Network Working Group
Internet-Draft
Updates: 6830 (if approved)
Intended status: Standards Track
Expires: August 7, 2017

M. Boucadair
C. Jacquenet
Orange

February 3, 2017

LISP Mapping Bulk Retrieval
draft-boucadair-lisp-bulk-04

Abstract

This document extends Locator/ID Separation Protocol (LISP) with a capability for bulk mapping retrieval. It does so by defining new LISP messages that are meant to facilitate state recovery of mapping tables and improve Ingress Tunnel Routers (ITR) recovery times, in particular. In addition, this document allows to request mappings that match destination IP prefixes, names, or AS numbers.

This document updates RFC6830.

Requirements Language

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 RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on August 7, 2017.

Table of Contents

1. Introduction

Locator/ID Separation Protocol (LISP, [RFC6830]) operation relies upon a mapping mechanism that is used by ingress/egress Tunnel Routers (xTR) to forward traffic over the LISP network. This document extends LISP with a capability for bulk mappings retrieval. It does so by defining new LISP messages that are meant to facilitate state recovery of mapping tables and improve Ingress Tunnel Routers (ITR) recovery times, in particular.

The base LISP specification does not define how a requestor may ask for multiple EIDs. Indeed, the current LISP specification [RFC6830] states the following:

Support for requesting multiple EIDs in a single Map-Request message will be specified in a future version of the protocol.

The extensions defined by this document allow for faster recovery of mapping entries. For example, whenever an ITR fails for some reason, the faulty ITR needs to recover at least the list of mappings for the most popular prefixes in a timely manner, etc. These extensions may be used by a leaf LISP network or enabled between mapping systems for the sake of global mapping table maintenance. Policies for the mapping entries to be recovered are deployment-specific.

The document defines a backward compatible extension of the LISP Map-Request message to request multiple records (Section 2). Also, it defines a more reliable method for the retrieval of mapping records from one or multiple Map-Resolvers (Section 3). It does so by using TCP ([RFC0793]) as a transport protocol for exchanges the bulk retrieval messages. Other proposals have been made to use TCP for reliable transport (e.g., [I-D.kouvelas-lisp-map-server-reliable-transport])

This document allows to request mappings that match destination IP prefixes, names, or AS numbers. Other filter types may be defined in future versions, if needed.

2. Map-Request with Multiple Records

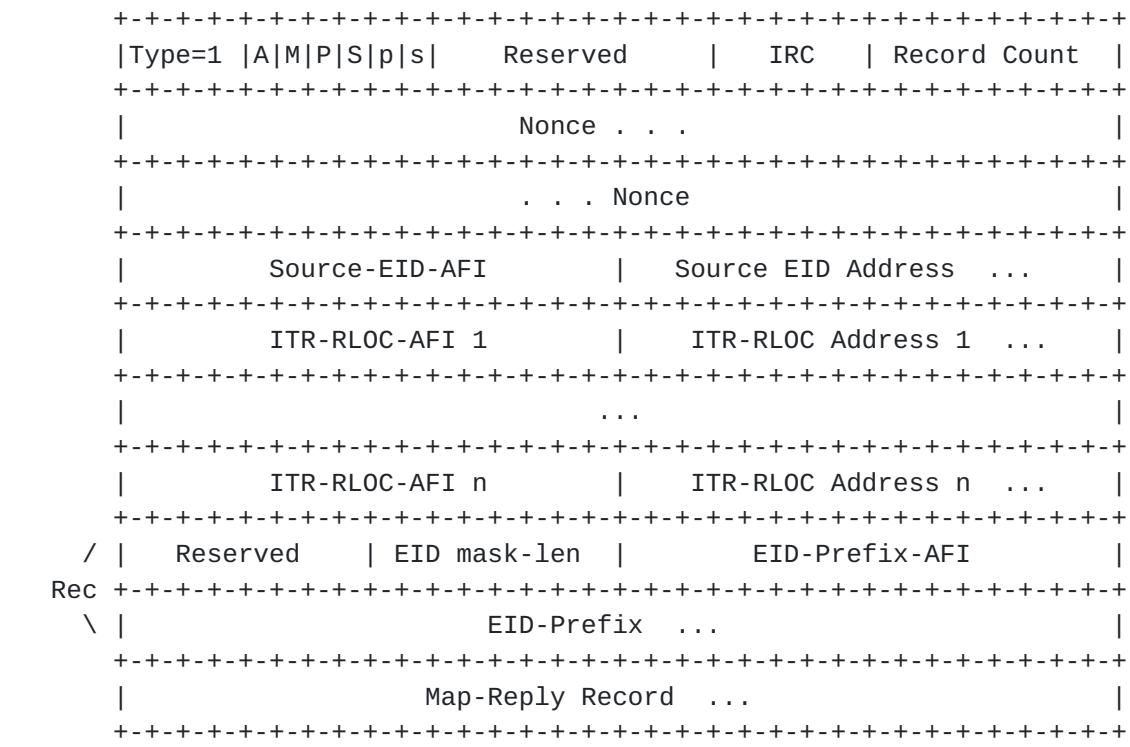
As mentioned in Section 1, [RFC6830] does not specify how an ITR can request for multiple EIDs using the same Map-Request message. This document fills that void.

Figure 1 shows the difference between the current Map-Request message format and the new format that includes the proposed extension. This extension is meant to allow an ITR to request multiple EID records by using the same Map-Request.

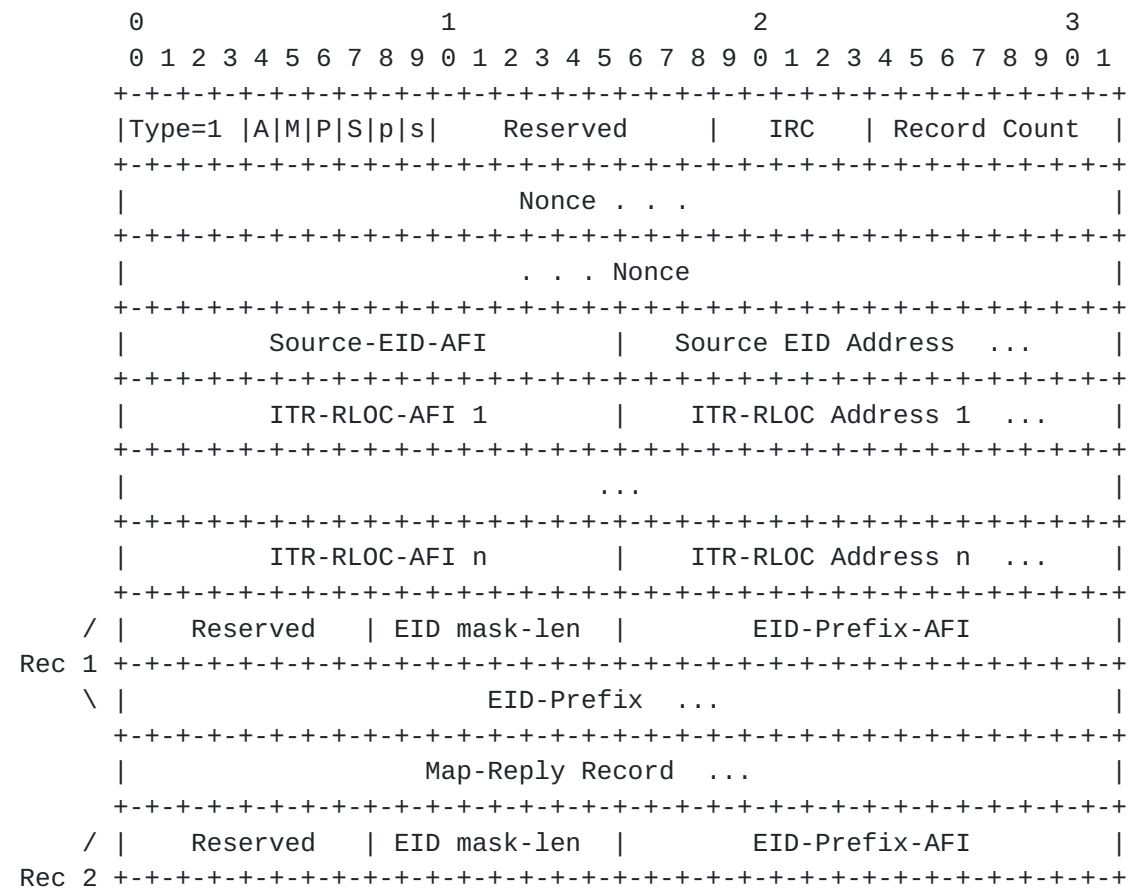
The proposed design is backward compatible since it aligns the additional requested EID records at the end of the Map-Request message.

As specified in [RFC6830], a mapping system must be prepared to receive a request for multiple EID records in a Map-Request message. A receiver relies upon the content of the "Record Count" field of the Map-Request message to detect whether one or multiple records are carried in the request.

OLD:


|Type=1 |A|M|P|S|p|s| Reserved | IRC | Record Count |
Nonce . . .
. . . Nonce
Source-EID-AFI | Source EID Address ...
ITR-RLOC-AFI 1 | ITR-RLOC Address 1 ...
...
ITR-RLOC-AFI n | ITR-RLOC Address n ...
Rec
Reserved | EID mask-len | EID-Prefix-AFI
EID-Prefix ...
Map-Reply Record ...

NEW:


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
|Type=1 |A|M|P|S|p|s| Reserved | IRC | Record Count |
Nonce . . .
. . . Nonce
Source-EID-AFI | Source EID Address ...
ITR-RLOC-AFI 1 | ITR-RLOC Address 1 ...
...
ITR-RLOC-AFI n | ITR-RLOC Address n ...
Rec 1
Reserved | EID mask-len | EID-Prefix-AFI
EID-Prefix ...
Map-Reply Record ...
Rec 2
Reserved | EID mask-len | EID-Prefix-AFI

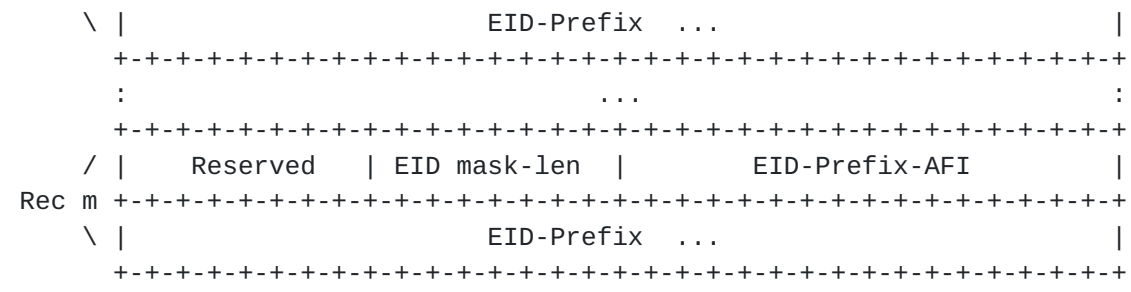


Figure 1

The description of the fields of the updated Map-Request message is exactly the same as in [RFC6830], except the additional records that are prepended after the "Map-Reply Record" . The structure of a record is exactly the same as in [RFC6830].

When extracting the records included in a Map-Request message, a Map-Resolver replies with the list of mappings that match these records. One or multiple Map-Reply messages may be required to carry the mapping records that match the requested EIDs included in a Map-Request.

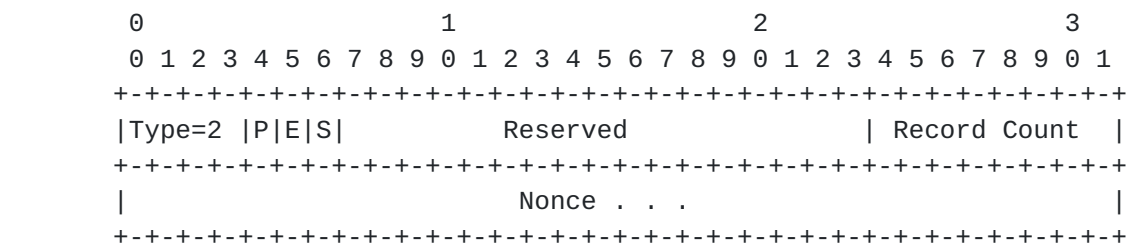
An ITR MUST be prepared to receive multiple Map-Reply messages from a Map-Resolver as a response to a bulk Map-Request message that it originally sent to that Map-Resolver.

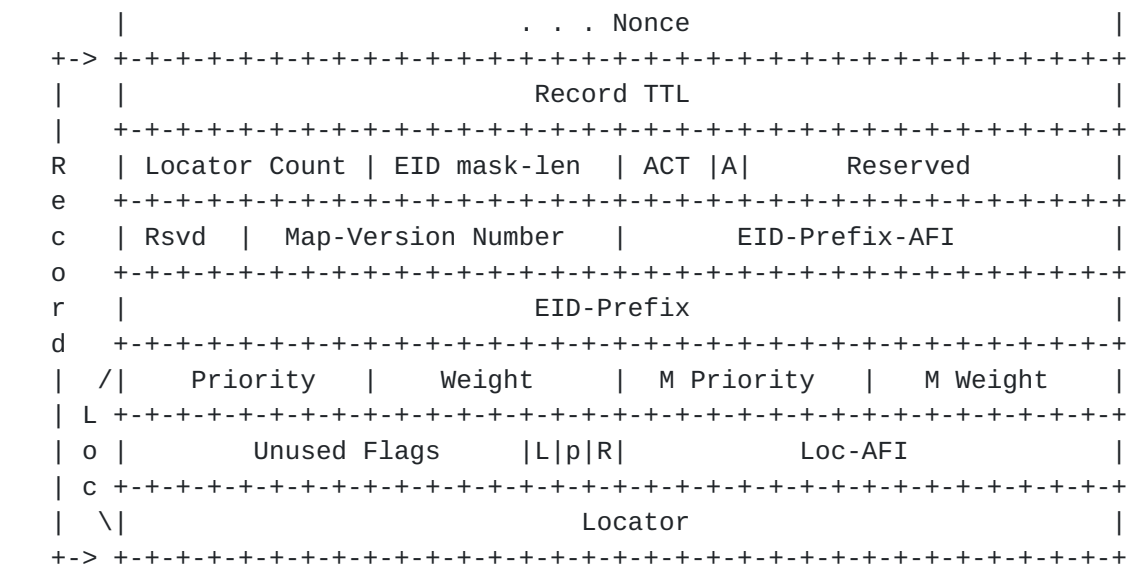
In order to inform an ITR that subsequent Map-Reply messages will follow (or not) , a dedicated flag bit is defined for this purpose: it is called the M-bit (more-map-reply bit).

When set, the M-bit (more-map-reply bit) flag indicates this is not the last Map-Reply message to be received by the requesting ITR; additional Map-Reply messages follow. An implementation uses this bit to decide when to terminate a request/response transaction.

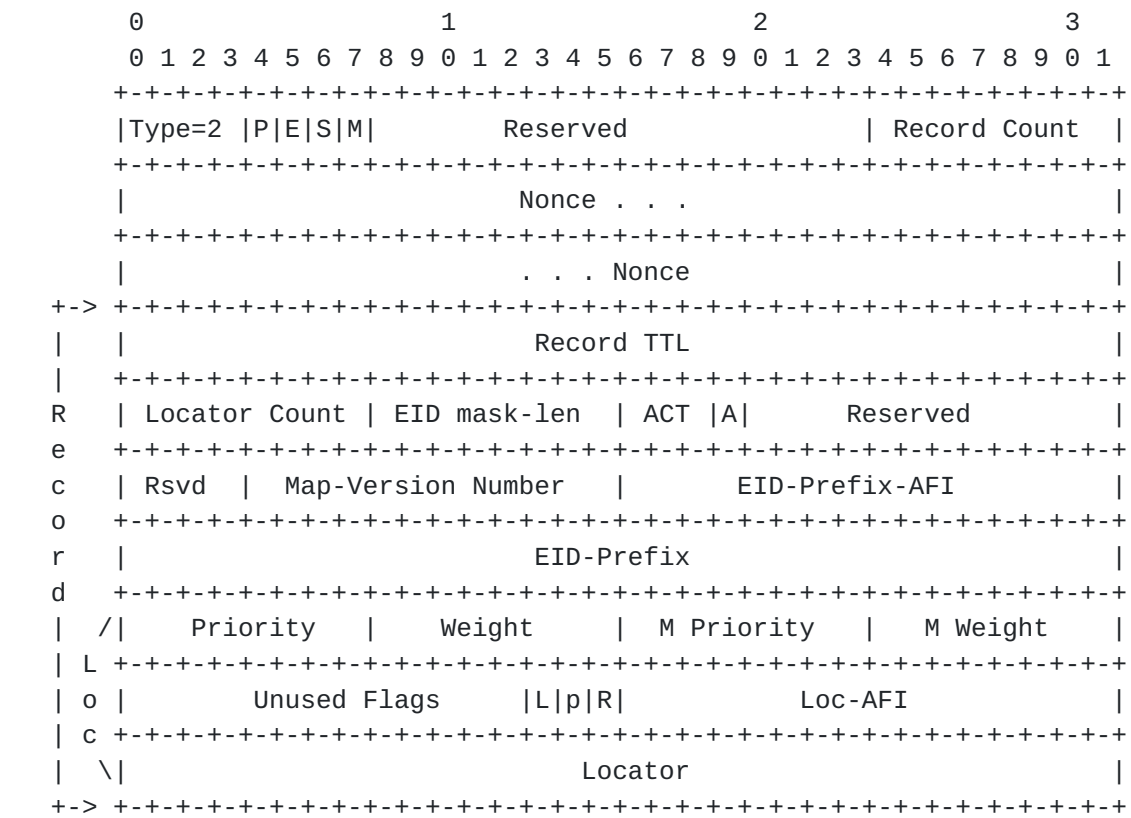
If multiple Map-Reply messages are required to respond to a Map-Request message, a Map-Resolver MUST set the M-bit flag for all Map-Reply messages except for the last Map-Reply message.

OLD:





NEW:



In order to prevent reordering issues that would lead to drop incoming Map-Reply messages, a more reliable solution is defined in Section 3.

3. Bulk Mapping Retrieval

To allow for a more reliable method when retrieving multiple EID mapping records from one or multiple Map-Resolvers, this section defines additional LISP messages that are, unlike LISP control messages, transported over TCP.

After establishing a TCP connection towards a Map-Resolver (using the LISP service port), the ITR sends a Map-Bulk-Request (Section 3.1). Upon receipt of that message, the Map-Resolver must reply with one or more Map-Bulk-Reply messages (Section 3.2). Once the last Map-Bulk-Reply is received from the Map-Resolver, the underlying TCP connection may be closed.

Figure 2 illustrates the example of a bulk mapping retrieval that is achieved with one single Map-Bulk-Reply, while Figure 3 shows an example of a bulk mapping retrieval that requires multiple Map-Bulk-Reply messages.

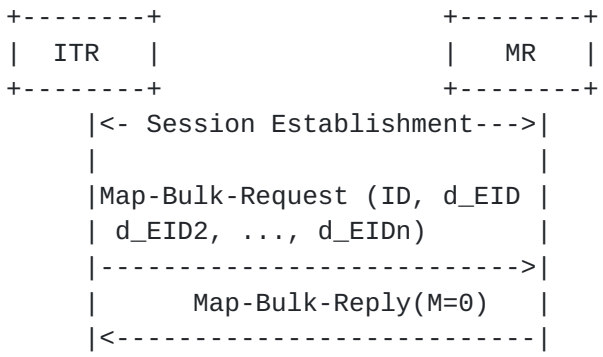


Figure 2: Example of Bulk Mapping Retrieval

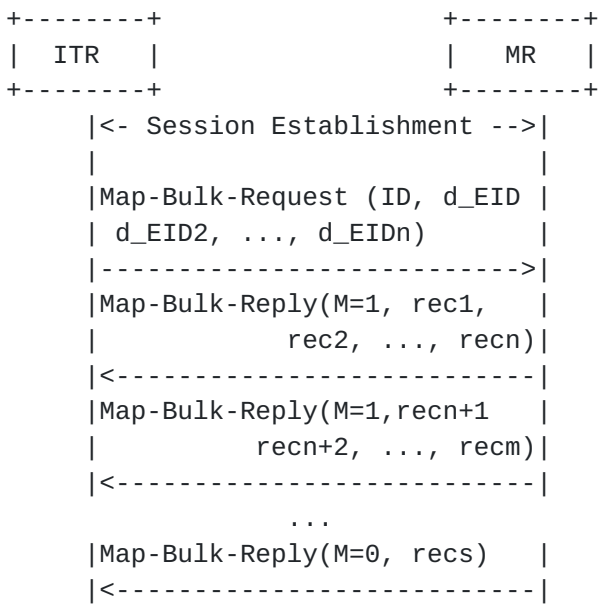


Figure 3: Example of Bulk Mapping Retrieval

The bulk mapping retrieval allows to retrieve records that do not only match IP prefixes, but also AS numbers or even names. When names or AS numbers are included, the Map-Resolver is responsible for identifying which IP prefixes are to be returned.

An ITR can establish multiple transactions with the same Map-Resolver as shown in Figure 4.

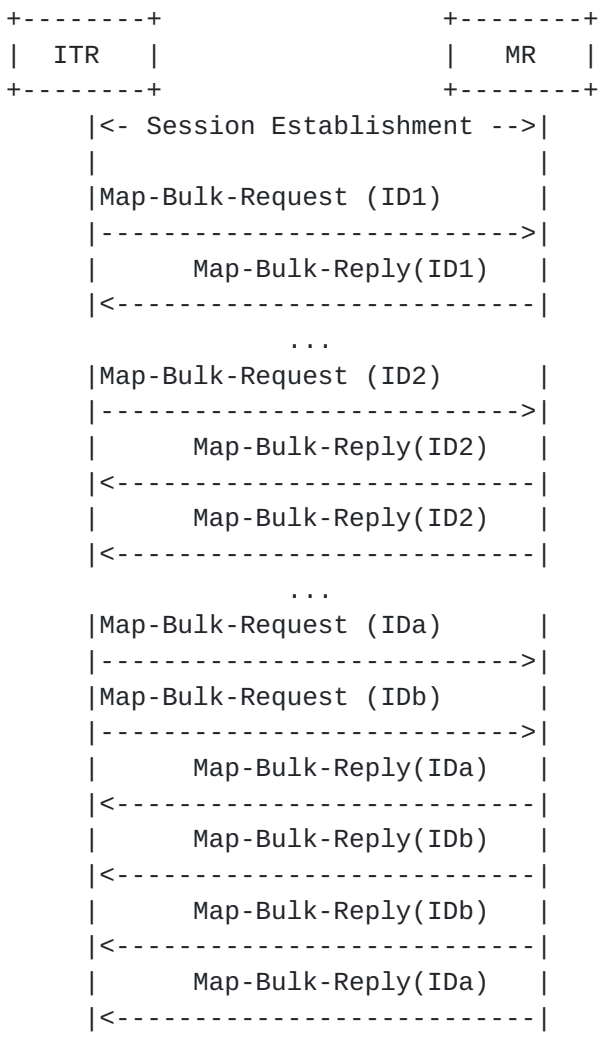


Figure 4: Multiple Transactions with the Same Map-Resolver

3.1. Map-Bulk-Request Message Format

The format of the Map-Bulk-Request message is shown in Figure 5.

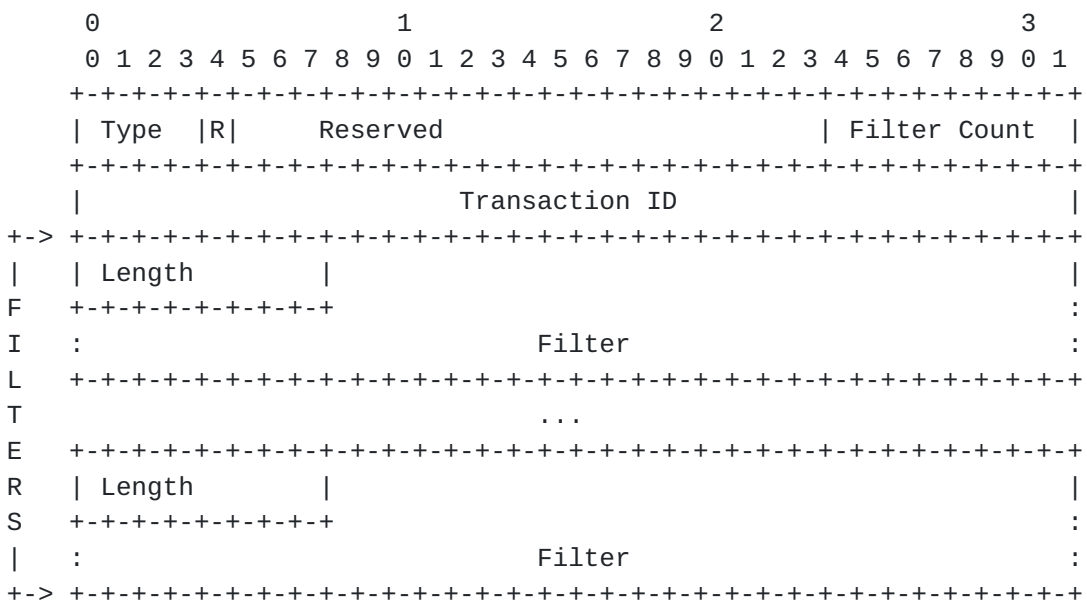


Figure 5: Map-Bulk-Request Message Format

The description of the fields is as follows:

- Type is to be defined (see Section 5).

- R bit: MUST be set to 0 for Map-Bulk-Request messages.

- Reserved: reserved bits, MUST be sent as zeros and MUST be ignored when received.

- Filter Count: This field indicates the number of filters included in the request.

- Transaction ID: This field is used to uniquely identify a connection context among those established with the same Map-Resolver. Demux connections established with distinct Map-Resolvers may rely on the address of the Map-Resolver. A transaction-id MUST be unique for connections bound to the same Map-Resolver.

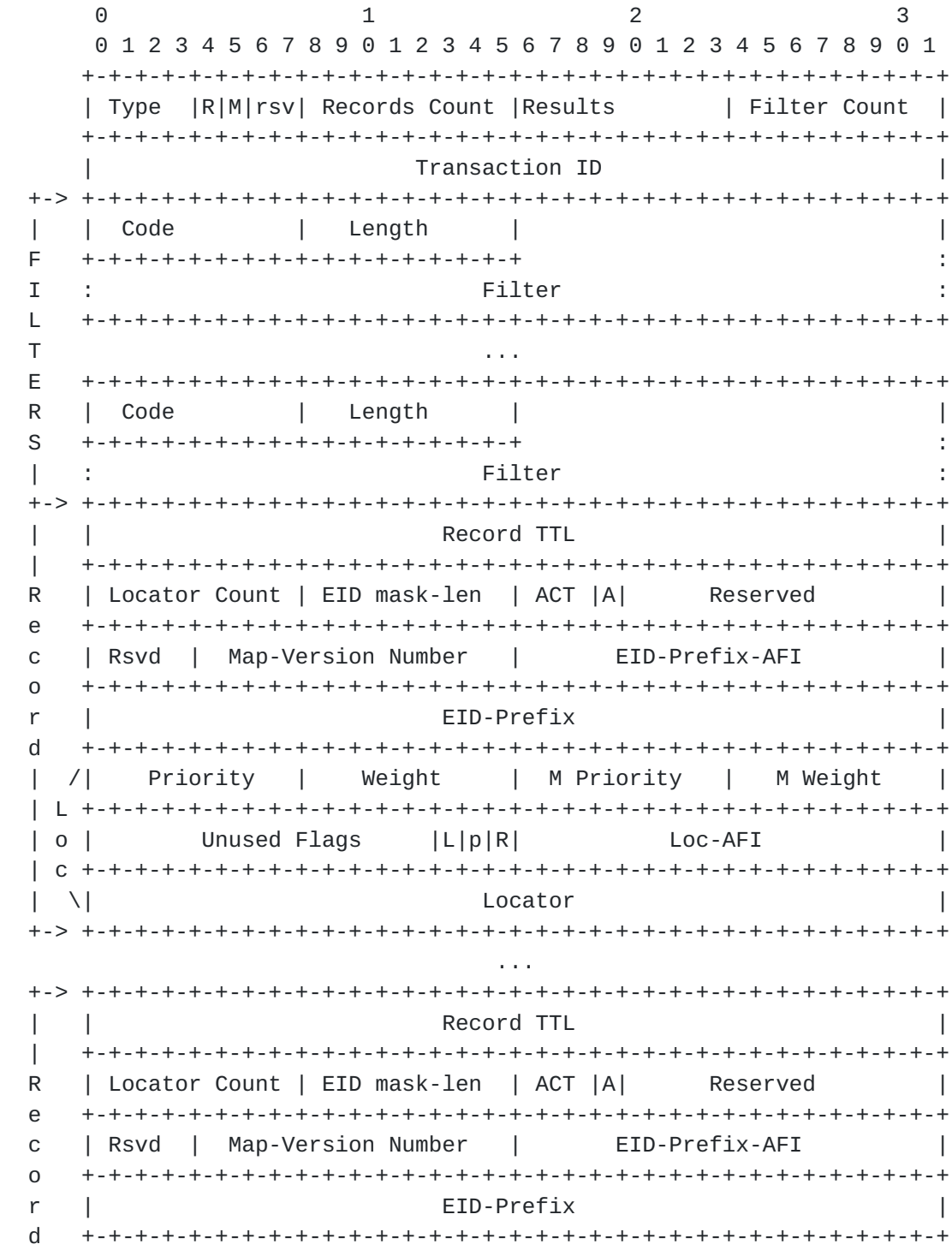
- Length: This field indicates, in octets, the length of the filter that is encoded in the "Filter" field.

- Filter: This field carries a destination EID (or a set thereof) that is encoded as an UTF-8 string. This specification allows to convey IP prefix literals, Names and/or AS numbers. One or multiple filters may be present in a request. IPv4 prefixes are encoded as IPv4-mapped IPv6 prefixes [RFC4291] (i.e., starting with ::ffff:0:0/96). A mix of names, IP prefixes and AS numbers

may be enclosed in the same request. The value 0 is used to indicate "ANY" mapping.

3.2. Map-Bulk-Response Message Format

The format of the Map-Bulk-Reply message is shown in Figure 6.



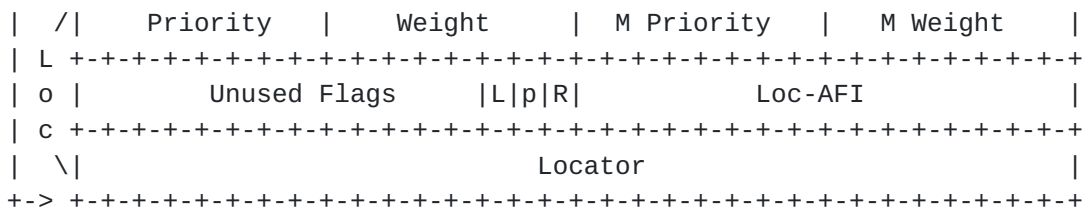


Figure 6: Map-Bulk-Reply Message Format

The description of the fields of the Map-Bulk-Reply is similar to those of a LISP Map-Request message ([RFC6830]), except the following:

- Type is to be defined. The same code is used for both Map-Bulk-Request and Map-Bulk-Reply.

- R bit: MUST be set to 1 for Map-Bulk-Reply messages.

- M (more-data bit): When set, this flag indicates that other records are to be expected from the Map-Resolver.

- rsv: reserved bits, MUST be sent as zeros and MUST be ignored when received.

- Records Count: Indicates the number of records included in the response.

- Result: indicates the result code of the processing of the Map-Bulk-Request message. The following codes are defined:

  0: SUCCESS. This code indicates the request is successfully processed.

  1: BULK-PROHIBITED. This code indicates the bulk mapping is blocked for this ITR, leaf LISP network, subscriber, etc.

  2: BULK-LIMIT. This code indicates a rate-limit is applied on the Map-Bulk-Request messages from the same ITR, leaf LISP network, subscriber, etc. The ITR SHOULD re-issue the request after the expiry of a timer; the default value of that timer is 60 seconds. Other values may be configured on the ITR.

  3: OUT-OF-RESOURCES. This code indicates a Map-Resolver is running out of resources. The ITR SHOULD re-iterate the same request after the expiry of a timer. The default value of that timer is 300 seconds. Other values MAY be configured on the ITR.

- Filter Count: This field indicates the number of filters that were not processed by the Map-Resolver. A filter MUST be included in a response if and only if an error was encountered when processing that filter at the Map-Resolver side. The "Result" code provides more details about the reason for not processing such filter. If all filters were successfully processed by the Map-Resolver, this field MUST be set to 0.

- Transaction ID: MUST echo the one included in the Map-Bulk-Request.

- Code: Indicates the reason why the filter is not included

  0: FILTER-UNSUPPORTED. This code indicates a request is successfully processed but this filter was not processed because the format of the filter is not supported.

  1: FILTER-BAD. This code indicates a request is successfully processed but the filter was not processed because it is malformed.

  2: FILTER-MAX. This code indicates a request is successfully processed but the filter was not processed because of a limit enforced on the maximum number of filters to be processed.

  3: FILTER-LOCAL. This code indicates a request is successfully processed but the filter was not processed because of local reasons. The ITR SHOULD, after a certain timer expires, send a Map-Bulk-Request message for the set of filters that are not processed with a reason code set to BULK-LOCAL.

- Length: Indicates the length of a filter this is not processed by the Map-Resolver.

- Filter: Conveys a filter that is not processed by the Map-Resolver.

3.3. Generating a Map-Bulk-Request Message

ITRs MUST support a configurable parameter to enable/disable bulk mapping retrieval over TCP. The default value is set to "enabled".

If distinct port number is used by remote Map-Resolvers, the destination port number to send Map-Bulk-Request messages SHOULD be configured to the ITR.

After establishing a TCP connection towards a Map-Resolver (using the LISP service port), the ITR MUST send a Map-Bulk-Request

(Section 3.1) to a Map-Resolver. Configuration information for triggering bulk retrieval request messages MAY be provisioned to each ITR. Multiple Map-Bulk-Request messages may be sent over the same TCP connection.

An ITR that loses its mapping cache for some reason SHOULD generate a Map-Bulk-Request message towards its Map-Resolver(s) with the set of filters that are configured locally.

An ITR MAY generate several Map-Bulk-Request messages to the same or distinct Map-Resolvers.

An ITR MUST generate a unique transaction-id per Map-Bulk-Request it issues.

An ITR MUST expect that the Map-Resolver may limit the number of filters that may be processed. Filters that are not accepted or not processed by the Map-Resolvers are included in a Map-Bulk-Reply.

3.4. Processing a Map-Bulk-Request Message

A Map-Resolver that does not support the Map-Bulk-Request message MUST silently ignore any Map-Bulk-Request message it receives.

Map-Resolvers MUST support a configurable parameter to enable/disable the processing of Map-Bulk-Request messages. The default value is set to "enabled".

A Map-Resolver that is enabled to process Map-Bulk-Request messages MUST listen to incoming TCP connections on the default LISP service port. ACLs MAY be configured to control the leaf networks that can invoke this feature.

A Map-Resolver SHOULD support a configuration parameter to rate-limit the number of simultaneous Map-Bulk-Request messages per leaf LISP network, per ITR, etc.

If a Map-Resolver receives a Map-Bulk-Request message and it is enabled to process it, a Map-Resolver MUST reply with one or multiple Map-Bulk-Reply messages.

If multiple Map-Bulk-Reply messages are required to respond to a given request, the Map-Resolver MUST:

- Echo the transaction-id.
- Set to R-bit.

- Set the M-bit for all Map-Bulk-Reply messages, except for the last one.

- Include the set of filters that are not successfully processed for some reason (e.g., malformed filter) and set the "Filter Count" accordingly.

If filters are included in the request, the Map-Resolver MUST extract those filters and lookup its mapping system accordingly. In particular, the Map-Resolver MUST reply with a full mapping table if a Null filter is included in the Map-Bulk-Request.

If bulk mapping retrieval is not allowed for a given ITR, the 'Result' field MUST be set to BULK-PROHIBITED.

If the Map-Resolver fails to process a request because limits for that ITR are exceeded, it MUST set the 'Result' field to BULK-LIMIT.

If a subset of filters are successfully processed, the 'Result' field MUST be set to SUCCESS. The set of filters that are not processed MUST be echoed in the Map-Bulk-Reply; each with a failure code that reflects the reason why the filter was not applied. If a filter type is not supported by the Map-Resolver, the 'Code' field MUST be set to FILTER-UNSUPPORTED. If the Map-Resolver fails to process some of the filters included in a request because these filters were malformed, it MUST echo the corresponding filters in the Map-Bulk-Reply message and MUST set the 'Code' field to FILTER-BAD. f the Map-Resolver fails to process some of the filters included in a request because a maximum number of filters is supported, it MUST echo the corresponding filters in the Map-Bulk-Reply message and set the 'Code' field to FILTER-MAX. If, for some other reasons, the Map-Resolver fails to apply some filters included in a request, it MUST echo the corresponding filter in the Map-Bulk-Reply message. The 'Code' field MUST be set to FILTER-LOCAL.

A Map-Resolver that is overloaded MUST reply with a Map-Bulk-Reply message with the "Result" code set to OUT-OF-RESOURCES.

3.5. Processing a Map-Bulk-Reply Message

Upon receipt of a Map-Bulk-Reply message, the ITR MUST check whether the message matches a Map-Bulk-Request it issued for the same Map-Resolver. If no matching state is found, the message MUST be silently dropped. If a state is found, the ITR MUST proceed as follows:

- An ITR that receives the result code set to BULK-PROHIBITED MUST NOT reissue a Map-Bulk-Request message to that Map-Resolver.

- An ITR that receives the result code set to BULK-LIMIT MUST NOT try to resend the same request before the expiry of the retransmission timeout (default value set to 60 seconds).

- An ITR that receives the result code set to OUT-OF-RESOURCES MUST NOT resend the same request before 300 seconds.

- If the M-bit is set, it should expect that other Map-Bulk-Reply messages will be received from this Map-Resolver. Appropriate security mechanisms (e.g., Access Control Lists) SHOULD be activated to allow the processing of these incoming unsolicited Map-Bulk-Reply messages.

- If the M-bit is unset, this is an indication that this message terminates the mapping bulk retrieval transaction. The ITR may decide to terminate the underlying TCP connections if no other transactions bound to the same Map-Resolver are active.

- Filters that are returned in the Map-Bulk-Reply message may not be valid or have exceeded a limit. The "Code" field indicates the reason for not processing these filters. In particular:

  * An ITR that receives the 'Code' field set to FILTER-BAD or FILTER-UNSUPPORTED MUST NOT resend the same filters that were returned in the Map-Bulk-Reply message, in subsequent Map-Bulk-Request messages. Furthermore, subsequent Map-Bulk-Request messages MUST NOT use the unsupported format to encode the filters.

  * An ITR that receives the 'Code' field set to FILTER-MAX SHOULD issue another Map-Bulk-Request message with the filters that were returned in the Map-Bulk-Reply message with this code.

  * An ITR that receives the 'Code' field set to FILTER-LOCAL SHOULD for at least 60 seconds before issuing another Map-Bulk-Request message with the filters that were returned in the Map-Bulk-Reply message with this code.

3.6. Bulk Mapping Retrival from Multiple Resolvers

In order to retrieve mapping entries from multiple Map-Resolvers, an ITR issues Map-Bulk-Request messages to a list of Map-Resolvers. Each of these requests is handled as specified in Section 3.3.

An ITR MAY be configured to issue multiple Map-Bulk-Request messages to distinct Map-Resolvers.

Conflicts may arise when contacting multiple Map-Resolvers. These conflicts are not specific to the bulk mapping retrieval as this is also an issue for individual mapping lookup.

4. Security Considerations

In addition to the security considerations discussed in [RFC6830] and [RFC6833], TCP-specific threats are valid for this specification (e.g., [I-D.ietf-tcpm-tcp-security]).

In order to avoid exhausting the resources of Map-Resolvers, Map-Bulk-Request messages SHOULD be rate-limited. Furthermore, a Map-Resolver MAY configure ACLs to control leaf LISP networks that are allowed to issue Map-Bulk-Request messages.

The structure of a record conveyed in a Map-Bulk-Reply is exactly the same as in [RFC6830]. As such, this specification does leak information that would not be revealed using the base LISP.

5. IANA Considerations

This document requests IANA to assign a new code from the LISP Packet Types registry ([I-D.ietf-lisp-type-iana]):

Message	Code	Reference
Map-Bulk-Request/Map-Bulk-Reply	TBD	[This document]

6. Acknowledgments

This work is partly funded by ANR LISP-Lab project #ANR-13-INFR-009-X.

Many thanks to S. Secci and Chi Dung Phung for the comments.

7. References

7.1. Normative references

[I-D.ietf-lisp-type-iana]
Boucadair, M. and C. Jacquenet, "LISP Shared Extension Message & IANA Registry for LISP Packet Type Allocations", draft-ietf-lisp-type-iana-06 (work in progress), February 2017.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <http://www.rfc-editor.org/info/rfc793>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <http://www.rfc-editor.org/info/rfc4291>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <http://www.rfc-editor.org/info/rfc6830>.

[RFC6833] Fuller, V. and D. Farinacci, "Locator/ID Separation Protocol (LISP) Map-Server Interface", RFC 6833, DOI 10.17487/RFC6833, January 2013, <http://www.rfc-editor.org/info/rfc6833>.

7.2. Informative references

[I-D.ietf-tcpm-tcp-security]
Gont, F., "Survey of Security Hardening Methods for Transmission Control Protocol (TCP) Implementations", draft-ietf-tcpm-tcp-security-03 (work in progress), March 2012.

[I-D.kouvelas-lisp-map-server-reliable-transport]
Cassar, C., Kouvelas, I., Lewis, D., Arango, J., and J. Leong, "LISP Map Server Reliable Transport", draft-kouvelas-lisp-map-server-reliable-transport-02 (work in progress), August 2016.

Authors' Addresses

Mohamed Boucadair
Orange
Rennes 35000
France

EMail: mohamed.boucadair@orange.com

Christian Jacquenet
Orange
Rennes 35000
France

EMail: christian.jacquenet@orange.com