain.be](mailto:olivier.bonaventure@uclouvain.be)

