

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: September 9, 2010

L. Iannone
TU Berlin - Deutsche Telekom
Laboratories AG
D. Saucez
O. Bonaventure
Universite catholique de Louvain
March 8, 2010

LISP Mapping Versioning
draft-iannone-lisp-mapping-versioning-01.txt

Abstract

The present document sketches an optional approach to provide 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. This versioning approach is particularly useful to inform communicating xTRs about modification of the mappings used to encapsulate packets. Modification of mappings could mean adding/removing an RLOC, or just a modification in the reachability, priority, or weight of one or more RLOCs. Each time a mapping is modified, a new version number is generated and propagated in the LISP data packet. The use of version numbers allows to avoid repeated Map-Request upon mappings change, limits the interaction between Control and Data planes, improves security, offer support for caching on Map-Servers, and could be used also in mobile scenarios.

The proposed mechanism is optional and does not need any modification on the base LISP encapsulation. Rather, it uses one of the reserved bits of the LISP specific header and overloads the Locator Status Bits. Similarly, no modification are necessary in the base LISP Map-Reply records. LISP versioning uses part of the reserved bits. In both cases, LISP encapsulation and Map-Reply records, bits used for LISP versioning can be safely ignored by xTRs that do not support the mechanism. Further, mappings can be distributed as usual through both existing and future mapping distribution system (e.g., ALT). The infrastructure build by each specific mapping distribution system does not change anyhow. Even more, existing mapping distribution protocol are able to rely LISP control plane packets containing version numbers and do not need modifications. All of these features make LISP versioning a completely transparent optional mechanism with respect to the LISP base specification.

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/lid-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 9, 2010.

Copyright Notice

Copyright (c) 2010 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

1. Requirements notation	4
2. Introduction	5
3. EID-to-RLOC Mapping Version Number	7
3.1. Mapping Version Numbers wrap-around	7
4. Dealing with Version Numbers	9
4.1. Handling Destination Mapping Version Number	9
4.2. Handling Source Mapping Version Number	10
5. LISP header and Mapping Version Numbers	12
6. Map Record and Mapping Version Number	14
7. Benefits and case studies for Mapping Versioning	15
7.1. Mapping Versioning to simplify LISP implementation	15
7.2. Synchronization of different xTRs	15
7.3. Map Versioning and unidirectional traffic	16
7.4. Mapping Versioning and interworking	17
7.5. Mapping Versioning vs. Checksum	17
7.6. Mapping Versioning and mobility	17
8. Incremental deployment and implementation status	19
9. Mapping Versioning and path-liveness	20
10. Security Considerations	21
10.1. Mapping Versioning against traffic disruption	21
10.2. Mapping Versioning against reachability information DoS	21
11. Acknowledgements	23
12. References	24
12.1. Normative References	24
12.2. Informative References	24
Authors' Addresses	25

1. 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].

2. Introduction

The present document introduces the use of version numbers in order to provide information on a change in the EID-to-RLOC mappings used in the LISP ([I-D.ietf-lisp]) context to perform encapsulation. The mechanism is optional and totally transparent to xTRs not supporting such a functionality. The basic mechanism is to associate version numbers to each 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. The change can even consist in the reachability of one or more RLOCs. Reachability information is intended from the local-domain perspective, and can be obtained for instance monitoring IGP's routing tables. This should not be confused with the intra-domain "Locator Path Liveness problem" described in [I-D.meyer-loc-id-implications].

With this approach, LISP-encapsulated data packets contain the version number of the mappings used to select the RLOCs in the outer header (both source and destination). These version numbers are contained in the second 32-bits of the LISP header and indicated a specific bit in the reserved flags (first 8 bits of the LISP header. Details about the header are described in Section 5. Note that it is not all packets need to carry version numbers.

When an ITR encapsulates a packet, 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 that sent it is using the latest mapping for the destination EID. If it is not the case the ETR can send to the ITR a Map-Request containing the updated mapping or invoking 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 the ETR receiving the packet to know if it has in its cache the latest mapping for the source EID. If it is not the case a Map-Request can be send.

With Mapping Versioning there is no need to re-design the mapping distribution infrastructure, which is always done through the mapping

distribution protocol (e.g., CONS, TREE, ALT [I-D.ietf-lisp-alt]). The mappings are distributed as usual through the Map-Request/Map-Reply message exchange. Map-Request and Map-Reply messages can carry mapping version in bits that are reserved (in the current version of the LISP specifications). Details on how to carry mapping version numbers in those messages are given in section Section 6.

3. EID-to-RLOC Mapping Version Number

The EID-to-RLOC Mapping Version Number consist in an unsigned 16-bit 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. The initial version number of a new mapping can be randomly generated.

The space of version numbers has a circular order where half of the version numbers is greater than the current mapping version number and the other half is smaller than current mapping 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 16 bits, as proposed in this document, and if the Mapping Version Number is 0, versions in $[1; (2^{15})-1]$ are greater and versions in $[2^{15}; (2^{16})-1]$ are smaller.

3.1. Mapping Version Numbers wrap-around

The proposed 16 bits size for the Mapping Version Number based on the assumption that Map-Requests are rate limited with a granularity of seconds. Using a granularity of seconds and assuming as worst case that a new version is issued each second, it takes around 18 hours before the versions wraps around, which is a reasonable time. Alternatively a granularity of minutes can also be used, as for the TTL of the Map-Reply ([I-D.ietf-lispl]). Using a granularity of minutes leads to very long time before wrap-around. Hereafter there is a table with a rough estimation of the time before wrap-around happens considering different sizes of the Mapping Version Number and different time granularity.

Version Number Size (bits)	Time before wrap around	
	Granularity: Minutes	Granularity: Seconds
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

Figure 1: Estimation of time before wrap-around

4. Dealing with Version Numbers

The main idea of using Mapping Version Numbers is that whenever there is a change in the mapping (e.g., adding/removing RLOCs, a change in the weights due to new 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 with a new mapping version number.

In order to announce in a data-driven fashion that the mapping has been updated, mapping 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 mapping version numbers:

- o A first one from the EID-to-RLOC mapping in the EID-to-RLOC Database used to select the source RLOC, and called Source Mapping Version Number.
- o A second one from the EID-to-RLOC mapping in the EID-to-RLOC Cache used to select the destination RLOC, and called Destination Mapping Version Number.

By embedding both Source Mapping Version Number and Destination Mapping Version Number an ETR can perform the following checks:

1. The ITR has an up-to-date mapping in its cache for the destination EID and is performing encapsulation correctly.
2. The mapping in the local ETR 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 4.1) or to updated local mapping in the cache (see section Section 4.2).

4.1. Handling Destination Mapping Version Number

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

- o The packets arrive with the same Destination Mapping 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.
- o The packet arrives with a Destination Mapping 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 packets carries a not valid version number and can be silently dropped.
- o The packets arrive with an Destination Mapping 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. Further actions are needed. The ITR sending the packet must 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 piggy-back the newer mapping. This is not a new mechanism, how to piggy-back mappings in Map-Request messages is already described in [I-D.ietf-lisp]. These Map-Request message should be rate limited (rate limitation policies are also described in [I-D.ietf-lisp]). The gain introduced by Mapping Version Numbers is that after a certain number of retries, if the Destination Mapping Version Number in the packets is not updated, packet can be silently dropped because either the ITR is refusing to use the mapping for which the ETR is authoritative or it might be some form of attack. Note that the rule can be even more restrictive. If the mapping has been the same for a period of time as long as the TTL (defined in LISP [I-D.ietf-lisp]) of the previous version of the mapping, all packets arriving with an old mapping version can be silently dropped right away without issuing any Map-Request. Indeed, if the new mapping with the updated version number has been stable for at least the same time as the TTL of the older mapping, all the entries in the caches of ITRs must have expired. If packets with old mapping version number are still received, the reason is that either someone has not respected the TTL, or it is a spoof. In both cases this is not valid behavior w.r.t. the specifications and the packet can be silently dropped.

4.2. Handling Source Mapping Version Number

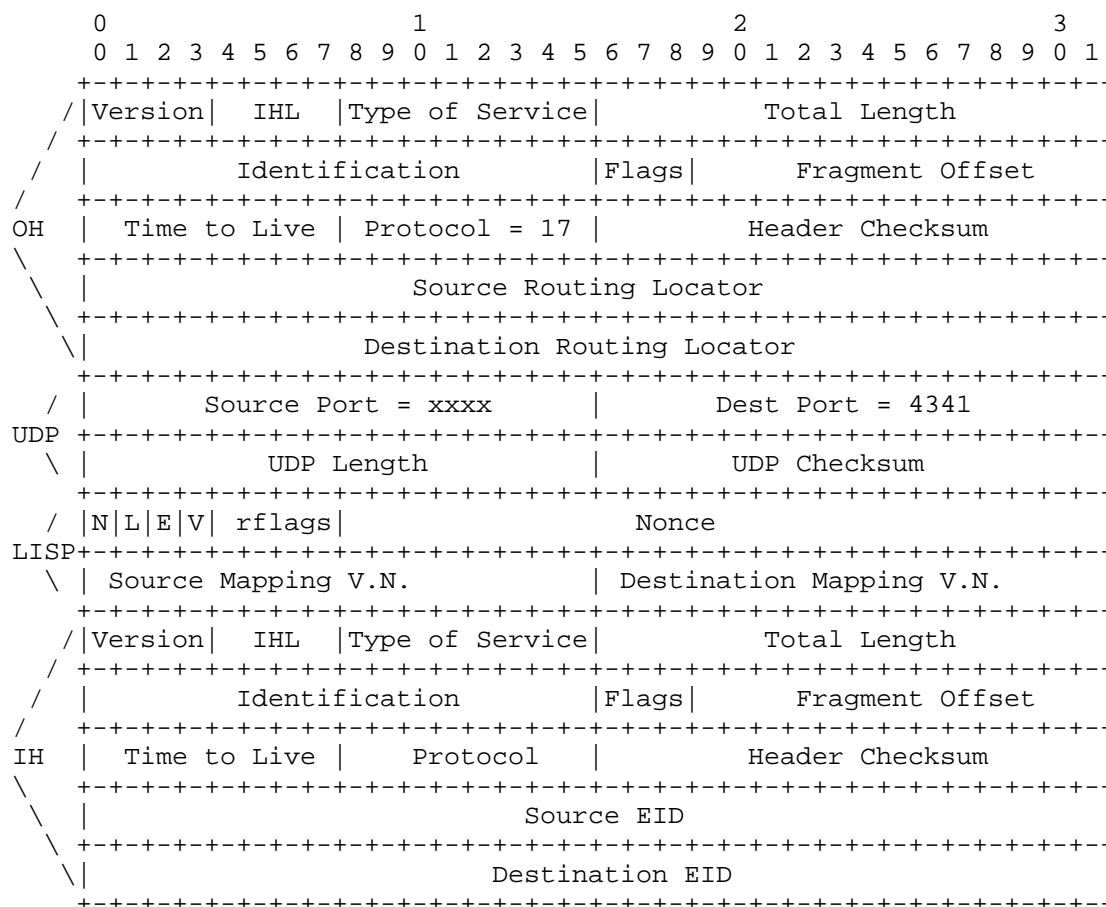
When an ETR receives a packet, the Source Mapping Version Number relates to the mapping for the source EID for which the ITR is authoritative. If the ETR has an entry in its LISP Cache a check is performed and the following cases can arise:

- o The packet arrives with the same Source Mapping Version Number stored in the LISP Cache. This is the correct regular case. The ETR has in its cache an up-to-date copy of the mapping. No further actions are needed.
- o The packet arrives with a Source Mapping Version Number greater (i.e., newer) than the one stored in the local LISP Cache. This means that ETR has in its cache a mapping that is stale and needs to be updated. The packet is considered valid but further actions are needed. In particular a Map-Request must be sent to get the new mapping for the source EID. This is a normal Map-Request message and must respect the specifications in [I-D.ietf-lisp].
- o The packet arrives with a Source Mapping Version Number smaller (i.e., older) than the one stored in the local LISP Cache. Such a case is not valid w.r.t. the specifications. Indeed, if the mapping is already present in the LISP Cache, this means that an explicit Map-Request has been send and a Map-Reply has been received from an authoritative source. Assuming that the mapping system is not corrupted anyhow, the mapping version in the LISP Cache is the correct one, hence the packet is not valid and can be silently dropped.

5. LISP header and Mapping Version Numbers

In order for the versioning approach to work, the LISP specific header has to carry both Source Mapping Version Number and Destination Mapping Version Number. This can be done by using one bit (indicated by V) of the reserved flags to state that the second 32 bits of the LISP header have to be interpreted as two version numbers of 16 bits each. The Source Version Number is carried in the 16 most significant bits of the second 32-bits of the LISP header. The Destination Version Number is carried in the 16 most significant bits of the second 32-bits of the LISP header.

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, IPv6-in-IPv6).



V: this is the Versioning bit. When this bit is set to 1 the second 32-bits of the LISP header contain version numbers.

Source Mapping Version Number (16 bits): Version of the mapping used by the ITR to select the RLOC present in the "Source Routing Locator" field. Note that the mapping used for such a selection is determined by the Source EID through a search in the LISP Database of the ITR.

Destination Mapping Version Number (16 bits): Version of the mapping used by the ITR to select the RLOC present in the "Destination Routing Locator" field. Note that the mapping used for such a selection is determined by the Destination EID, used as lookup key in the LISP Cache of the ITR.

Not all of the LISP encapsulated packets need to carry version numbers. When mapping version number are carried the V bit must be set to 1. The V bit is conflict with the L bit, since both relate to the second 32 bits of the LISP header. The possible combinations (and related meaning) for L and V bits are the following:

L=0, V=0: This is a valid combination. In this case neither Locator-Status-Bits nor Version Number are used. The second 32 bits of the LISP header can be ignored.

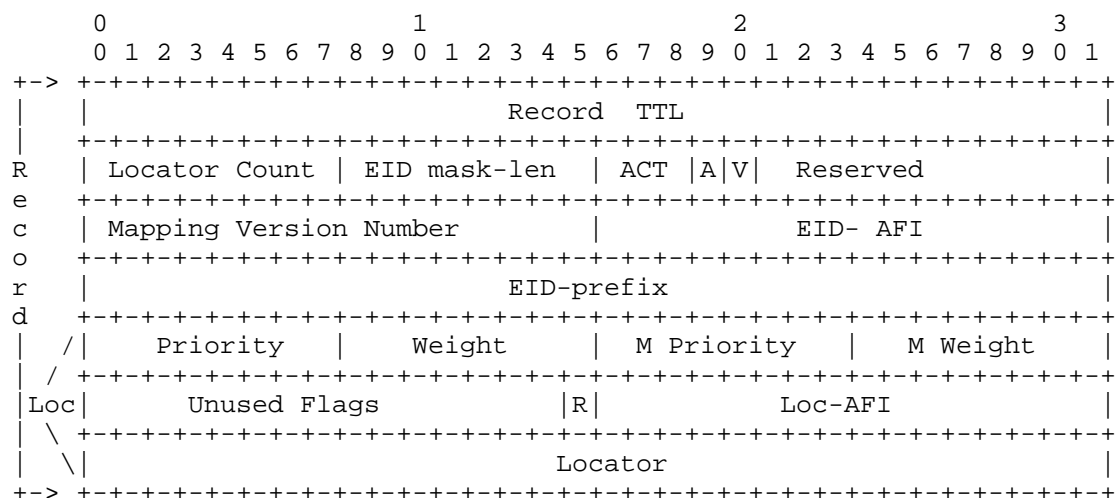
L:0 V:1 This is a valid combination. In this case the second 32 bits of the LISP header contain version numbers and should be treated according to the present document.

L:1 V:1 This is no a valid combination since two different bits indicate different content for the same 32 bits. For compatibility with the LISP specifications ([I-D.ietf-lisp]) each time the the L bit is set to 1 the V bit must be ignored and the second 32 bits of the LISP header interpreted as Locator-Status-Bits. This approach ensures transparent and coherent interoperability between xTRs using Versioning and xTRs that do not use it.

L:1 V:0 This is a valid combination. In this case the second 32 bits of the LISP header contain Locator-Status-Bit. Note that according with the previous combination, since the L bit is set to 1 the V bit can be safely ignored.

6. Map Record and Mapping Version Number

To accommodate the proposed mechanism, the Map records that are transported on Map-Request/Map-Reply messages need to carry the Mapping Version Number as well. For this purpose it is possible to use part of the reserved bits of the record. The original definition of Record is in [I-D.ietf-lisp].



Mapping Version Number: Version Number of the mapping in the Record.

This is a simple change to carry the version number assigned to the mapping in this message and works perfectly with xTR that do not support mapping 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.

7. Benefits and case studies for Mapping Versioning

In the following sections we provide more discussion on various aspects of the mapping versioning. Security observations are instead grouped in Section 10.

7.1. Mapping Versioning to simplify LISP implementation

The use of mapping versioning can help in simplifying the implementation of LISP. 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]). When using mapping versioning such type of mechanisms are not necessary anymore since there is no direct relation between the order of the locators and the bits of the mapping version number.

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 the LISP draft. A new mapping with a new version number will be issued, and since the old locators are still valid the transition will be disruptionless. 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 Loc Status Bits, for sites that are not using the RLOC that has been withdrawn, the transition will be disruptionless.

It is even possible to perform a graceful shutdown. This is obtained by simply issuing a mapping where the specific RLOC to be shut down is withdrawn or announced as unreachable (R bit in the Map Record), 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.

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

Finally, with the versioning approach there is no need to perform a "clock sweep" as described in Section 6.5.1 of the LISP draft. 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.

7.2. Synchronization of different xTRs

Mapping Versioning does not require additional synchronization mechanism compared to the normal functioning of LISP without mapping versioning. Clearly all the ETRs have to reply with the same

versioning number, otherwise there can be an inconsistency that creates additional control traffic.

As an example, let's consider the topology of figure Figure 2 where ITR A.1 of domain A is sending unidirectional traffic to the xTR B of domain B, while xTR A.2 of domain A and xTR B of domain B exchange bidirectional traffic.

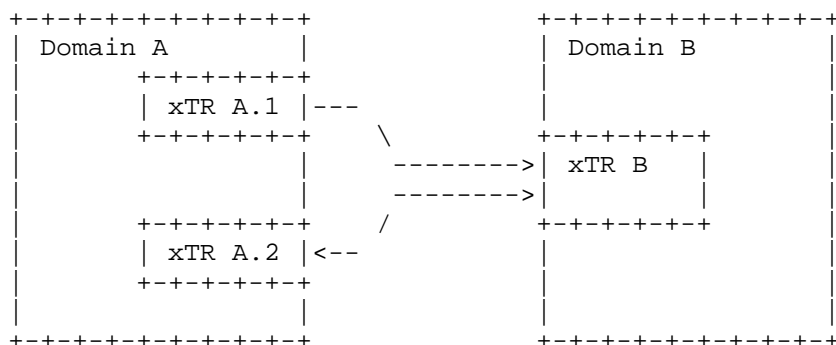


Figure 2

Obviously in the case of Mapping Versioning both xTRs of domain A must use the same value otherwise the xTR of domain B will start to send Map-Requests.

The same problem can, however, arise without mapping versioning. For instance if the two xTRs of domain A send different Loc Status Bits. In this case either the traffic is disrupted, if the xTR B trusts the Loc Status Bits, or it xTR B 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 mapping versioning. If in the future any synchronization mechanism is provided, mapping versioning will take advantage of it automatically if the record format described in Section 6 is used.

7.3. Map Versioning and unidirectional traffic

When using mapping versioning as specified in this document the LIS specific header carries two mapping version numbers, for both source and destination mapping. This can raise the question on what will happen in the case of unidirectional flows, like for instance in the case presented in Figure 3, since LISP specification do not mandate

for ETR to have a mapping for the source EID.

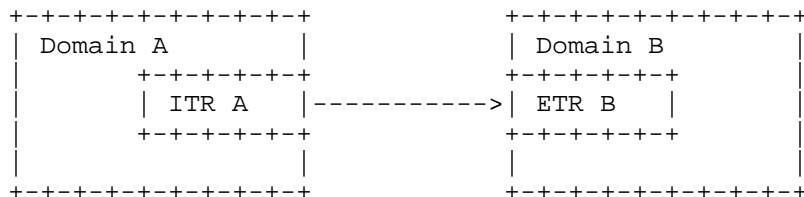


Figure 3

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

For what concerns the ETR, it simply checks only the destination version number in the same way as described in Section 4, ignoring the source mapping version number.

7.4. Mapping Versioning and interworking

Mapping versioning works also in the context of interworking between LISP and IPv4 and IPv6 ([I-D.ietf-lisp-interworking]). The case of PTR encapsulating packet for LISP sites is basically the same as the unidirectional traffic case presented in the previous section. The same rules can be applied.

7.5. Mapping Versioning vs. Checksum

Noel Chiappa has several times proposed on the LISP WG mailing list to use a form of "checksum" as a mapping version number. This is an interesting idea. Nevertheless, from our understanding, this implies that the notion of ordering between different mappings for the same EID-Prefix (e.g., whether a mapping is more recent) get lost. This means that a large part of the filtering that can be done on not valid version numbers (see Section 4) cannot be done anymore, hence losing an important feature of mapping versioning. Certainly, if it would be possible to find a "checksum" function that embeds some form of ordering, this can be discussed and integrated in future version of this document.

7.6. Mapping Versioning and mobility

Mapping versioning can help in managing mobility in the LISP context ([I-D.meyer-lisp-mn]). We are working in deploying Mapping Versioning on a Wireless Mesh Network. Results concerning this

deployment will be provided in future versions of this document.

8. Incremental deployment and implementation status

The solution proposed in this draft includes the use of bits that are marked as reserved in the main LISP specifications. This means that any LISP element that does not support Mapping Versioning will safely ignore them. Further, there is no need of any specific mechanism to discover if an xTR supports or not Mapping Versioning. This information is already included in the Map Record.

Mapping Versioning can be incrementally deployed without any negative impact on existing LISP xTRs. Mapping 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].

9. Mapping Versioning and path-liveness

When the reachability problem occurs on the path between two RLOCs of different LISP sites (this is called path-liveness problem in the recent draft by D. Meyer and D. Lewis [I-D.meyer-loc-id-implications]), the versioning approach does not help. In this case other mechanisms are necessary, however, such an issue is not new and is part of all loc/ID split solutions, thus versioning does not introduce a new issue.

10. Security Considerations

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

10.1. Mapping Versioning against traffic disruption

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

This does not happen in the case of Mapping Versioning. As described in Section 4, 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. Note also that in the case of Mapping Versioning the attacker should "guess" a valid version number that triggers a Map-Request, as described in Section 4, otherwise the packet is simply dropped.

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 Loc Status Bits loose 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 mapping versioning in order to drop packets that do not carry a valid mapping version number. In the case of Loc 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. Mapping Versioning against reachability information DoS

Attackers can try to trigger a large amount of Map-Request by simply by forging packets with random mapping version or random Loc Status Bits. In both cases the Map-Requests are rate limited as described in [I-D.ietf-lisp]. However, differently from Loc Status Bit where there is no filtering possible, in the case of mapping 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 mapping 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 mapping 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 mapping versioning and is the same for Loc Status Bits.

11. Acknowledgements

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

12. References

12.1. Normative References

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

12.2. 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]
Farinacci, D., Fuller, V., Meyer, D., and D. Lewis,
"Locator/ID Separation Protocol (LISP)",
draft-ietf-lisp-06 (work in progress), January 2010.
- [I-D.ietf-lisp-alt]
Fuller, V., Farinacci, D., Meyer, D., and D. Lewis, "LISP Alternative Topology (LISP+ALT)", draft-ietf-lisp-alt-02
(work in progress), January 2010.
- [I-D.ietf-lisp-interworking]
Lewis, D., Meyer, D., Farinacci, D., and V. Fuller,
"Interworking LISP with IPv4 and IPv6",
draft-ietf-lisp-interworking-00 (work in progress),
May 2009.
- [I-D.ietf-lisp-ms]
Fuller, V. and D. Farinacci, "LISP Map Server",
draft-ietf-lisp-ms-04 (work in progress), October 2009.
- [I-D.meyer-lisp-mn]
Farinacci, D., Fuller, V., Lewis, D., and D. Meyer, "LISP Mobile Node", draft-meyer-lisp-mn-01 (work in progress),
February 2010.
- [I-D.meyer-loc-id-implications]
Meyer, D. and D. Lewis, "Architectural Implications of Locator/ID Separation", draft-meyer-loc-id-implications-01
(work in progress), January 2009.

Authors' Addresses

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

Email: 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

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

Email: olivier.bonaventure@uclouvain.be

