Network Working Group
Internet-Draft
Intended status: Experimental
Expires: April 19, 2015

D. Farinacci
lispers.net
D. Meyer
Brocade
J. Snijders
Hibernia Networks
October 16, 2014

LISP Canonical Address Format (LCAF)
draft-ietf-lisp-lcaf-06

Abstract

This draft defines a canonical address format encoding used in LISP control messages and in the encoding of lookup keys for the LISP Mapping Database System.

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 April 19, 2015.

Copyright Notice

Copyright (c) 2014 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 Simplified BSD License.

Table of Contents

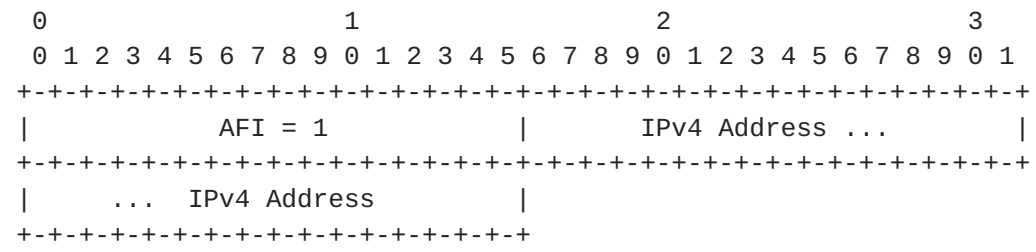
1. Introduction 3
2. Definition of Terms 4
3. LISP Canonical Address Format Encodings 4
4. LISP Canonical Address Applications 6
4.1. Segmentation using LISP 6
4.2. Carrying AS Numbers in the Mapping Database 8
4.3. Convey Application Specific Data 9
4.4. Assigning Geo Coordinates to Locator Addresses 10
4.5. Generic Database Mapping Lookups 11
4.6. NAT Traversal Scenarios 13
4.7. PETR Admission Control Functionality 15
4.8. Multicast Group Membership Information 16
4.9. Traffic Engineering using Re-encapsulating Tunnels . . . 18
4.10. Storing Security Data in the Mapping Database 19
4.11. Source/Destination 2-Tuple Lookups 20
4.12. Replication List Entries for Multicast Forwarding 21
4.13. Data Model Encoding 22
4.14. Encoding Key/Value Address Pairs 23
4.15. Applications for AFI List Type 23
4.15.1. Binding IPv4 and IPv6 Addresses 23
4.15.2. Layer-2 VPNs 25
4.15.3. ASCII Names in the Mapping Database 25
4.15.4. Using Recursive LISP Canonical Address Encodings . . 26
4.15.5. Compatibility Mode Use Case 27
5. Security Considerations 28
6. IANA Considerations 28
7. References 28
7.1. Normative References 28
7.2. Informative References 28
Appendix A. Acknowledgments 29
Appendix B. Document Change Log 30
B.1. Changes to draft-ietf-lisp-lcaf-06.txt 30
B.2. Changes to draft-ietf-lisp-lcaf-05.txt 30
B.3. Changes to draft-ietf-lisp-lcaf-04.txt 30
B.4. Changes to draft-ietf-lisp-lcaf-03.txt 30
B.5. Changes to draft-ietf-lisp-lcaf-02.txt 31
B.6. Changes to draft-ietf-lisp-lcaf-01.txt 31
B.7. Changes to draft-ietf-lisp-lcaf-00.txt 31
Authors' Addresses 31

1. Introduction

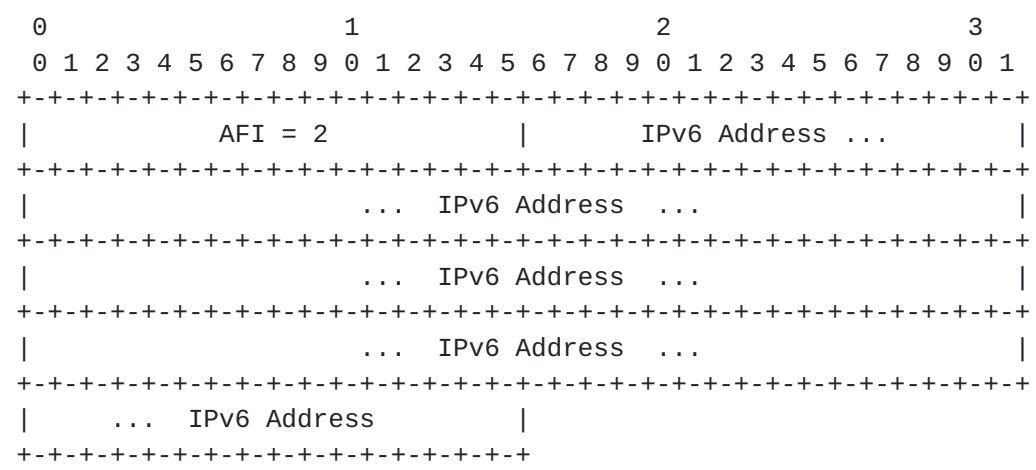
The LISP architecture and protocols [RFC6830] introduces two new numbering spaces, Endpoint Identifiers (EIDs) and Routing Locators (RLOCs) which are intended to replace most use of IP addresses on the Internet. To provide flexibility for current and future applications, these values can be encoded in LISP control messages using a general syntax that includes Address Family Identifier (AFI), length, and value fields.

Currently defined AFIs include IPv4 and IPv6 addresses, which are formatted according to code-points assigned in [AFI] as follows:

IPv4 Encoded Address:



IPv6 Encoded Address:

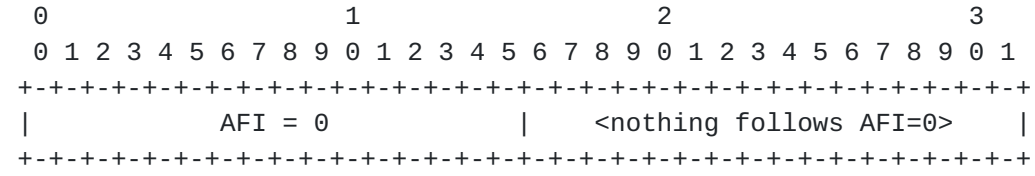


This document describes the currently-defined AFIs the LISP protocol uses along with their encodings and introduces the LISP Canonical Address Format (LCAF) that can be used to define the LISP-specific encodings for arbitrary AFI values.

2. Definition of Terms

Address Family Identifier (AFI): a term used to describe an address encoding in a packet. An address family currently defined for IPv4 or IPv6 addresses. See [AFI] and [RFC1700] for details. The reserved AFI value of 0 is used in this specification to indicate an unspecified encoded address where the the length of the address is 0 bytes following the 16-bit AFI value of 0.

Unspecified Address Format:



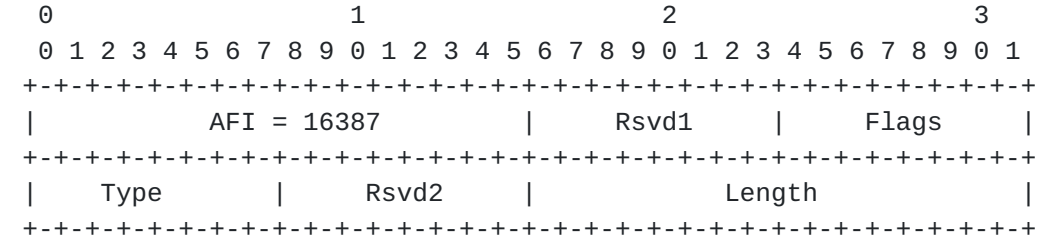
Endpoint ID (EID): a 32-bit (for IPv4) or 128-bit (for IPv6) value used in the source and destination address fields of the first (most inner) LISP header of a packet. The host obtains a destination EID the same way it obtains a destination address today, for example through a DNS lookup or SIP exchange. The source EID is obtained via existing mechanisms used to set a host's "local" IP address. An EID is allocated to a host from an EID-prefix block associated with the site where the host is located. An EID can be used by a host to refer to other hosts.

Routing Locator (RLOC): the IPv4 or IPv6 address of an egress tunnel router (ETR). It is the output of a EID-to-RLOC mapping lookup. An EID maps to one or more RLOCs. Typically, RLOCs are numbered from topologically aggregatable blocks that are assigned to a site at each point to which it attaches to the global Internet; where the topology is defined by the connectivity of provider networks, RLOCs can be thought of as PA addresses. Multiple RLOCs can be assigned to the same ETR device or to multiple ETR devices at a site.

3. LISP Canonical Address Format Encodings

IANA has assigned AFI value 16387 (0x4003) to the LISP architecture and protocols. This specification defines the encoding format of the LISP Canonical Address (LCA).

The first 4 bytes of an LISP Canonical Address are followed by a variable length of fields:



Rsvd1: this 8-bit field is reserved for future use and MUST be transmitted as 0 and ignored on receipt.

Flags: this 8-bit field is for future definition and use. For now, set to zero on transmission and ignored on receipt.

Type: this 8-bit field is specific to the LISP Canonical Address formatted encodings, values are:

Type 0: Null Body Type

Type 1: AFI List Type

Type 2: Instance ID Type

Type 3: AS Number Type

Type 4: Application Data Type

Type 5: Geo Coordinates Type

Type 6: Opaque Key Type

Type 7: NAT-Traversal Type

Type 8: Nonce Locator Type

Type 9: Multicast Info Type

Type 10: Explicit Locator Path Type

Type 11: Security Key Type

Type 12: Source/Dest Key Type

Type 13: Replication List Entry Type

Type 14: JSON Data Model Type

Type 15: Key/Value Address Pair Type

Rsvd2: this 8-bit field is reserved for future use and MUST be transmitted as 0 and ignored on receipt.

Length: this 16-bit field is in units of bytes and covers all of the LISP Canonical Address payload, starting and including the byte after the Length field. So any LCAF encoded address will have a minimum length of 8 bytes when the Length field is 0. The 8 bytes include the AFI, Flags, Type, Reserved, and Length fields. When the AFI is not next to encoded address in a control message, then the encoded address will have a minimum length of 6 bytes when the Length field is 0. The 6 bytes include the Flags, Type, Reserved, and Length fields.

[RFC6830] states RLOC records are sorted when encoded in control messages so the locator-set has consistent order across all xTRs for a given EID. The sort order is based on sort-key {afi, RLOC-address}. When an RLOC is LCAF encoded, the sort-key is {afi, LCAF-Type, payload}. Therefore, when a locator-set has a mix of AFI records and LCAF records, all LCAF records will appear after all the AFI records.

4. LISP Canonical Address Applications

4.1. Segmentation using LISP

When multiple organizations inside of a LISP site are using private addresses [RFC1918] as EID-prefixes, their address spaces must remain segregated due to possible address duplication. An Instance ID in the address encoding can aid in making the entire AFI based address unique.

Another use for the Instance ID LISP Canonical Address Format is when creating multiple segmented VPNs inside of a LISP site where keeping EID-prefix based subnets is desirable.

Instance ID LISP Canonical Address Format:



IID mask-len: if the AFI is set to 0, then this format is not encoding an extended EID-prefix but rather an instance-ID range where the 'IID mask-len' indicates the number of high-order bits used in the Instance ID field for the range.

Length value n: length in bytes of the AFI address that follows the Instance ID field including the AFI field itself.

Instance ID: the low-order 24-bits that can go into a LISP data header when the I-bit is set. See [RFC6830] for details.

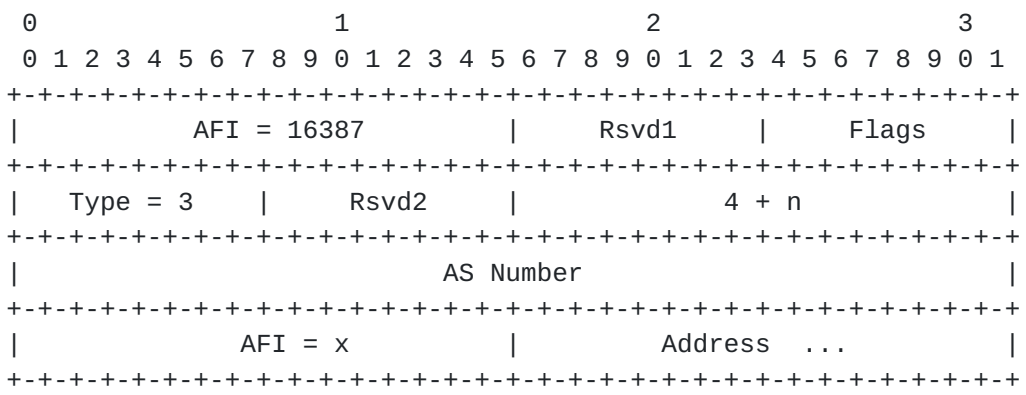
AFI = x: x can be any AFI value from [AFI].

This LISP Canonical Address Type can be used to encode either EID or RLOC addresses.

4.2. Carrying AS Numbers in the Mapping Database

When an AS number is stored in the LISP Mapping Database System for either policy or documentation reasons, it can be encoded in a LISP Canonical Address.

AS Number LISP Canonical Address Format:



Length value n: length in bytes of the AFI address that follows the AS Number field including the AFI field itself.

AS Number: the 32-bit AS number of the autonomous system that has been assigned either the EID or RLOC that follows.

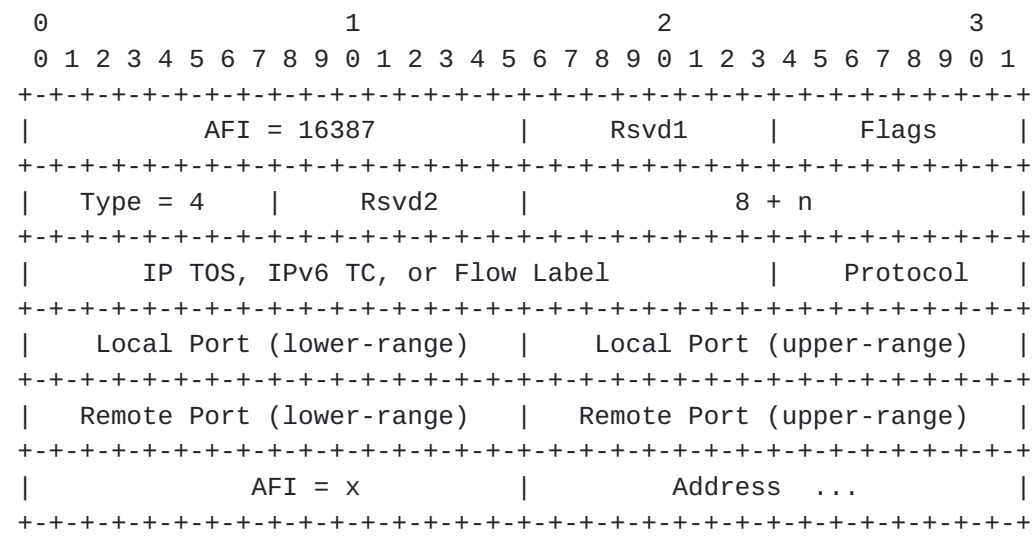
AFI = x: x can be any AFI value from [AFI].

The AS Number Canonical Address Type can be used to encode either EID or RLOC addresses. The former is used to describe the LISP-ALT AS number the EID-prefix for the site is being carried for. The latter is used to describe the AS that is carrying RLOC based prefixes in the underlying routing system.

4.3. Convey Application Specific Data

When a locator-set needs to be conveyed based on the type of application or the Per-Hop Behavior (PHB) of a packet, the Application Data Type can be used.

Application Data LISP Canonical Address Format:



Length value n: length in bytes of the AFI address that follows the 8-byte Application Data fields including the AFI field itself.

IP TOS, IPv6 TC, or Flow Label: this field stores the 8-bit IPv4 TOS field used in an IPv4 header, the 8-bit IPv6 Traffic Class or Flow Label used in an IPv6 header.

Local Port/Remote Port Ranges: these fields are from the TCP, UDP, or SCTP transport header. A range can be specified by using a lower value and an upper value. When a single port is encoded, the lower and upper value fields are the same.

AFI = x: x can be any AFI value from [AFI].

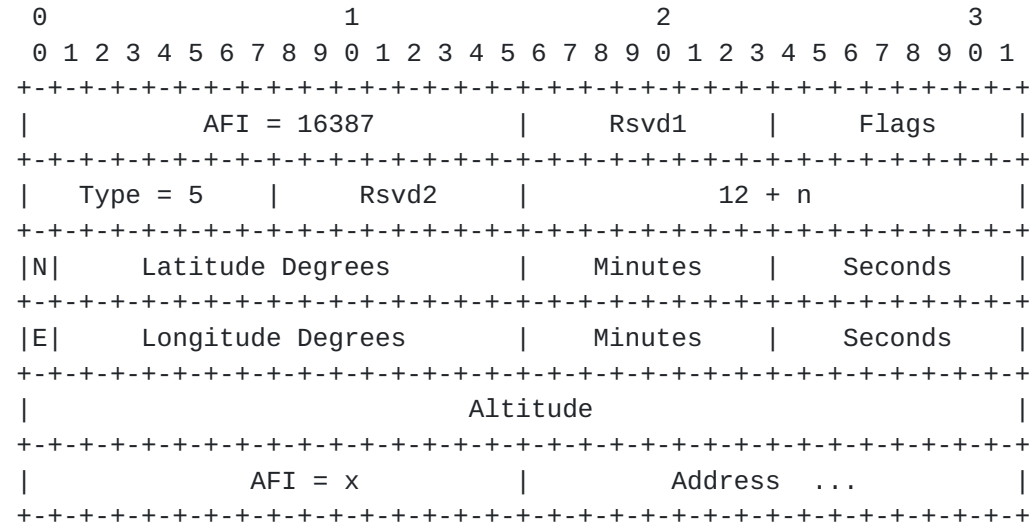
The Application Data Canonical Address Type is used for an EID encoding when an ITR wants a locator-set for a specific application. When used for an RLOC encoding, the ETR is supplying a locator-set for each specific application is has been configured to advertise.

4.4. Assigning Geo Coordinates to Locator Addresses

If an ETR desires to send a Map-Reply describing the Geo Coordinates for each locator in its locator-set, it can use the Geo Coordinate Type to convey physical location information.

Coordinates are specified using the WGS-84 (World Geodetic System) reference coordinate system [WGS-84].

Geo Coordinate LISP Canonical Address Format:



Length value n: length in bytes of the AFI address that follows the 8-byte Longitude and Latitude fields including the AFI field itself.

N: When set to 1 means North, otherwise South.

Latitude Degrees: Valid values range from 0 to 90 degrees above or below the equator (northern or southern hemisphere, respectively).

Latitude Minutes: Valid values range from 0 to 59.

Latitude Seconds: Valid values range from 0 to 59.

E: When set to 1 means East, otherwise West.

Longitude Degrees: Value values are from 0 to 180 degrees right or left of the Prime Meridian.

Longitude Minutes: Valid values range from 0 to 59.

Longitude Seconds: Valid values range from 0 to 59.

Altitude: Height relative to sea level in meters. This is a signed integer meaning that the altitude could be below sea level. A value of 0x7fffffff indicates no Altitude value is encoded.

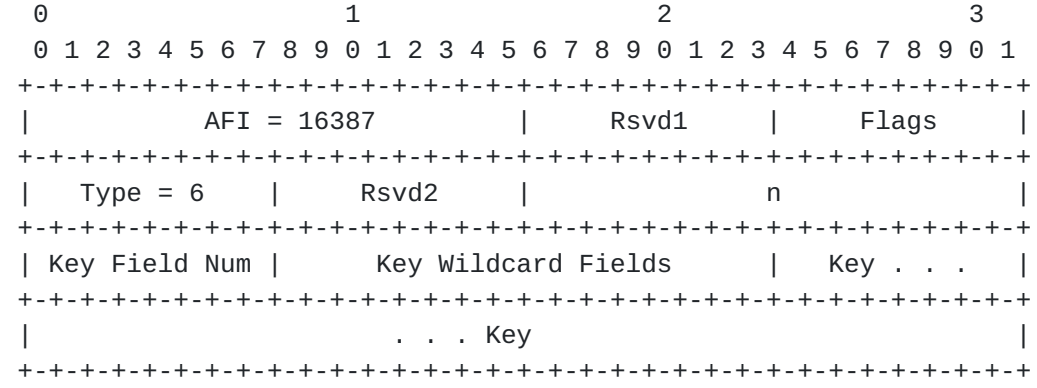
AFI = x: x can be any AFI value from [AFI].

The Geo Coordinates Canonical Address Type can be used to encode either EID or RLOC addresses. When used for EID encodings, you can determine the physical location of an EID along with the topological location by observing the locator-set.

4.5. Generic Database Mapping Lookups

When the LISP Mapping Database system holds information accessed by a generic formatted key (where the key is not the usual IPv4 or IPv6 address), an opaque key may be desirable.

Opaque Key LISP Canonical Address Format:



Length value n: length in bytes of the type's payload. The value n is the number of bytes that follow this Length field.

Key Field Num: the number of fields (minus 1) the key can be broken up into. The width of the fields are fixed length. So for a key size of 8 bytes, with a Key Field Num of 4 allows 4 fields of 2 bytes in length. Valid values for this field range from 0 to 15 supporting a maximum of 16 field separations.

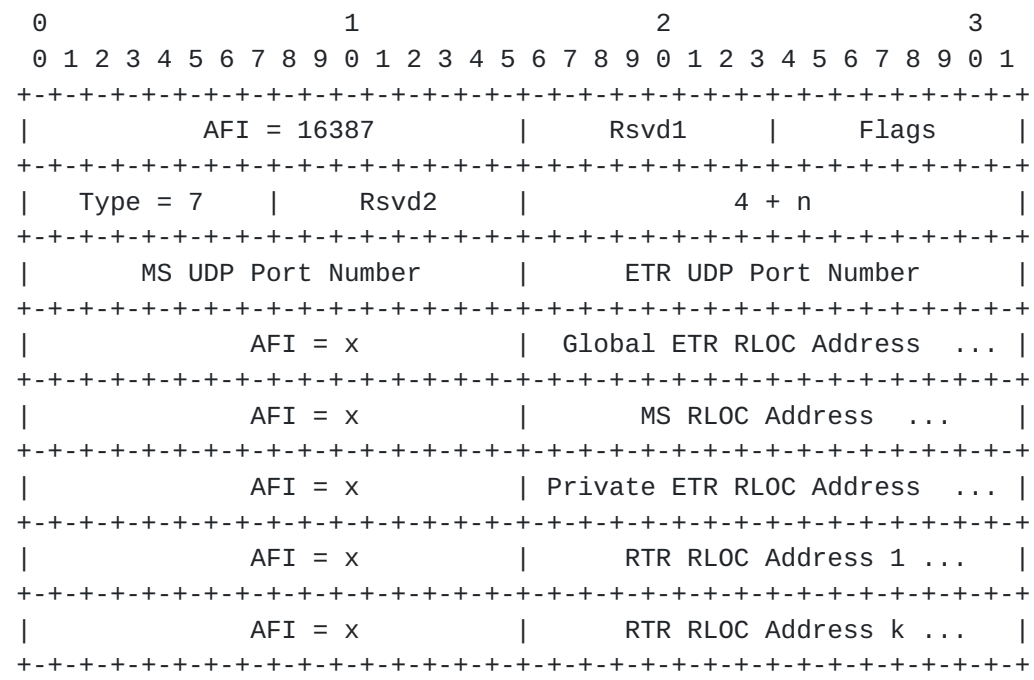
Key Wildcard Fields: describes which fields in the key are not used as part of the key lookup. This wildcard encoding is a bitfield. Each bit is a don't-care bit for a corresponding field in the key. Bit 0 (the low-order bit) in this bitfield corresponds the first field, right-justified in the key, bit 1 the second field, and so on. When a bit is set in the bitfield it is a don't-care bit and should not be considered as part of the database lookup. When the entire 16-bits is set to 0, then all bits of the key are used for the database lookup.

Key: the variable length key used to do a LISP Database Mapping lookup. The length of the key is the value n (shown above) minus 3.

4.6. NAT Traversal Scenarios

When a LISP system is conveying global address and mapped port information when traversing through a NAT device, the NAT-Traversal LCAF Type is used. See [LISP-NATT] for details.

NAT-Traversal Canonical Address Format:



Length value n: length in bytes of the AFI addresses that follows the UDP Port Number field including the AFI fields themselves.

MS UDP Port Number: this is the UDP port number of the Map-Server and is set to 4342.

ETR UDP Port Number: this is the port number returned to a LISP system which was copied from the source port from a packet that has flowed through a NAT device.

AFI = x: x can be any AFI value from [AFI].

Global ETR RLOC Address: this is an address known to be globally unique built by NAT-traversal functionality in a LISP router.

MS RLOC Address: this is the address of the Map-Server used in the destination RLOC of a packet that has flowed through a NAT device.

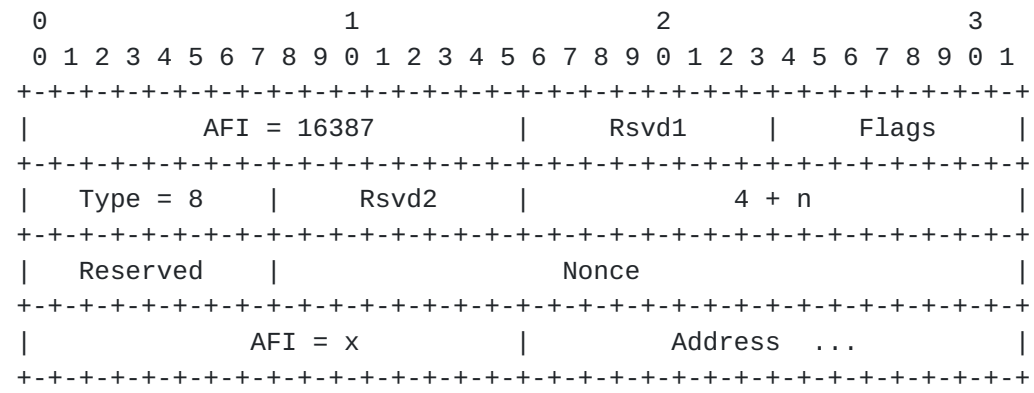
Private ETR RLOC Address: this is an address known to be a private address inserted in this LCAF format by a LISP router that resides on the private side of a NAT device.

RTR RLOC Address: this is an encapsulation address used by an ITR or PITR which resides behind a NAT device. This address is known to have state in a NAT device so packets can flow from it to the LISP ETR behind the NAT. There can be one or more NTR addresses supplied in these set of fields. The number of NTRs encoded is determined by the LCAF length field. When there are no NTRs supplied, the NTR fields can be omitted and reflected by the LCAF length field or an AFI of 0 can be used to indicate zero NTRs encoded.

4.7. PETR Admission Control Functionality

When a public PETR device wants to verify who is encapsulating to it, it can check for a specific nonce value in the LISP encapsulated packet. To convey the nonce to admitted ITRs or PITRs, this LCAF format is used in a Map-Register or Map-Reply locator-record.

Nonce Locator Canonical Address Format:



Length value n: length in bytes of the AFI address that follows the Nonce field including the AFI field itself.

Reserved: must be set to zero and ignore on receipt.

Nonce: this is a nonce value returned by an ETR in a Map-Reply locator-record to be used by an ITR or PITR when encapsulating to the locator address encoded in the AFI field of this LCAF type.

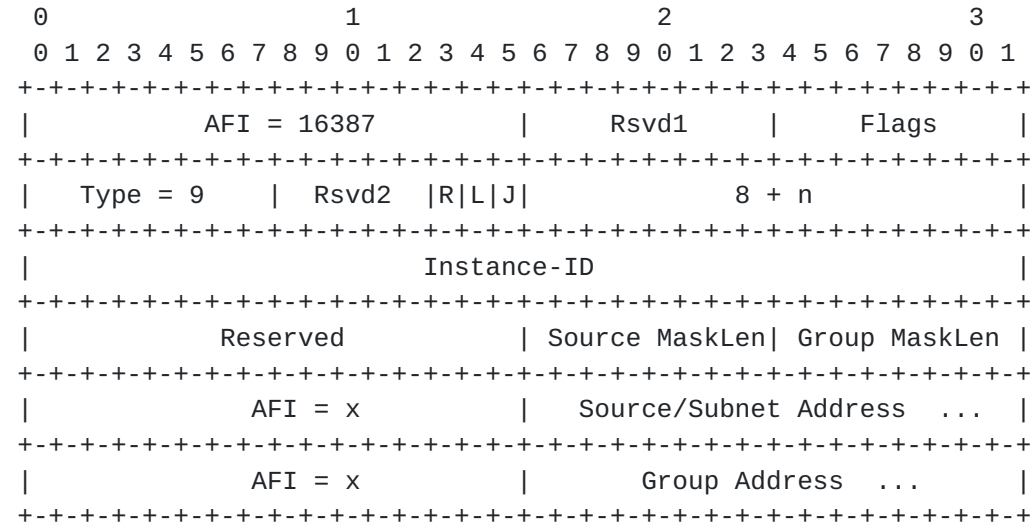
AFI = x: x can be any AFI value from [AFI].

4.8. Multicast Group Membership Information

Multicast group information can be published in the mapping database so a lookup on an EID based group address can return a replication list of group addresses or a unicast addresses for single replication or multiple head-end replications. The intent of this type of unicast replication is to deliver packets to multiple ETRs at receiver LISP multicast sites. The locator-set encoding for this EID record type can be a list of ETRs when they each register with "Merge Semantics". The encoding can be a typical AFI encoded locator address. When an RTR list is being registered (with multiple levels according to [LISP-RE]), the Replication List Entry LCAF type is used for locator encoding.

This LCAF encoding can be used to send broadcast packets to all members of a subnet when each EIDs are away from their home subnet location.

Multicast Info Canonical Address Format:



Length value n: length in bytes of fields that follow.

Reserved: must be set to zero and ignore on receipt.

R-bit: this is the RP-bit that represents PIM (S,G,RP-bit) multicast state. This bit can be set for Joins (when the J-bit is set) or for Leaves (when the L-bit is set). See [LISP-MRSIG] for more usage details.

L-bit: this is the Leave-Request bit and is used when this LCAF type is present in the destination EID-prefix field of a Map-Request. See [LISP-MRSIG] for details.

J-bit: this is the Join-Request bit and is used when this LCAF type is present in the destination EID-prefix field of a Map-Request. See [LISP-MRSIG] for details. The J-bit MUST not be set when the L-bit is also set in the same LCAF block. A receiver should not take any specific Join or Leave action when both bits are set.

Instance ID: the low-order 24-bits that can go into a LISP data header when the I-bit is set. See [RFC6830] for details. The use of the Instance-ID in this LCAF type is to associate a multicast forwarding entry for a given VPN. The instance-ID describes the VPN and is registered to the mapping database system as a 3-tuple of (Instance-ID, S-prefix, G-prefix).

Source MaskLen: the mask length of the source prefix that follows.

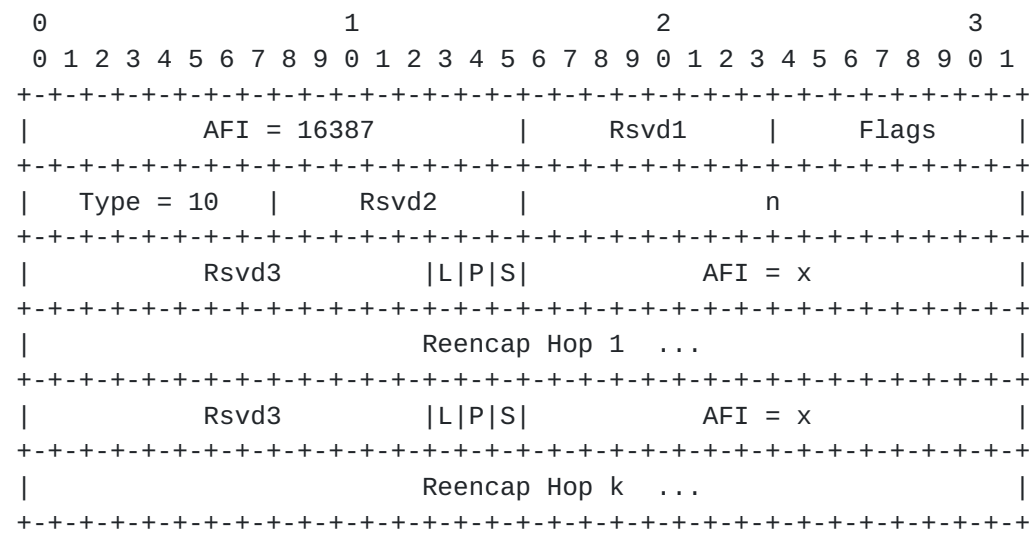
Group MaskLen: the mask length of the group prefix that follows.

AFI = x: x can be any AFI value from [AFI]. When a specific AFI has its own encoding of a multicast address, this field must be either a group address or a broadcast address.

4.9. Traffic Engineering using Re-encapsulating Tunnels

For a given EID lookup into the mapping database, this LCAF format can be returned to provide a list of locators in an explicit re-encapsulation path. See [LISP-TE] for details.

Explicit Locator Path (ELP) Canonical Address Format:



Length value n: length in bytes of fields that follow.

Lookup bit (L): this is the Lookup bit used to indicate to the user of the ELP to not use this address for encapsulation but to look it up in the mapping database system to obtain an encapsulating RLOC address.

RLOC-Probe bit (P): this is the RLOC-probe bit which means the Reencap Hop allows RLOC-probe messages to be sent to it. When the R-bit is set to 0, RLOC-probes must not be sent. When a Reencap Hop is an anycast address then multiple physical Reencap Hops are using the same RLOC address. In this case, RLOC-probes are not needed because when the closest RLOC address is not reachable another RLOC address can reachable.

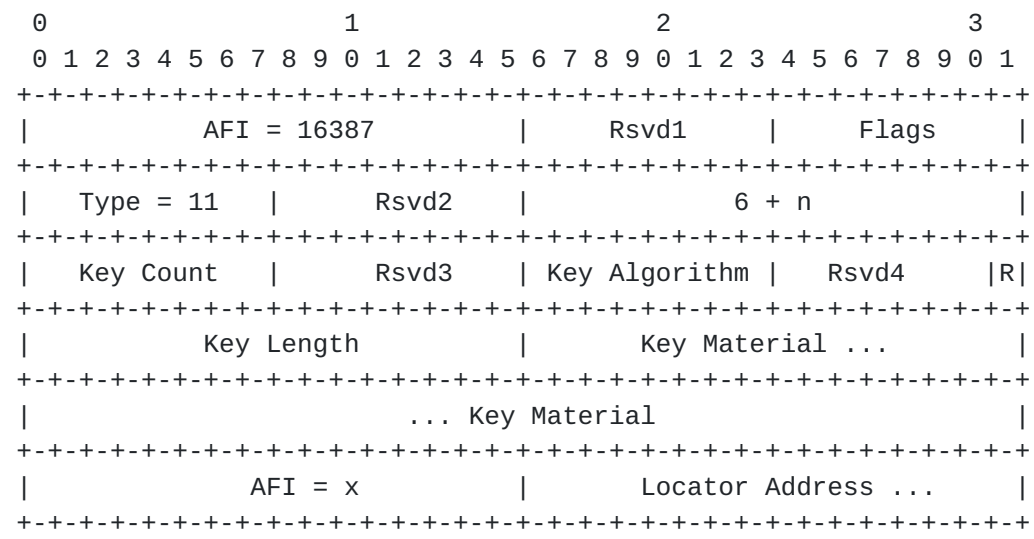
Strict bit (S): this the strict bit which means the associated Rencap Hop is required to be used. If this bit is 0, the reencapsulator can skip this Reencap Hop and go to the next one in the list.

AFI = x: x can be any AFI value from [AFI]. When a specific AFI has its own encoding of a multicast address, this field must be either a group address or a broadcast address.

4.10. Storing Security Data in the Mapping Database

When a locator in a locator-set has a security key associated with it, this LCAF format will be used to encode key material. See [LISP-DDT] for details.

Security Key Canonical Address Format:



Length value n: length in bytes of fields that start with the Key Material field.

Key Count: the Key Count field declares the number of Key sections included in this LCAF.

Key Algorithm: the Algorithm field identifies the key's cryptographic algorithm and specifies the format of the Public Key field.

R bit: this is the revoke bit and, if set, it specifies that this Key is being Revoked.

Key Length: this field determines the length in bytes of the Key Material field.

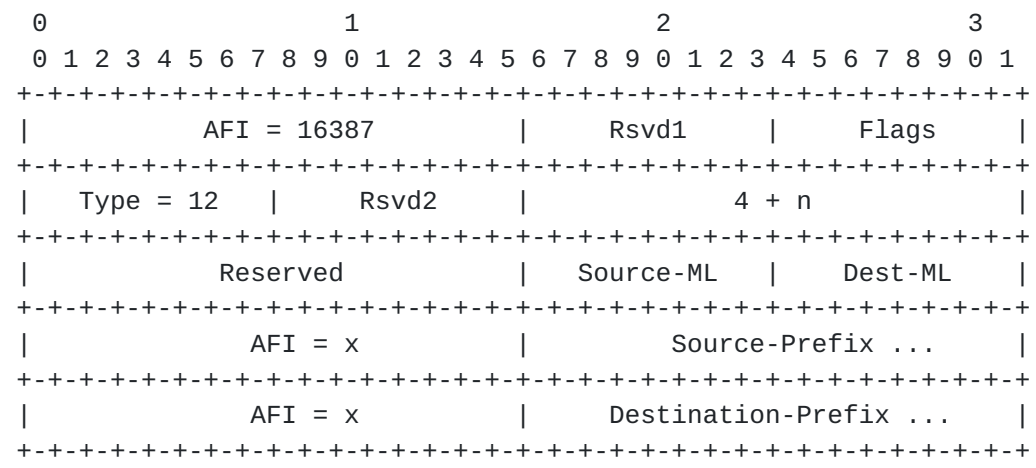
Key Material: the Key Material field stores the key material. The format of the key material stored depends on the Key Algorithm field.

AFI = x: x can be any AFI value from [AFI].This is the locator address that owns the encoded security key.

4.11. Source/Destination 2-Tuple Lookups

When both a source and destination address of a flow needs consideration for different locator-sets, this 2-tuple key is used in EID fields in LISP control messages. When the Source/Dest key is registered to the mapping database, it can be encoded as a source-prefix and destination-prefix. When the Source/Dest is used as a key for a mapping database lookup the source and destination come from a data packet.

Source/Dest Key Canonical Address Format:



Length value n: length in bytes of fields that follow.

Reserved: must be set to zero and ignore on receipt.

Source-ML: the mask length of the source prefix that follows.

Dest-ML: the mask length of the destination prefix that follows.

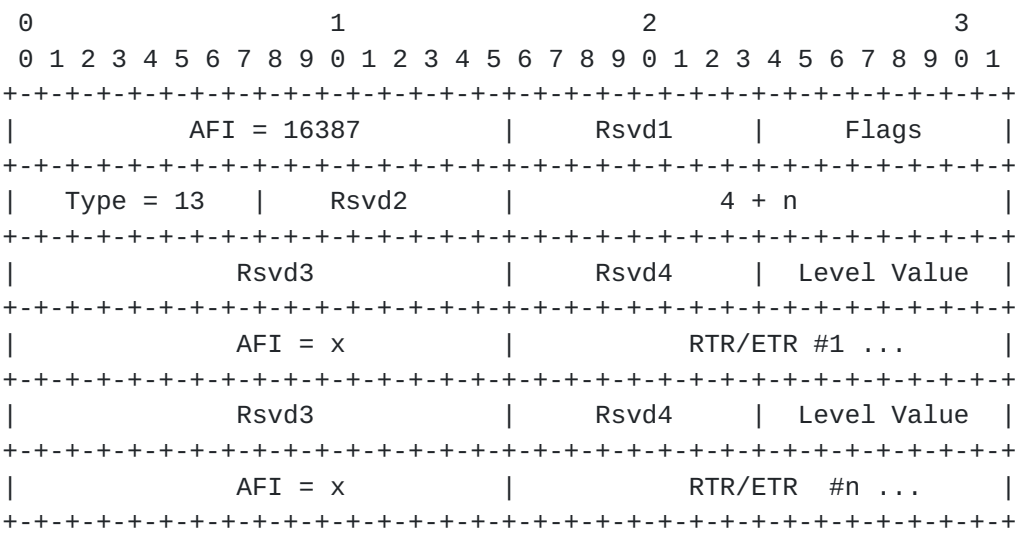
AFI = x: x can be any AFI value from [AFI]. When a specific AFI has its own encoding of a multicast address, this field must be either a group address or a broadcast address.

Refer to [LISP-TE] for usage details.

4.12. Replication List Entries for Multicast Forwarding

The Replication List Entry LCAF type is an encoding for a locator being used for unicast replication according to the specification in [LISP-RE]. This locator encoding is pointed to by a Multicast Info LCAF Type and is registered by Re-encapsulating Tunnel Routers (RTRs) that are participating in an overlay distribution tree. Each RTR will register its locator address and its configured level in the distribution tree.

Replication List Entry Address Format:



Length value n: length in bytes of fields that follow.

Rsvd{1,2,3,4}: must be set to zero and ignore on receipt.

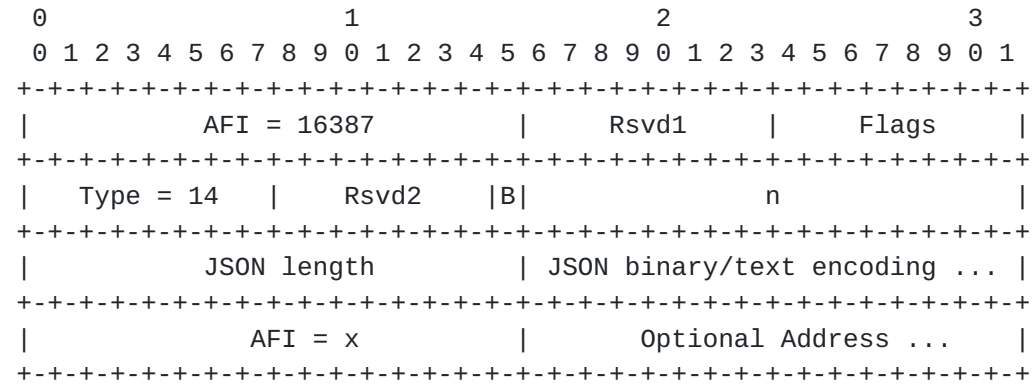
Level Value: this value is associated with the level within the overlay distribution tree hierarchy where the RTR resides. The level numbers are ordered from lowest value being close to the ITR (meaning that ITRs replicate to level-0 RTRs) and higher levels are further downstream on the distribution tree closer to ETRs of multicast receiver sites.

AFI = x: x can be any AFI value from [AFI]. A specific AFI has its own encoding of either a unicast or multicast locator address. All RTR/ETR entries for the same level should be combined together by a Map-Server to avoid searching through the entire multi-level list of locator entries in a Map-Reply message.

4.13. Data Model Encoding

This type allows a JSON data model to be encoded either as an EID or RLOC.

JSON Data Model Type Address Format:



Length value n: length in bytes of fields that follow.

Rsvd{1,2}: must be set to zero and ignore on receipt.

B bit: indicates that the JSON field is binary encoded according to [JSON-BINARY] when the bit is set to 1. Otherwise the encoding is based on text encoding according to [RFC4627].

JSON length: length in octets of the following 'JSON binary/text encoding' field.

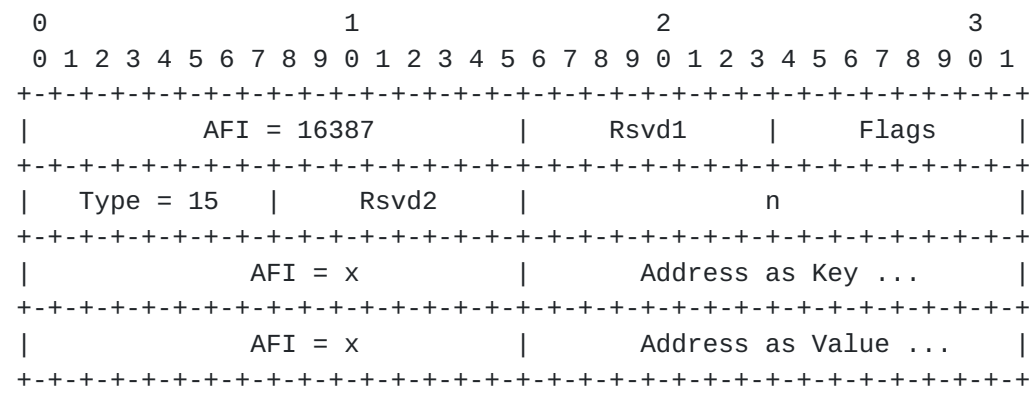
JSON binary/text encoding field: a variable length field that contains either binary or text encodings.

AFI = x: x can be any AFI value from [AFI]. A specific AFI has its own encoding of either a unicast or multicast locator address. All RTR/ETR entries for the same level should be combined together by a Map-Server to avoid searching through the entire multi-level list of locator entries in a Map-Reply message.

4.14. Encoding Key/Value Address Pairs

The Key/Value pair is for example useful for attaching attributes to other elements of LISP packets, such as EIDs or RLOCs. When attaching attributes to EIDs or RLOCs, it's necessary to distinguish between the element that should be used as EID or RLOC, and hence as key for lookups, and additional attributes. This is especially the case when the difference cannot be determined from the types of the elements, such as when two IP addresses are being used.

Key/Value Pair Address Format:



Length value n: length in bytes of fields that follow.

Rsvd{1,2}: must be set to zero and ignore on receipt.

AFI = x: x can be any AFI value from [AFI]. A specific AFI has its own encoding of either a unicast or multicast locator address. All RTR/ETR entries for the same level should be combined together by a Map-Server to avoid searching through the entire multi-level list of locator entries in a Map-Reply message.

Address as Key: this AFI encoded address will be attached with the attributes encoded in "Address as Value" which follows this field.

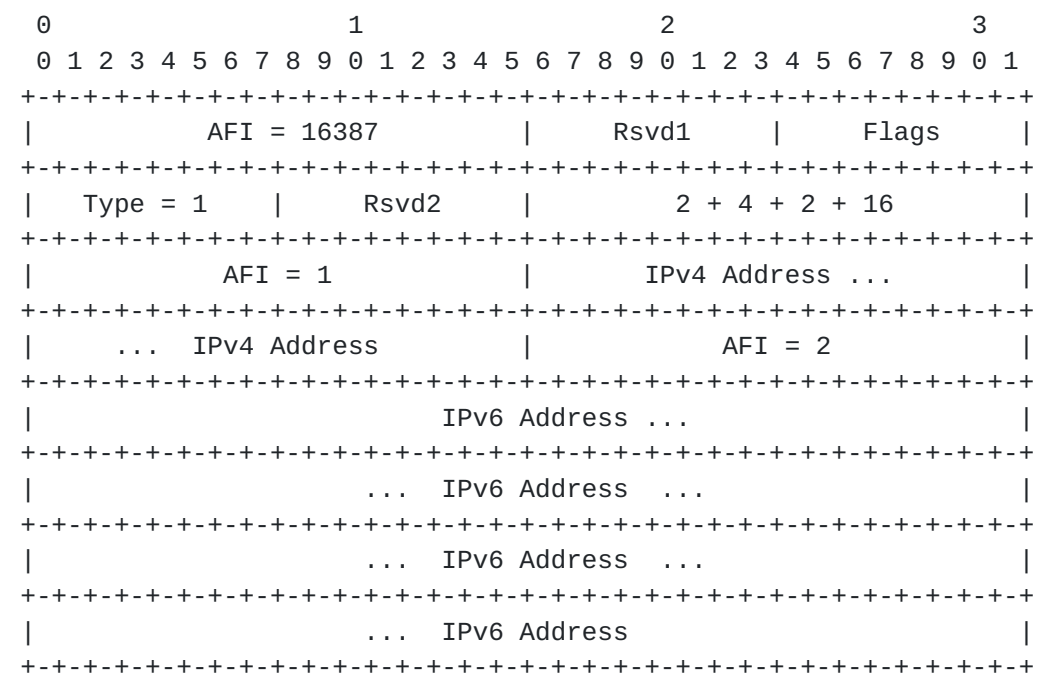
Address as Value: this AFI encoded address will be the attribute address that goes along with "Address as Key" which precedes this field.

4.15. Applications for AFI List Type

4.15.1. Binding IPv4 and IPv6 Addresses

When header translation between IPv4 and IPv6 is desirable a LISP Canonical Address can use the AFI List Type to carry multiple AFIs in one LCAF AFI.

Address Binding LISP Canonical Address Format:



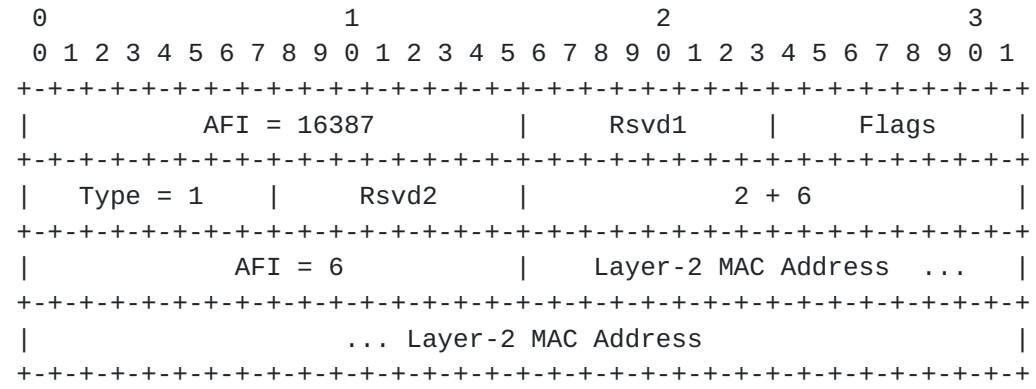
Length: length in bytes is fixed at 24 when IPv4 and IPv6 AFI encoded addresses are used.

This type of address format can be included in a Map-Request when the address is being used as an EID, but the Mapping Database System lookup destination can use only the IPv4 address. This is so a Mapping Database Service Transport System, such as LISP-ALT [RFC6836], can use the Map-Request destination address to route the control message to the desired LISP site.

4.15.2. Layer-2 VPNs

When MAC addresses are stored in the LISP Mapping Database System, the AFI List Type can be used to carry AFI 6.

MAC Address LISP Canonical Address Format:



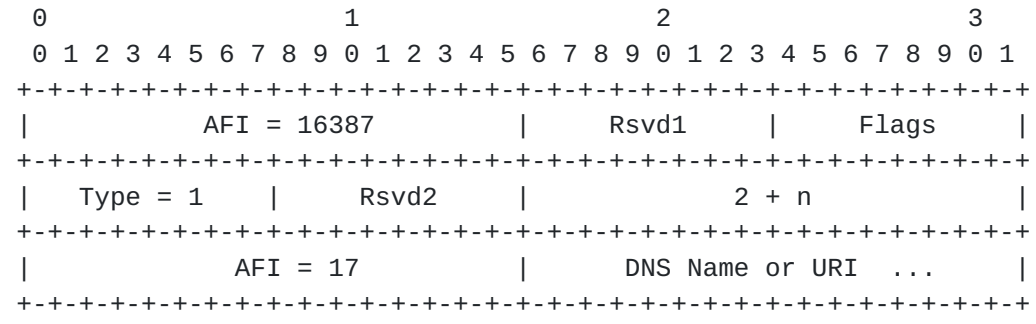
Length: length in bytes is fixed at 8 when MAC address AFI encoded addresses are used.

This address format can be used to connect layer-2 domains together using LISP over an IPv4 or IPv6 core network to create a layer-2 VPN. In this use-case, a MAC address is being used as an EID, and the locator-set that this EID maps to can be an IPv4 or IPv6 RLOCs, or even another MAC address being used as an RLOC.

4.15.3. ASCII Names in the Mapping Database

If DNS names or URIs are stored in the LISP Mapping Database System, the AFI List Type can be used to carry an ASCII string where it is delimited by length 'n' of the LCAF Length encoding.

ASCII LISP Canonical Address Format:

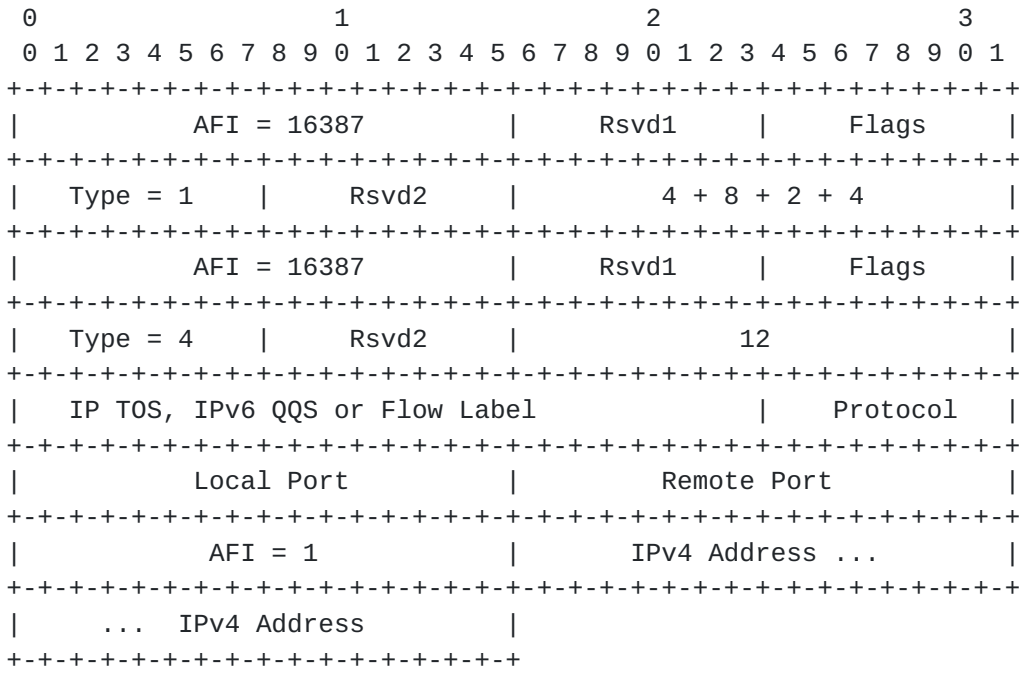


Length value n: length in bytes AFI=17 field and the null-terminated ASCII string (the last byte of 0 is included).

4.15.4. Using Recursive LISP Canonical Address Encodings

When any combination of above is desirable, the AFI List Type value can be used to carry within the LCAF AFI another LCAF AFI.

Recursive LISP Canonical Address Format:



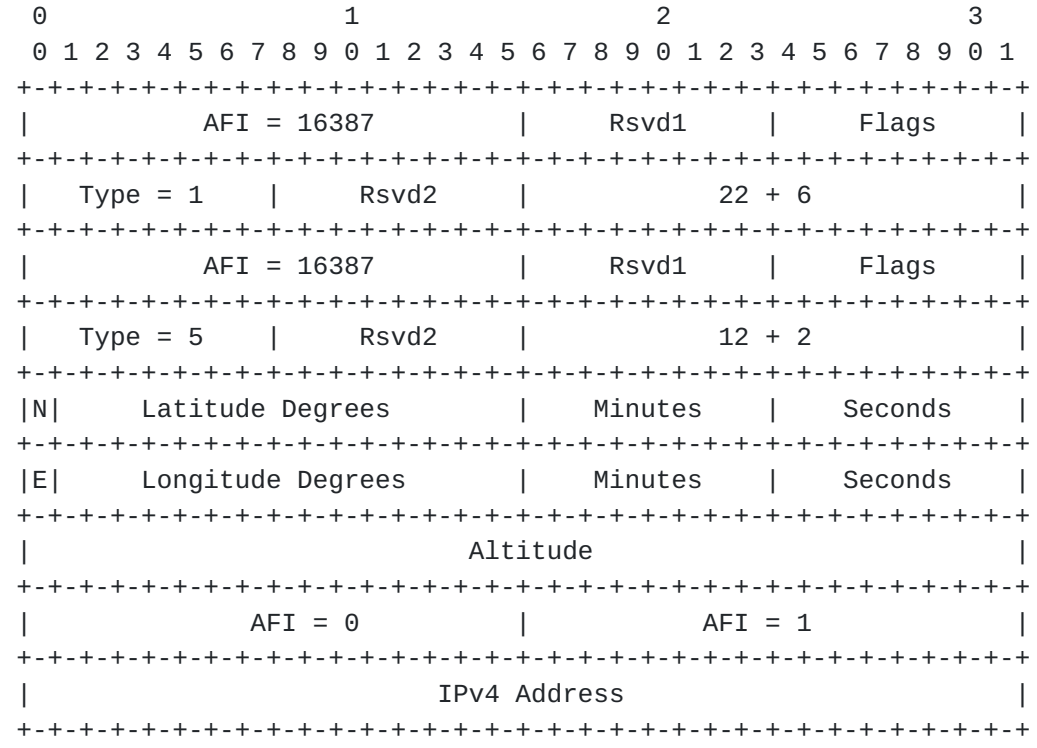
Length: length in bytes is fixed at 18 when an AFI=1 IPv4 address is included.

This format could be used by a Mapping Database Transport System, such as LISP-ALT [RFC6836], where the AFI=1 IPv4 address is used as an EID and placed in the Map-Request destination address by the sending LISP system. The ALT system can deliver the Map-Request to the LISP destination site independent of the Application Data Type AFI payload values. When this AFI is processed by the destination LISP site, it can return different locator-sets based on the type of application or level of service that is being requested.

4.15.5. Compatibility Mode Use Case

A LISP system should use the AFI List Type format when sending to LISP systems that do not support a particular LCAF Type used to encode locators. This allows the receiving system to be able to parse a locator address for encapsulation purposes. The list of AFIs in an AFI List LCAF Type has no semantic ordering and a receiver should parse each AFI element no matter what the ordering.

Compatibility Mode Address Format:



If a system does not recognized the Geo Coordinate LCAF Type that is accompanying a locator address, an encoder can include the Geo Coordinate LCAF Type embedded in a AFI List LCAF Type where the AFI in the Geo Coordinate LCAF is set to 0 and the AFI encoded next in the list is encoded with a valid AFI value to identify the locator address.

A LISP system is required to support the AFI List LCAF Type to use this procedure. It would skip over 10 bytes of the Geo Coordinate LCAF Type to get to the locator address encoding (an IPv4 locator address). A LISP system that does support the Geo Coordinate LCAF Type can support parsing the locator address within the Geo Coordinate LCAF encoding or in the locator encoding that follows in the AFI List LCAF.

5. Security Considerations

There are no security considerations for this specification. The security considerations are documented for the protocols that use LISP Canonical Addressing. Refer to the those relevant specifications.

6. IANA Considerations

The Address Family AFI definitions from [AFI] only allocate code-points for the AFI value itself. The length of the address or entity that follows is not defined and is implied based on conventional experience. Where the LISP protocol uses LISP Canonical Addresses specifically, the address length definitions will be in this specification and take precedent over any other specification.

An IANA Registry for LCAF Type values will be created. The values that are considered for use by the main LISP specification [RFC6830] will be in the IANA Registry. Other Type values used for experimentation will be defined and described in this document.

7. References

7.1. Normative References

[RFC1700] Reynolds, J. and J. Postel, "Assigned Numbers", RFC 1700, October 1994.

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC4627] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", RFC 4627, July 2006.

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

[RFC6836] Fuller, V., Farinacci, D., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol Alternative Logical Topology (LISP+ALT)", RFC 6836, January 2013.

7.2. Informative References

[AFI] IANA, , "Address Family Identifier (AFIs)", ADDRESS FAMILY NUMBERS http://www.iana.org/numbers.html, Febuary 2007.

[JSON-BINARY]
"Universal Binary JSON Specification", URL http://ubjson.org, .

[LISP-DDT]
Fuller, V., Lewis, D., and V. Ermagan, "LISP Delegated Database Tree", draft-ietf-lisp-ddt-01.txt (work in progress), .

[LISP-MRSIG]
Farinacci, D. and M. Napierala, "LISP Control-Plane Multicast Signaling", draft-farinacci-lisp-mr-signaling-03.txt (work in progress), .

[LISP-NATT]
Ermagan, V., Farinacci, D., Lewis, D., Skriver, J., Maino, F., and C. White, "NAT traversal for LISP", draft-ermagan-lisp-nat-traversal-03.txt (work in progress), .

[LISP-RE] Coras, F., Cabellos-Aparicio, A., Domingo-Pascual, J., Maino, F., and D. Farinacci, "LISP Replication Engineering", draft-coras-lisp-re-03.txt (work in progress), .

[LISP-TE] Farinacci, D., Lahiri, P., and M. Kowal, "LISP Traffic Engineering Use-Cases", draft-farinacci-lisp-te-03.txt (work in progress), .

[WGS-84] Geodesy and Geophysics Department, DoD., "World Geodetic System 1984", NIMA TR8350.2, January 2000, <http://earth-info.nga.mil/GandG/publications/tr8350.2/wgs84fin.pdf>.

Appendix A. Acknowledgments

The authors would like to thank Vince Fuller, Gregg Schudel, Jesper Skriver, Luigi Iannone, Isidor Kouvelas, and Sander Steffann for their technical and editorial commentary.

The authors would like to thank Victor Moreno for discussions that lead to the definition of the Multicast Info LCAF type.

The authors would like to thank Parantap Lahiri and Michael Kowal for discussions that lead to the definition of the Explicit Locator Path (ELP) LCAF type.

The authors would like to thank Fabio Maino and Vina Ermagan for discussions that lead to the definition of the Security Key LCAF type.

The authors would like to thank Albert Cabellos-Aparicio and Florin Coras for discussions that lead to the definition of the Replication List Entry LCAF type.

Thanks goes to Michiel Blokzijl and Alberto Rodriguez-Natal for suggesting new LCAF types.

Thanks also goes to Terry Manderson for assistance obtaining a LISP AFI value from IANA.

Appendix B. Document Change Log

B.1. Changes to draft-ietf-lisp-lcaf-06.txt

- Submitted October 2014.
- Make it clear how sorted RLOC records are done when LCAFs are used as the RLOC record.

B.2. Changes to draft-ietf-lisp-lcaf-05.txt

- Submitted May 2014.
- Add a length field of the JSON payload that can be used for either binary or text encoding of JSON data.

B.3. Changes to draft-ietf-lisp-lcaf-04.txt

- Submitted January 2014.
- Agreement among ELP implementors to have the AFI 16-bit field adjacent to the address. This will make the encoding consistent with all other LCAF type address encodings.

B.4. Changes to draft-ietf-lisp-lcaf-03.txt

- Submitted September 2013.
- Updated references and author's affilations.
- Added Instance-ID to the Multicast Info Type so there is relative ease in parsing (S,G) entries within a VPN.
- Add port range encodings to the Application Data LCAF Type.
- Add a new JSON LCAF Type.

- Add Address Key/Value LCAF Type to allow attributes to be attached to an address.

B.5. Changes to draft-ietf-lisp-lcaf-02.txt

- Submitted March 2013.
- Added new LCAF Type "Replication List Entry" to support LISP replication engineering use-cases.
- Changed references to new LISP RFCs.

B.6. Changes to draft-ietf-lisp-lcaf-01.txt

- Submitted January 2013.
- Change longitude range from 0-90 to 0-180 in section 4.4.
- Added reference to WGS-84 in section 4.4.

B.7. Changes to draft-ietf-lisp-lcaf-00.txt

- Posted first working group draft August 2012.
- This draft was renamed from draft-farinacci-lisp-lcaf-10.txt.

Authors' Addresses

Dino Farinacci
lispers.net
San Jose, CA
USA

Email: farinacci@gmail.com

Dave Meyer
Brocade
San Jose, CA
USA

Email: dmm@1-4-5.net

Job Snijders
Hibernia Networks
Tupolevlaan 103a
Schiphol-Rijk 1119 PA
NL

Email: job.snijders@hibernianetworks.com