Internet Engineering Task Force F. Maino, Ed.
Internet-Draft Cisco
Intended status: Standards Track J. Lemon
Expires: May 7, 2020 Broadcom
P. Agarwal
Innovium
D. Lewis
M. Smith
Cisco
November 4, 2019

LISP Generic Protocol Extension
draft-ietf-lisp-gpe-11

Abstract

This document describes extentions to the Locator/ID Separation Protocol (LISP) Data-Plane, via changes to the LISP header, to support multi-protocol encapsulation.

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 May 7, 2020.

Table of Contents

1. Introduction

The LISP Data-Plane is defined in [I-D.ietf-lisp-rfc6830bis]. It specifies an encapsulation format that carries IPv4 or IPv6 packets (henceforth jointly referred to as IP) in a LISP header and outer UDP/IP transport.

The LISP Data-Plane header does not specify the protocol being encapsulated and therefore is currently limited to encapsulating only

skipping to change at *page 4, line 38*

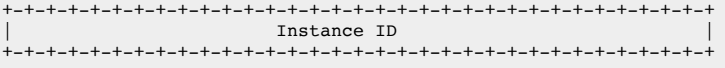


Figure 2: LISP-GPE Header

Internet Engineering Task Force F. Maino, Ed.
Internet-Draft Cisco
Intended status: Standards Track J. Lemon
Expires: May 22, 2020 Broadcom
P. Agarwal
Innovium
D. Lewis
M. Smith
Cisco
November 19, 2019

LISP Generic Protocol Extension
draft-ietf-lisp-gpe-12

Abstract

This document describes extentions to the Locator/ID Separation Protocol (LISP) Data-Plane, via changes to the LISP header, to support multi-protocol encapsulation.

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 https://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 May 22, 2020.

Table of Contents

1. Introduction

The LISP Data-Plane is defined in [I-D.ietf-lisp-rfc6830bis]. It specifies an encapsulation format that carries IPv4 or IPv6 packets (henceforth jointly referred to as IP) in a LISP header and outer UDP/IP transport.

The LISP Data-Plane header does not specify the protocol being encapsulated and therefore is currently limited to encapsulating only

skipping to change at *page 4, line 36*

Figure 2: LISP-GPE Header

Bits 0-3 and 8-23: Bits 0-3 and 8-23 of the LISP-GPE header are Reserved. They MUST be set to zero on transmission and ignored on receipt.

Features that were implemented with bits 0-3 in [I-D.ietf-lisp-rfc6830bis], such as echo-noncing, map-versioning and reachability, can be implemented by defining the appropriate shim headers.

Instance ID When the I-Bit is set to 1 the high-order 24 bits of the Instance ID field are used as an Instance ID, as specified in [I-D.ietf-lisp-rfc6830bis]. The low-order 8 bits are set to zero, as the Locator-Status-Bits feature is not supported in LISP-GPE.

P-Bit: Flag bit 5 is defined as the Next Protocol bit.

Bits 0-3 and 8-23: Bits 0-3 and 8-23 of the LISP-GPE header are Reserved. They MUST be set to zero on transmission and ignored on receipt.

Features that were implemented with bits 0-3 and 8-23 in [I-D.ietf-lisp-rfc6830bis], such as echo-noncing, map-versioning and reachability, can be implemented by defining the appropriate shim headers.

Instance ID When the I-Bit is set to 1 the high-order 24 bits of the Instance ID field are used as an Instance ID, as specified in [I-D.ietf-lisp-rfc6830bis]. The low-order 8 bits are set to zero, as the Locator-Status-Bits feature is not supported in LISP-GPE.

P-Bit: Flag bit 5 is defined as the Next Protocol bit.

skipping to change at *page 5, line 34*

skipping to change at *page 5, line 32*

0x02 : IPv6

0x03 : Ethernet

0x04 : Network Service Header (NSH) [RFC8300]

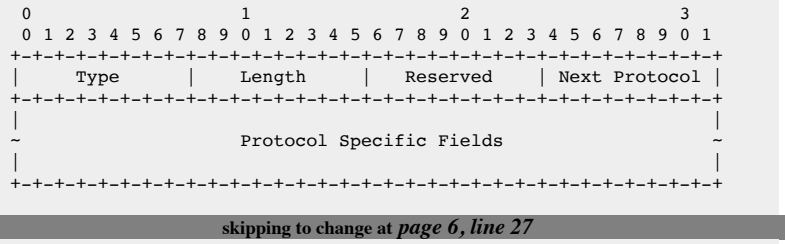
0x05 to 0x7F: Unassigned

0x80 to 0xFF: Unassigned (shim headers)

The values are tracked in an IANA registry as described in Section 6.1.

Next protocol values from 0x80 to 0xFF are assigned to protocols encoded as generic "shim" headers. Shim protocols all use a common header structure, which includes a next header field using the same values as described above. When a shim header protocol is used with other data described by protocols identified by next protocol values from 0x0 to 0x7F, the shim header MUST come before the further protocol, and the next header of the shim will indicate what follows the shim protocol.

Implementations that are not aware of a given shim header MUST ignore the header and proceed to parse the next protocol. Shim protocols MUST have the first 32 bits defined as:



0x02 : IPv6

0x03 : Ethernet

0x04 : Network Service Header (NSH) [RFC8300]

0x05 to 0x7F: Unassigned

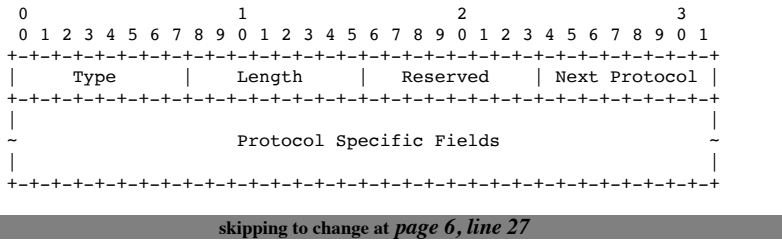
0x80 to 0xFF: Unassigned (shim headers)

The values are tracked in the IANA LISP-GPE Next Protocol Registry as described in Section 6.1.

Next protocol values from 0x80 to 0xFF are assigned to protocols encoded as generic "shim" headers. Shim protocols all use a common header structure, which includes a next header field using the same values as described above. When a shim header protocol is used with other data described by protocols identified by next protocol values from 0x0 to 0x7F, the shim header MUST come before the further protocol, and the next header of the shim will indicate what follows the shim protocol.

Shim headers can be used to incrementally deploy new GPE features, keeping the processing of shim headers known to a given xTR implementation in the 'fast' path (typically an ASIC), while punting the processing of the remaining new GPE features to the 'slow' path.

Shim protocols MUST have the first 32 bits defined as:



skipping to change at *page 6, line 27*

skipping to change at *page 6, line 27*

Where:

Type: This field identifies the different messages of this protocol.

Length: The length, in 4-octect units, of this protocol message not including the first 4 octects.

Reserved: The use of this field is reserved to the protocol defined in this message.

Next Protocol Field: This next protocol field contains the protocol of the encapsulated payload. The protocol registry will be requested from IANA as per section 10.2.

Where:

Type: This field identifies the different messages of this protocol.

Length: The length, in 4-octect units, of this protocol message not including the first 4 octects.

Reserved: The use of this field is reserved to the protocol defined in this message.

Next Protocol Field: The next protocol field contains the protocol of the encapsulated payload. The values are tracked in the IANA LISP-GPE Next Protocol Registry as described in Section 6.1.

4. Implementation and Deployment Considerations

4.1. Applicability Statement

LISP-GPE conforms, as an UDP-based encapsulation protocol, to the UDP usage guidelines as specified in [RFC8085]. The applicability of these guidelines are dependent on the underlay IP network and the nature of the encapsulated payload.

4. Implementation and Deployment Considerations

4.1. Applicability Statement

LISP-GPE conforms, as an UDP-based encapsulation protocol, to the UDP usage guidelines as specified in [RFC8085]. The applicability of these guidelines are dependent on the underlay IP network and the nature of the encapsulated payload.

skipping to change at *page 10, line 9*

skipping to change at *page 10, line 9*

Class' field.

When a LISP-GPE router performs Ethernet encapsulation, the inner header 802.1Q [IEEE.802.1Q_2014] VLAN Identifier (VID) MAY be mapped to, or used to determine the LISP Instance IDentifier (IID) field.

5. Backward Compatibility

LISP-GPE uses the same UDP destination port (4341) allocated to LISP.

The next Section describes a method to determine the Data-Plane capabilities of a LISP ETR, based on the use of the "Multiple Data-Planes" LISP Canonical Address Format (LCAF) type defined in [RFC8060]. Other mechanisms can be used, including static ETR/ITR (xTR) configuration, but are out of the scope of this document.

Class' field.

When a LISP-GPE router performs Ethernet encapsulation, the inner header 802.1Q [IEEE.802.1Q_2014] VLAN Identifier (VID) MAY be mapped to, or used to determine the LISP Instance IDentifier (IID) field.

5. Backward Compatibility

LISP-GPE uses the same UDP destination port (4341) allocated to LISP.

When encapsulating IP packets to a non LISP-GPE capable router the P-bit MUST be set to 0. That is, the encapsulation format defined in this document MUST NOT be sent to a router that has not indicated that it supports this specification because such a router would ignore the P-bit (as described in [I-D.ietf-lisp-rfc6830bis]) and so would misinterpret the other LISP header fields possibly causing significant errors.

5.1. Use of "Multiple Data-Planes" LCAF to Determine ETR Capabilities

LISP Canonical Address Format (LCAF) [RFC8060] defines the "Multiple Data-Planes" LCAF type, that can be included by an ETR in a Map-Reply to encode the encapsulation formats supported by a given RLOC. In this way an ITR can be made aware of the capability to support LISP-GPE, as well as other encapsulations, on a given RLOC of that ETR.

The 3rd 32-bit word of the "Multiple Data-Planes" LCAF type, as defined in [RFC8060], is a bitmap whose bits are set to one (1) to represent support for each Data-Plane encapsulation. The values are tracked in an IANA registry as described in Section 6.2.

This document defines bit 24 in the third 32-bit word of the "Multiple Data-Planes" LCAF as:

g-Bit: The RLOCs listed in the Address Family Identifier (AFI) encoded addresses in the next longword can accept LISP-GPE (Generic Protocol Extension) encapsulation using destination UDP port 4341

6. IANA Considerations

6.1. LISP-GPE Next Protocol Registry

IANA is requested to set up a registry of LISP-GPE "Next Protocol". These are 8-bit values. Next Protocol values in the table below are defined in this document. New values are assigned under the Specification Required policy [RFC8126]. The protocols that are being assigned values do not themselves need to be IETF standards

skipping to change at *page 11, line 19*

0x00	Reserved	This Document
0x01	IPv4	This Document
0x02	IPv6	This Document
0x03	Ethernet	This Document
0x04	NSH	This Document
0x05..0x7F	Unassigned	
0x82..0xFF	Unassigned	

6.2. Multiple Data-Planes Encapsulation Bitmap Registry

IANA is requested to set up a registry of "Multiple Data-Planes Encapsulation Bitmap" to identify the encapsulations supported by an ETR in the Multiple Data-Planes LCAF Type defined in [RFC8060]. The bitmap is the 3rd 32-bit word of the Multiple Data-Planes LCAF type. Each bit of the bitmap represents a Data-Plane Encapsulation. New values are assigned under the Specification Required policy [RFC8126].

Bits 0-23 are unassigned. This document assigns bits 24-31. Bit 24 (bit 'g') is assigned to LISP-GPE.

Bit Position	Bit Name	Assigned to	Reference
0-23		Unassigned	
24	g	LISP Generic Protocol Extension (LISP-GPE)	This Document
25	U	Generic UDP Encapsulation (GUE)	This Document
26	G	Generic Network Virtualization Encapsulation (GENEVE)	This Document
27	N	Network Virtualization - Generic Routing Encapsulation (NV-GRE)	This Document
28	v	VXLAN Generic Protocol Extension (VXLAN-GPE)	This Document
29	V	Virtual eXtensible Local Area Network (VXLAN)	This Document
30	l	Layer 2 LISP (LISP-L2)	This Document
31	L	Locator/ID Separation Protocol (LISP)	This Document

Editorial Note (The following paragraph to be removed by the RFC Editor before publication)

The "Multiple Data-Planes Encapsulation Bitmap" was "hardcoded" in RFC8060, assigning values to bits 25-31. This draft allocates the "Multiple Data-Planes Encapsulation Bitmap" registry assigning a value to bit 24 for the LISP-GPE encapsulation, assigning bits 25-31 values that are conformant with RFC8060. This will allow future allocation of values 0-23.

7. Security Considerations

When encapsulating IP packets to a non LISP-GPE capable router the P-bit MUST be set to 0. That is, the encapsulation format defined in this document MUST NOT be sent to a router that has not indicated that it supports this specification because such a router would ignore the P-bit (as described in [I-D.ietf-lisp-rfc6830bis]) and so would misinterpret the other LISP header fields possibly causing significant errors.

5.1. Detection of ETR Capabilities

The detection of ETR capabilities to support multiple data plane encapsulations and shim headers is out of the scope of this document. Given that the applicability domain of LISP-GPE is a traffic-managed controlled environment, ITR/ETR (xTR) configuration mechanisms may be used for this purpose.

6. IANA Considerations

6.1. LISP-GPE Next Protocol Registry

IANA is requested to set up a registry of LISP-GPE "Next Protocol". These are 8-bit values. Next Protocol values in the table below are defined in this document. New values are assigned under the Specification Required policy [RFC8126]. The protocols that are being assigned values do not themselves need to be IETF standards

skipping to change at *page 10, line 48*

0x00	Reserved	This Document
0x01	IPv4	This Document
0x02	IPv6	This Document
0x03	Ethernet	This Document
0x04	NSH	This Document
0x05..0x7F	Unassigned	
0x82..0xFF	Unassigned	

7. Security Considerations

LISP-GPE security considerations are similar to the LISP security considerations and mitigation techniques documented in [RFC7835].

LISP-GPE, as many encapsulations that use optional extensions, is subject to on-path adversaries that by manipulating the g-Bit and the packet itself can remove part of the payload. Typical integrity protection mechanisms (such as IPsec) SHOULD be used in combination with LISP-GPE by those protocol extensions that want to protect from

LISP-GPE security considerations are similar to the LISP security considerations and mitigation techniques documented in [RFC7835].

LISP-GPE, as many encapsulations that use optional extensions, is subject to on-path adversaries that by manipulating the g-Bit and the packet itself can remove part of the payload. Typical integrity protection mechanisms (such as IPsec) SHOULD be used in combination with LISP-GPE by those protocol extensions that want to protect from

skipping to change at *page 14, line 7*

skipping to change at *page 12, line 14*

[IEEE.802.1Q_2014]
IEEE, "IEEE Standard for Local and metropolitan area networks--Bridges and Bridged Networks", IEEE 802.1Q-2014, DOI 10.1109/ieeestd.2014.6991462, December 2014, <http://ieeexplore.ieee.org/servlet/opac?punumber=6991460>.

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

[RFC6040] Briscoe, B., "Tunnelling of Explicit Congestion Notification", RFC 6040, DOI 10.17487/RFC6040, November 2010, <https://www.rfc-editor.org/info/rfc6040>.

9.2. Informative References

[I-D.brockners-ippm-ioam-vxlan-gpe]
Brockners, F., Bhandari, S., Govindan, V., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A.,

skipping to change at *page 14, line 40*

skipping to change at *page 12, line 47*

Lemon, J., Maino, F., Smith, M., and A. Isaac, "Group Policy Encoding with VXLAN-GPE and LISP-GPE", draft-lemon-vxlan-lisp-gpe-gbp-02 (work in progress), April 2019.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <https://www.rfc-editor.org/info/rfc2460>.

[RFC6935] Eubanks, M., Chimento, P., and M. Westerlund, "IPv6 and UDP Checksums for Tunneled Packets", RFC 6935, DOI 10.17487/RFC6935, April 2013, <https://www.rfc-editor.org/info/rfc6935>.

[RFC6936] Fairhurst, G. and M. Westerlund, "Applicability Statement for the Use of IPv6 UDP Datagrams with Zero Checksums", RFC 6936, DOI 10.17487/RFC6936, April 2013, <https://www.rfc-editor.org/info/rfc6936>.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <https://www.rfc-editor.org/info/rfc7348>.

[RFC7835] Saucez, D., Iannone, L., and O. Bonaventure, "Locator/ID Separation Protocol (LISP) Threat Analysis", RFC 7835, DOI 10.17487/RFC7835, April 2016, <https://www.rfc-editor.org/info/rfc7835>.

[RFC8060] Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format (LCAF)", RFC 8060, DOI 10.17487/RFC8060, February 2017, <https://www.rfc-editor.org/info/rfc8060>.

[RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <https://www.rfc-editor.org/info/rfc8085>.

[RFC8086] Yong, L., Ed., Crabbe, E., Xu, X., and T. Herbert, "GRE-in-UDP Encapsulation", RFC 8086, DOI 10.17487/RFC8086, March 2017, <https://www.rfc-editor.org/info/rfc8086>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <https://www.rfc-editor.org/info/rfc8300>.

Authors' Addresses

Fabio Maino (editor)
Cisco Systems
San Jose, CA 95134
USA

Email: fmaino@cisco.com
Jennifer Lemon

End of changes. 21 change blocks.

114 lines changed or deleted ***45 lines changed or added***