< draft-ietf-lisp-gpe-16.txt	draft-ietf-lisp-gpe-17.txt >

draft-ietf-lisp-gpe-16.txt

Internet Engineering Task Force F. Maino, Ed.
Internet-Draft Cisco
Intended status: Standards Track J. Lemon
Expires: December 5, 2020 Broadcom
P. Agarwal
Innovium
D. Lewis
M. Smith
Cisco
June 3, 2020

LISP Generic Protocol Extension
draft-ietf-lisp-gpe-16

Abstract

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

Status of This Memo

draft-ietf-lisp-gpe-17.txt

Internet Engineering Task Force F. Maino, Ed.
Internet-Draft Cisco
Intended status: Standards Track J. Lemon
Expires: January 8, 2021 Broadcom
P. Agarwal
Innovium
D. Lewis
M. Smith
Cisco
July 7, 2020

LISP Generic Protocol Extension
draft-ietf-lisp-gpe-17

Abstract

This document describes extensions 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

skipping to change at *page 1, line 38* | skipping to change at *page 1, line 38*

draft-ietf-lisp-gpe-16.txt

Copyright Notice

draft-ietf-lisp-gpe-17.txt

Copyright Notice

skipping to change at *page 2, line 19* | skipping to change at *page 2, line 19*

draft-ietf-lisp-gpe-16.txt

Table of Contents

1. Introduction

draft-ietf-lisp-gpe-17.txt

Table of Contents

1. Introduction

skipping to change at *page 3, line 10* | skipping to change at *page 3, line 10*

draft-ietf-lisp-gpe-16.txt

This document defines an extension for the LISP header, as defined in [I-D.ietf-lisp-rfc6830bis], 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, called 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 [I-D.ietf-lisp-rfc6830bis]. Those two features are no longer available when the P-bit is used. However, appropriate 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 implements new data plane functions not supported in the LISP header. For example, the use of the Group-Based Policy (GBP) header [I-D.lemon-vxlan-lisp-gpe-gbp] or of the In-situ Operations, Administration, and Maintenance (IOAM) header [I-D.brockners-ippm-ioam-vxlan-gpe] with LISP-GPE, can be considered

draft-ietf-lisp-gpe-17.txt

This document defines an extension for the LISP header, as defined in [I-D.ietf-lisp-rfc6830bis], 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, called 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 [I-D.ietf-lisp-rfc6830bis]. Those two features are no longer available when the P-bit is used. However, appropriate LISP-GPE (LISP Generic Protocol Extension) 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 implements new data plane functions not supported in the LISP header. For example, the use of the Group-Based Policy (GBP) header [I-D.lemon-vxlan-lisp-gpe-gbp] or of the In-situ Operations, Administration, and Maintenance (IOAM) header [I-D.brockners-ippm-ioam-vxlan-gpe] with LISP-GPE, can be considered an extension to add support in the Data-Plane for Group-Based Policy

skipping to change at *page 4, line 43*

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 [I-D.ietf-lisp-rfc6830bis].

When the P-bit is set to 1, bits N, E, V, and bits 8-23 of the 'Nonce/Map-Version/Next Protocol' field MUST be set to zero on transmission and ignored on receipt. Features equivalent to those that were implemented with bits N,E and V in [I-D.ietf-lisp-rfc6830bis], 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:

```
  0 x 0 0 x 1 x x
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |N|L|E|V|I|P|K|K|             0x0000            | Next Protocol |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

skipping to change at *page 5, line 21*

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:

0x01 : IPv4

0x02 : IPv6

0x03 : Ethernet

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

0x05 to 0x7D Unassigned

0x7E to 0x7F: Experimentation and testing

0x80 to 0xFD: Unassigned (shim headers)

0xFE to 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 and 0xFE, 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 a shim header is used with other protocols identified by next protocol values from 0x0 to 0x7D, the shim header MUST come before the further protocol, and the next header of the shim will indicate which protocol follows the shim header.

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:

```
 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
```

skipping to change at *page 6, line 28*

```
~                    Protocol Specific Fields                   ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Shim Header

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

an extension to add support in the Data-Plane for Group-Based Policy

skipping to change at *page 4, line 43*

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 [I-D.ietf-lisp-rfc6830bis].

When the P-bit is set to 1, bits N, E, 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 [I-D.ietf-lisp-rfc6830bis], 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:

```
  0 x 0 0 x 1 x x
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |N|L|E|V|I|P|K|K|             0x0000            | Next Protocol |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

skipping to change at *page 5, line 21*

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, 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 and 0xFE, 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 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:

```
 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
```

skipping to change at *page 6, line 30*

```
~                    Protocol Specific Fields                   ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Shim Header

Where:

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

Length: 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 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

skipping to change at *page 7, line 42*	**skipping to change at *page 7, line 45***
LISP-GPE does not natively 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.	LISP-GPE does not natively 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.
Encapsulated payloads may have Explicit Congestion Notification mechanisms that may or may not be mapped to the outer IP header ECN field. Such new encapsulated payolads, when registered with LISP-GPE, MUST be accompanied by a set of guidelines derived from [I-D.ietf-tsvwg-ecn-encap-guidelines] and [RFC6040].	Encapsulated payloads may have Explicit Congestion Notification mechanisms that may or may not be mapped to the outer IP header ECN field. Such new encapsulated payloads, when registered with LISP-GPE, MUST be accompanied by a set of guidelines derived from [I-D.ietf-tsvwg-ecn-encap-guidelines] and [RFC6040].
4.3. UDP Checksum	4.3. UDP Checksum
For IP payloads, section 5.3 of [I-D.ietf-lisp-rfc6830bis] 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.	For IP payloads, section 5.3 of [I-D.ietf-lisp-rfc6830bis] 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.
skipping to change at *page 8, line 22*	**skipping to change at *page 8, line 29***
layer errors or malicious modification 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 offered by IPSec.	layer errors or malicious modification 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 offered by IPSec.
An operator MAY choose to disable UDP checksum and use 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, c are met.	An operator MAY choose to disable UDP checksum and use 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, c are met.
	4.3.1. UDP Zero Checksum Handling with IPv6
By default, UDP checksum MUST be used when LISP-GPE is transported over IPv6. A tunnel endpoint MAY be configured for use with zero UDP checksum if additional requirements in Section 4.3.1 are met.	By default, UDP checksum MUST be used when LISP-GPE is transported over IPv6. A tunnel endpoint MAY be configured for use with zero UDP checksum if additional requirements in Section 4.3.1 are met.
4.3.1. UDP Zero Checksum Handling with IPv6	
When LISP-GPE is used over IPv6, UDP checksum is used to protect IPv6 headers, UDP headers and LISP-GPE headers and payload from potential data corruption. As such by default LISP-GPE MUST use UDP checksum when transported over IPv6. An operator MAY choose to configure to operate with zero UDP checksum if operating in a traffic managed controlled environment as stated in Section 4.1 if one of the following conditions are met:	When LISP-GPE is used over IPv6, UDP checksum is used to protect IPv6 headers, UDP headers and LISP-GPE headers and payload from potential data corruption. As such by default LISP-GPE MUST use UDP checksum when transported over IPv6. An operator MAY choose to configure to operate with zero UDP checksum if operating in a traffic managed controlled environment as stated in Section 4.1 if one of the following conditions are met:
a. It is known that the packet corruption is exceptionally unlikely (perhaps based on knowledge of equipment types in their underlay	a. It is known that the packet corruption is exceptionally unlikely (perhaps based on knowledge of equipment types in their underlay
skipping to change at *page 9, line 46*	**skipping to change at *page 10, line 5***
specified in [RFC6936].	specified in [RFC6936].
The requirement to check the source IPv6 address in addition to the destination IPv6 address, plus the recommendation against 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 three conditions listed at the beginning of this section provides additional assurance.	The requirement to check the source IPv6 address in addition to the destination IPv6 address, plus the recommendation against 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 three conditions listed at the beginning of this section provides additional assurance.
4.4. DSCP, ECN and TTL	4.4. DSCP, ECN, TTL, and 802.1Q
When encapsulating IP (including over Ethernet) packets [RFC2983] provides guidance for mapping DSCP between inner and outer IP headers. The Pipe model typically fits better 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	When encapsulating IP (including over Ethernet) packets [RFC2983] provides guidance for mapping DSCP between inner and outer IP headers. The Pipe model typically fits better 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
End of changes. 19 change blocks.	
25 lines changed or deleted	***27 lines changed or added***