ï»¿

Internet Engineering Task Force (IETF)
Request for Comments: 9305
Category: Standards Track
ISSN: 2070-1721

F. Maino, Ed.
Cisco
J. Lemon
P. Agarwal
Innovium
D. Lewis
M. Smith
Cisco

October 2022

Locator/ID Separation Protocol (LISP) Generic Protocol Extension

Abstract

This document describes extensions to the Locator/ID Separation Protocol (LISP) data plane, via changes to the LISP header, to support multiprotocol encapsulation and allow the introduction of new protocol capabilities.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9305.

Table of Contents

1. Introduction

The LISP data plane is defined in [RFC9300]. 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 IP packet payloads. Other protocols, most notably the Virtual eXtensible Local Area Network (VXLAN) protocol [RFC7348] (which defines a header format similar to LISP), are used to encapsulate Layer 2 (L2) protocols, such as Ethernet.

This document defines an extension for the LISP header, as defined in [RFC9300], to indicate the inner protocol, enabling the encapsulation of Ethernet, IP, or any other desired protocol, all the while ensuring compatibility with existing LISP deployments.

A flag in the LISP header -- the P-bit -- is used to signal the presence of the 8-bit 'Next Protocol' field. The 'Next Protocol' field, when present, uses 8 bits of the field that was allocated to the Echo-Noncing and Map-Versioning features in [RFC9300]. Those two features are no longer available when the P-bit is used. However, appropriate LISP Generic Protocol Extension (LISP-GPE) shim headers can be defined to specify capabilities that are equivalent to Echo-Noncing and/or Map-Versioning.

Since all of the reserved bits of the LISP data plane header have been allocated, LISP-GPE can also be used to extend the LISP data plane header by defining Next Protocol shim headers that implement new data plane functions not supported in the LISP header. For example, the use of the Group-Based Policy (GBP) header [VXLAN-LISP] or of the In situ Operations, Administration, and Maintenance (IOAM) header [VXLAN-GPE] with LISP-GPE can be considered an extension to add support in the data plane for GBP functionalities or IOAM metadata.

1.1. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Definitions of Terms

This document uses terms already defined in [RFC9300].

2. LISP Header without Protocol Extensions

As described in Section 1, the LISP header has no protocol identifier that indicates the type of payload being carried. Because of this, LISP is limited to carrying IP payloads.

The LISP header [RFC9300] contains a series of flags (some defined, some reserved), a 'Nonce/Map-Version' field, and an 'Instance ID/Locator-Status-Bits' field. The flags provide flexibility to define how the various fields are encoded. Notably, Flag bit 5 is the last reserved bit in the LISP header.

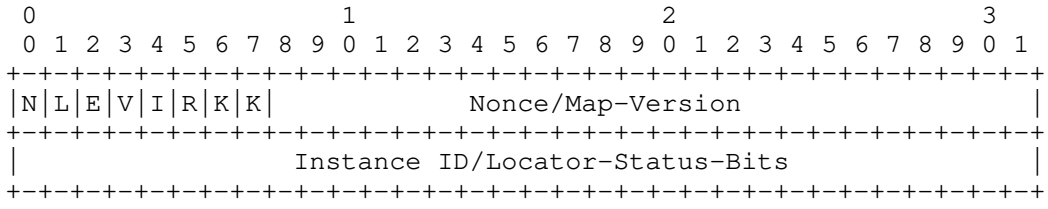

Figure 1: LISP Header

3. LISP Generic Protocol Extension (LISP-GPE)

This document defines two changes to the LISP header in order to support multiprotocol encapsulation: the introduction of the P-bit and the definition of a 'Next Protocol' field. This document specifies the protocol behavior when the P-bit is set to 1; no changes are introduced when the P-bit is set to 0. The LISP-GPE header is shown in Figure 2 and described below.

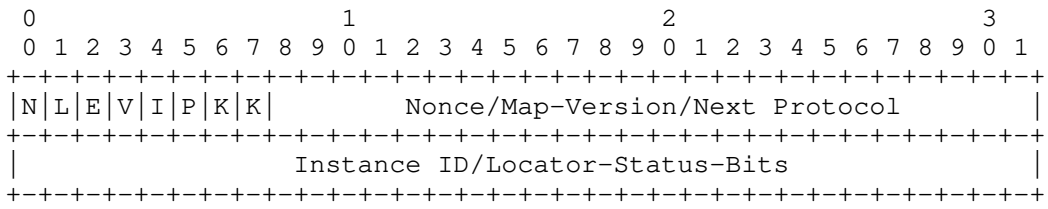

Figure 2: LISP-GPE Header

P-Bit: Flag bit 5 is defined as the Next Protocol bit. The P-bit is set to 1 to indicate the presence of the 8-bit 'Next Protocol' field.

If the P-bit is clear (0), the LISP header is bit-by-bit equivalent to the definition in [RFC9300].

When the P-bit is set to 1, bits N, E, and V, and bits 8-23 of the 'Nonce/Map-Version/Next Protocol' field MUST be set to zero on transmission and MUST be ignored on receipt. Features equivalent to those that were implemented with bits N, E, and V in [RFC9300], such as Echo-Noncing and Map-Versioning, can be implemented by defining appropriate LISP-GPE shim headers.

When the P-bit is set to 1, the LISP-GPE header is encoded as:

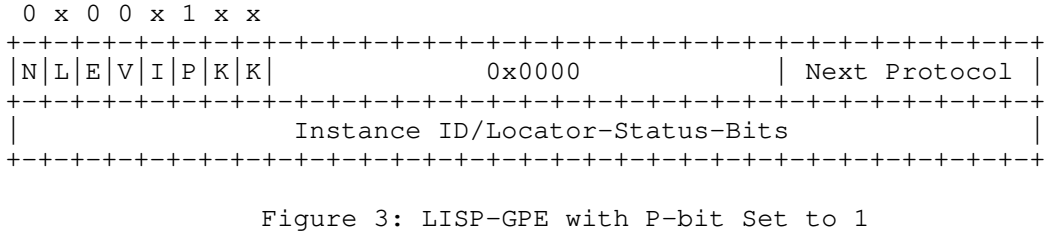

Figure 3: LISP-GPE with P-bit Set to 1

Next Protocol: When the P-bit is set to 1, the lower 8 bits of the first 32-bit word are used to carry a Next Protocol. This 'Next Protocol' field contains the protocol of the encapsulated payload packet.

This document defines the following Next Protocol values:

0x00: Reserved

0x01: IPv4

0x02: IPv6

0x03: Ethernet

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

0x05 to 0x7D: Unassigned

0x7E and 0x7F: Experimentation and testing

0x80 to 0xFD: Unassigned (shim headers)

0xFE, 0xFF: Experimentation and testing (shim headers)

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

Next Protocol values 0x7E, 0x7F, 0xFE, and 0xFF are assigned for experimentation and testing, as per [RFC3692].

Next Protocol values from 0x80 to 0xFD are assigned to protocols encoded as generic shim headers. All shim protocols MUST use the header structure in Figure 4, which includes a 'Next Protocol' field. When shim headers are used with other protocols identified by Next Protocol values from 0x00 to 0x7F, all the shim headers MUST come first.

Shim headers can be used to incrementally deploy new GPE features, keeping the processing of shim headers known to a given Tunnel Router (xTR) implementation in the 'fast' path (typically an Application-Specific Integrated Circuit (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:

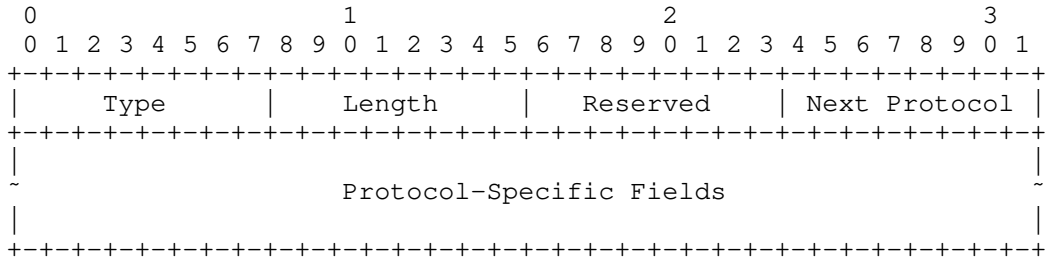

Figure 4: Shim Header

Where:

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

Length: This field indicates the length, in 4-octet units, of this protocol message, not including the first 4 octets.

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

Next Protocol: This 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 a UDP-based encapsulation protocol, to the UDP usage guidelines specified in [RFC8085]. The applicability of these guidelines is dependent on the underlay IP network and the nature of the encapsulated payload.

[RFC8085] outlines two applicability scenarios for UDP applications: 1) the general Internet and 2) a controlled environment. A controlled environment means a single administrative domain or

adjacent set of cooperating domains. A network in a controlled environment can be managed to operate under certain conditions, whereas, in the general Internet, this cannot be done. Hence, requirements for a tunnel protocol operating under a controlled environment can be less restrictive than the requirements of the general Internet.

The LISP-GPE scope of applicability is the same set of use cases covered by [RFC9300] for the LISP data plane protocol. The common property of these use cases is a large set of cooperating entities seeking to communicate over the public Internet or other large underlay IP infrastructures while keeping the addressing and topology of the cooperating entities separate from the underlay and Internet topology, routing, and addressing.

LISP-GPE is meant to be deployed in network environments operated by a single operator or adjacent set of cooperating network operators that fit with the definition of controlled environments in [RFC8085].

For the purpose of this document, a Traffic-Managed Controlled Environment (TMCE), outlined in [RFC8086], is defined as an IP network that is traffic-engineered and/or otherwise managed (e.g., via the use of traffic rate limiters) to avoid congestion. Significant portions of the text in this section are based on [RFC8086].

It is the responsibility of the network operators to ensure that the guidelines/requirements in this section are followed as applicable to their LISP-GPE deployments.

4.2. Congestion-Control Functionality

LISP-GPE does not provide congestion-control functionality and relies on the payload protocol traffic for congestion control. As such, LISP-GPE MUST be used with congestion-controlled traffic or within a network that is traffic managed to avoid congestion (TMCE). An operator of a traffic-managed network (TMCE) may avoid congestion by careful provisioning of their networks, rate limiting of user data traffic, and traffic engineering according to path capacity.

Keeping in mind the recommendation above, new encapsulated payloads, when registered with LISP-GPE, MUST be accompanied by a set of guidelines derived from Section 5 of [RFC9300]. Such new protocols should be designed for explicit congestion signals to propagate consistently from lower-layer protocols into IP. Then, the IP internetwork layer can act as a portability layer to carry congestion notifications from non-IP-aware congested nodes up to the transport layer (L4). By following the guidelines in [ENCAP-GUIDE], subnetwork designers can enable a Layer 2 protocol to participate in congestion control without dropping packets, via propagation of Explicit Congestion Notification (ECN) data [RFC3168] to receivers.

4.3. UDP Checksum

For IP payloads, Section 5.3 of [RFC9300] specifies how to handle UDP checksums, encouraging implementors to consider UDP checksum usage guidelines in Section 3.4 of [RFC8085] when it is desirable to protect UDP and LISP headers against corruption.

In order to protect the integrity of LISP-GPE headers, options, and payloads (for example, to avoid misdelivery of payloads to different tenant systems in the case of data corruption), the outer UDP checksum SHOULD be used with LISP-GPE when transported over IPv4. The UDP checksum provides a statistical guarantee that a payload was not corrupted in transit. These integrity checks are not strong from a coding or cryptographic perspective and are not designed to detect physical-layer errors or malicious modifications of the datagram (see Section 3.4 of [RFC8085]). In deployments where such a risk exists, an operator SHOULD use additional data integrity mechanisms, such as those offered by IPsec.

An operator MAY choose to disable a UDP checksum and use a zero checksum if LISP-GPE packet integrity is provided by other data integrity mechanisms, such as IPsec or additional checksums, or if one of the conditions in Section 4.3.1 (a, b, or c) is met.

4.3.1. UDP Zero Checksum Handling with IPv6

By default, a UDP checksum MUST be used when LISP-GPE is transported over IPv6. A tunnel endpoint MAY be configured for use with a zero UDP checksum if additional requirements described in this section are met.

When LISP-GPE is used over IPv6, a UDP checksum is used to protect IPv6 headers, UDP headers, and LISP-GPE headers and payloads from potential data corruption. As such, by default, LISP-GPE MUST use a UDP checksum when transported over IPv6. An operator MAY choose to configure to operate with a zero UDP checksum if operating in a traffic-managed controlled environment, as stated in Section 4.1, if one of the following conditions is met:

a. It is known that packet corruption is exceptionally unlikely (perhaps based on an operator's knowledge of equipment types in their underlay network), and the operator is willing to take the risk of undetected packet corruption.

b. It is determined through observational measurements (perhaps through historic or current traffic flows that use a non-zero checksum) that the level of packet corruption is tolerably low, and the operator is willing to take the risk of undetected corruption.

c. LISP-GPE payloads are carrying applications that are tolerant of misdelivered or corrupted packets (perhaps through higher-layer checksum validation and/or reliability through retransmission).

In addition, LISP-GPE tunnel implementations using a zero UDP checksum MUST meet the following requirements:

1. Use of a UDP checksum over IPv6 MUST be the default configuration for all LISP-GPE tunnels.

2. If LISP-GPE is used with a zero UDP checksum over IPv6, then such xTR implementations MUST meet all the requirements specified in Section 4 of [RFC6936] and requirement 1 specified in Section 5 of [RFC6936].

3. The Egress Tunnel Router (ETR) that decapsulates the packet SHOULD check that the source and destination IPv6 addresses are valid for the LISP-GPE tunnel that is configured to receive a zero UDP checksum and discard other packets that fail such checks.

4. The Ingress Tunnel Router (ITR) that encapsulates the packet MAY use different IPv6 source addresses for each LISP-GPE tunnel that uses zero UDP checksum mode in order to strengthen the decapsulator's check of the IPv6 source address (i.e., the same IPv6 source address is not to be used with more than one IPv6 destination address, irrespective of whether that destination address is a unicast or multicast address). When this is not possible, it is RECOMMENDED to use each source address for as few LISP-GPE tunnels that use a zero UDP checksum as is feasible.

5. Measures SHOULD be taken to prevent LISP-GPE traffic over IPv6 with a zero UDP checksum from escaping into the general Internet. Examples of such measures include employing packet filters at the Proxy Egress Tunnel Router (PETR) and/or keeping logical or physical separation of the LISP network from networks in the general Internet.

The above requirements do not change the requirements specified in [RFC6935], [RFC6936], or [RFC8200].

The requirement to check the source IPv6 address in addition to the destination IPv6 address, plus the recommendation against the reuse of source IPv6 addresses among LISP-GPE tunnels, collectively provide some mitigation for the absence of UDP checksum coverage of the IPv6 header. A traffic-managed controlled environment that satisfies at least one of the three conditions listed at the beginning of this section provides additional assurance.

4.4. DSCP, ECN, TTL, and 802.1Q

When encapsulating IP (including over Ethernet) packets, [RFC2983] provides guidance for mapping packets that contain Differentiated Services Code Point (DSCP) information between inner and outer IP headers. The Pipe model typically fits better with network virtualization. The DSCP value on the tunnel header is set based on a policy (which may be a fixed value, one based on the inner traffic class, or some other mechanism for grouping traffic). Some aspects of the Uniform model (which treats the inner and outer DSCP value as a single field by copying on ingress and egress) may also apply, such as the ability to remark the inner header on tunnel egress based on transit marking. However, the Uniform model is not conceptually consistent with network virtualization, which seeks to provide strong isolation between encapsulated traffic and the physical network.

[RFC6040] describes the mechanism for exposing ECN capabilities on IP tunnels and propagating congestion markers to the inner packets. This behavior MUST be followed for IP packets encapsulated in LISP-GPE.

Though the Uniform model or the Pipe model could be used for TTL (or Hop Limit in the case of IPv6) handling when tunneling IP packets, the Pipe model is more aligned with network virtualization. [RFC2003] provides guidance on handling TTL between inner IP headers and outer IP tunnels; this model is more aligned with the Pipe model and is recommended for use with LISP-GPE for network-virtualization applications.

When a LISP-GPE router performs Ethernet encapsulation, the inner 802.1Q 3-bit Priority Code Point ('PCP') field [IEEE.802.1Q_2014] MAY be mapped from the encapsulated frame to the DSCP codepoint of the Differentiated Services ('DS') field defined in [RFC2474].

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

Refer to Section 7 for considerations about the use of integrity protection for deployments, such as the public Internet, concerned with on-path attackers.

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 [RFC9300]) and so would misinterpret the other LISP header fields, possibly causing significant errors.

5.1. Detection of ETR Capabilities

The discovery of xTR capabilities to support LISP-GPE 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 has created a registry called "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 Track protocols.

Next Protocol	Description	Reference
0x00	Reserved	RFC 9305
0x01	IPv4	RFC 9305
0x02	IPv6	RFC 9305
0x03	Ethernet	RFC 9305
0x04	NSH	RFC 9305
0x05-0x7D	Unassigned	
0x7E-0x7F	Experimentation and testing	RFC 9305
0x80-0xFD	Unassigned (shim headers)	
0xFE-0xFF	Experimentation and testing (shim headers)	RFC 9305

Table 1

7. Security Considerations

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

As is the case for many encapsulations that use optional extensions, LISP-GPE is subject to on-path adversaries that can make arbitrary modifications to the packet (including the P-bit) to change or remove any part of the payload, or claim to encapsulate any protocol payload type. Typical integrity protection mechanisms (such as IPsec) SHOULD be used in combination with LISP-GPE by those protocol extensions that want to protect against on-path attackers.

With LISP-GPE, issues such as data plane spoofing, flooding, and traffic redirection may depend on the particular protocol payload encapsulated.

8. References

8.1. Normative References

[IEEE.802.1Q_2014]
IEEE, "IEEE Standard for Local and metropolitan area networks--Bridges and Bridged Networks", IEEE Std 802.1Q-2014, December 2014, <https://ieeexplore.ieee.org/document/6991462>.

[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>.

[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>.

[RFC9300] Farinacci, D., Fuller, V., Meyer, D., Lewis, D., and A. Cabellos, Ed., "The Locator/ID Separation Protocol (LISP)", RFC 9300, DOI 10.17487/RFC9300, October 2022, <https://www.rfc-editor.org/info/rfc9300>.

8.2. Informative References

[ENCAP-GUIDE] Briscoe, B. and J. Kaippallimalil, "Guidelines for Adding Congestion Notification to Protocols that Encapsulate IP", Work in Progress, Internet-Draft, draft-ietf-tsvwg-ecn-encap-guidelines-17, 11 July 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-tsvwg-ecn-encap-guidelines-17>.

[RFC2003] Perkins, C., "IP Encapsulation within IP", RFC 2003, DOI 10.17487/RFC2003, October 1996, <https://www.rfc-editor.org/info/rfc2003>.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <https://www.rfc-editor.org/info/rfc2474>.

[RFC2983] Black, D., "Differentiated Services and Tunnels", RFC 2983, DOI 10.17487/RFC2983, October 2000, <https://www.rfc-editor.org/info/rfc2983>.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <https://www.rfc-editor.org/info/rfc3168>.

[RFC3692] Narten, T., "Assigning Experimental and Testing Numbers Considered Useful", BCP 82, RFC 3692, DOI 10.17487/RFC3692, January 2004, <https://www.rfc-editor.org/info/rfc3692>.

[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>.

[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>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[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>.

[VXLAN-GPE] Brockners, F., Bhandari, S., Govindan, V., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A., Gafni, B., Lapukhov, P., and M. Spiegel, "VXLAN-GPE Encapsulation for In-situ OAM Data", Work in Progress, Internet-Draft, draft-brockners-ippm-ioam-vxlan-gpe-03, 4 November 2019, <https://datatracker.ietf.org/doc/html/draft-brockners-ippm-ioam-vxlan-gpe-03>.

[VXLAN-LISP] Lemon, J., Ed., Maino, F., Smith, M., and A. Isaac, "Group Policy Encoding with VXLAN-GPE and LISP-GPE", Work in Progress, Internet-Draft, draft-lemon-vxlan-lisp-gpe-gbp-02, 30 April 2019, <https://datatracker.ietf.org/doc/html/draft-lemon-vxlan-lisp-gpe-gbp-02>.

Acknowledgments

A special thank you goes to Dino Farinacci for his guidance and detailed review. Thanks to Tom Herbert for the suggestion to assign codepoints for experimentations and testing.

Contributors

The editor of this document would like to thank and recognize the following coauthors and contributors for their contributions. These coauthors and contributors provided invaluable concepts and content for this document's creation.

Darrel Lewis
Cisco Systems, Inc.

Fabio Maino
Cisco Systems, Inc.

Paul Quinn
Cisco Systems, Inc.

Michael Smith
Cisco Systems, Inc.

Navindra Yadav
Cisco Systems, Inc.

Larry Kreeger

Jennifer Lemon

Broadcom

Puneet Agarwal
Innovium

Authors' Addresses

Fabio Maino (editor)
Cisco Systems
San Jose, CA
United States of America
Email: fmaino@cisco.com

Jennifer Lemon
Email: jalemon@meus.us

Puneet Agarwal
Innovium
United States of America
Email: puneet@acm.org

Darrel Lewis
Cisco Systems
San Jose, CA
United States of America
Email: darlewis@cisco.com

Michael Smith
Cisco Systems
San Jose, CA 95134
United States of America
Email: michsmit@cisco.com