IDR Working Group
Internet-Draft
Obsoletes: 5512 (if approved)
Intended status: Standards Track
Expires: April 2, 2020

K. Patel
Arrcus, Inc
G. Van de Velde
Nokia
S. Sangli
Juniper Networks, Inc
September 30, 2019

The BGP Tunnel Encapsulation Attribute
draft-ietf-idr-tunnel-encaps-14.txt

Abstract

RFC 5512 defines a BGP Path Attribute known as the "Tunnel Encapsulation Attribute". This attribute allows one to specify a set of tunnels. For each such tunnel, the attribute can provide the information needed to create the tunnel and the corresponding encapsulation header. The attribute can also provide information that aids in choosing whether a particular packet is to be sent through a particular tunnel. RFC 5512 states that the attribute is only carried in BGP UPDATEs that have the "Encapsulation Subsequent Address Family (Encapsulation SAFI)". This document deprecates the Encapsulation SAFI (which has never been used in production), and specifies semantics for the attribute when it is carried in UPDATEs of certain other SAFIs. This document adds support for additional tunnel types, and allows a remote tunnel endpoint address to be specified for each tunnel. This document also provides support for specifying fields of any inner or outer encapsulations that may be used by a particular tunnel.

This document obsoletes RFC 5512.

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 April 2, 2020.

Copyright Notice

Copyright (c) 2019 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 (https://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
1.1. Brief Summary of RFC 5512 4
1.2. Deficiencies in RFC 5512 4
1.3. Brief Summary of Changes from RFC 5512 5
1.4. Impact on RFC 5566 6
2. The Tunnel Encapsulation Attribute 6
3. Tunnel Encapsulation Attribute Sub-TLVs 8
3.1. The Tunnel Endpoint Sub-TLV 8
3.2. Encapsulation Sub-TLVs for Particular Tunnel Types . . . 10
3.2.1. VXLAN 10
3.2.2. VXLAN-GPE 12
3.2.3. NVGRE 13
3.2.4. L2TPv3 14
3.2.5. GRE 15
3.2.6. MPLS-in-GRE 15
3.2.7. IP-in-IP 16
3.3. Outer Encapsulation Sub-TLVs 16
3.3.1. IPv4 DS Field 16
3.3.2. UDP Destination Port 17
3.4. Sub-TLVs for Aiding Tunnel Selection 17
3.4.1. Protocol Type Sub-TLV 17
3.4.2. Color Sub-TLV 17
3.5. Embedded Label Handling Sub-TLV 18
3.6. MPLS Label Stack Sub-TLV 19
3.7. Prefix-SID Sub-TLV 20
4. Extended Communities Related to the Tunnel Encapsulation Attribute 21
4.1. Encapsulation Extended Community 21
4.2. Router's MAC Extended Community 23

4.3. Color Extended Community 23
5. Semantics and Usage of the Tunnel Encapsulation attribute . . 23
6. Routing Considerations 27
6.1. Impact on BGP Decision Process 27
6.2. Looping, Infinite Stacking, Etc. 27
7. Recursive Next Hop Resolution 28
8. Use of Virtual Network Identifiers and Embedded Labels when Imposing a Tunnel Encapsulation 28
8.1. Tunnel Types without a Virtual Network Identifier Field . 29
8.2. Tunnel Types with a Virtual Network Identifier Field . . 29
8.2.1. Unlabeled Address Families 30
8.2.2. Labeled Address Families 30
9. Applicability Restrictions 31
10. Scoping 32
11. Error Handling 32
12. IANA Considerations 34
12.1. Subsequent Address Family Identifiers 34
12.2. BGP Path Attributes 34
12.3. Extended Communities 35
12.4. BGP Tunnel Encapsulation Attribute Sub-TLVs 35
12.5. Tunnel Types 36
12.6. Flags Field of Vxlan Encapsulation sub-TLV 36
12.7. Flags Field of Vxlan-GPE Encapsulation sub-TLV 36
12.8. Flags Field of NVGRE Encapsulation sub-TLV 36
12.9. Embedded Label Handling sub-TLV 36
13. Security Considerations 37
14. Acknowledgments 38
15. Contributor Addresses 38
16. References 38
16.1. Normative References 38
16.2. Informative References 40
Authors' Addresses 41

1. Introduction

This document obsoletes RFC 5512. The deficiencies of RFC 5512, and a summary of the changes made, are discussed in Sections 1.1-1.3. The material from RFC 5512 that is retained has been incorporated into this document.

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.

1.1. Brief Summary of RFC 5512

[RFC5512] defines a BGP Path Attribute known as the Tunnel Encapsulation attribute. This attribute consists of one or more TLVs. Each TLV identifies a particular type of tunnel. Each TLV also contains one or more sub-TLVs. Some of the sub-TLVs, e.g., the "Encapsulation sub-TLV", contain information that may be used to form the encapsulation header for the specified tunnel type. Other sub-TLVs, e.g., the "color sub-TLV" and the "protocol sub-TLV", contain information that aids in determining whether particular packets should be sent through the tunnel that the TLV identifies.

[RFC5512] only allows the Tunnel Encapsulation attribute to be attached to BGP UPDATE messages of the Encapsulation Address Family. These UPDATE messages have an AFI (Address Family Identifier) of 1 or 2, and a SAFI of 7. In an UPDATE of the Encapsulation SAFI, the NLRI (Network Layer Reachability Information) is an address of the BGP speaker originating the UPDATE. Consider the following scenario:

- BGP speaker R1 has received and installed UPDATE U;
- UPDATE U's SAFI is the Encapsulation SAFI;
- UPDATE U has the address R2 as its NLRI;
- UPDATE U has a Tunnel Encapsulation attribute.
- R1 has a packet, P, to transmit to destination D;
- R1's best path to D is a BGP route that has R2 as its next hop;

In this scenario, when R1 transmits packet P, it should transmit it to R2 through one of the tunnels specified in U's Tunnel Encapsulation attribute. The IP address of the tunnel egress endpoint of each such tunnel is R2. Packet P is known as the tunnel's "payload".

1.2. Deficiencies in RFC 5512

While the ability to specify tunnel information in a BGP UPDATE is useful, the procedures of [RFC5512] have certain limitations:

- The requirement to use the "Encapsulation SAFI" presents an unfortunate operational cost, as each BGP session that may need to carry tunnel encapsulation information needs to be reconfigured to support the Encapsulation SAFI. The Encapsulation SAFI has never been used, and this requirement has served only to discourage the use of the Tunnel Encapsulation attribute.

- There is no way to use the Tunnel Encapsulation attribute to specify the tunnel egress endpoint address of a given tunnel; [RFC5512] assumes that the tunnel egress endpoint of each tunnel is specified as the NLRI of an UPDATE of the Encapsulation-SAFI.

- If the respective best paths to two different address prefixes have the same next hop, [RFC5512] does not provide a straightforward method to associate each prefix with a different tunnel.

- If a particular tunnel type requires an outer IP or UDP encapsulation, there is no way to signal the values of any of the fields of the outer encapsulation.

- In [RFC5512]'s specification of the sub-TLVs, each sub-TLV has one-octet length field. In some cases, a two-octet length field may be needed.

1.3. Brief Summary of Changes from RFC 5512

In this document we address these deficiencies by:

- Deprecating the Encapsulation SAFI.

- Defining a new "Tunnel Endpoint sub-TLV" that can be included in any of the TLVs contained in the Tunnel Encapsulation attribute. This sub-TLV can be used to specify the remote endpoint address of a particular tunnel.

- Allowing the Tunnel Encapsulation attribute to be carried by BGP UPDATEs of additional AFI/SAFIs. Appropriate semantics are provided for this way of using the attribute.

- Defining a number of new sub-TLVs that provide additional information that is useful when forming the encapsulation header used to send a packet through a particular tunnel.

- Defining the sub-TLV type field so that a sub-TLV whose type is in the range from 0 to 127 inclusive has a one-octet length field, but a sub-TLV whose type is in the range from 128 to 255 inclusive has a two-octet length field.

One of the sub-TLVs defined in [RFC5512] is the "Encapsulation sub-TLV". For a given tunnel, the encapsulation sub-TLV specifies some of the information needed to construct the encapsulation header used when sending packets through that tunnel. This document defines encapsulation sub-TLVs for a number of tunnel types not discussed in [RFC5512]: VXLAN (Virtual Extensible Local Area Network, [RFC7348]),

VXLAN-GPE (Generic Protocol Extension for VXLAN, [I-D.ietf-nvo3-vxlan-gpe]), NVGRE (Network Virtualization Using Generic Routing Encapsulation [RFC7637]), and MPLS-in-GRE (MPLS in Generic Routing Encapsulation [RFC2784], [RFC2890], [RFC4023]). MPLS-in-UDP [RFC7510] is also supported, but an Encapsulation sub-TLV for it is not needed.

Some of the encapsulations mentioned in the previous paragraph need to be further encapsulated inside UDP and/or IP. [RFC5512] provides no way to specify that certain information is to appear in these outer IP and/or UDP encapsulations. This document provides a framework for including such information in the TLVs of the Tunnel Encapsulation attribute.

When the Tunnel Encapsulation attribute is attached to a BGP UPDATE whose AFI/SAFI identifies one of the labeled address families, it is not always obvious whether the label embedded in the NLRI is to appear somewhere in the tunnel encapsulation header (and if so, where), or whether it is to appear in the payload, or whether it can be omitted altogether. This is especially true if the tunnel encapsulation header itself contains a "virtual network identifier". This document provides a mechanism that allows one to signal (by using sub-TLVs of the Tunnel Encapsulation attribute) how one wants to use the embedded label when the tunnel encapsulation has its own virtual network identifier field.

[RFC5512] defines a Tunnel Encapsulation Extended Community, that can be used instead of the Tunnel Encapsulation attribute under certain circumstances. This document addresses the issue of how to handle a BGP UPDATE that carries both a Tunnel Encapsulation attribute and one or more Tunnel Encapsulation Extended Communities.

1.4. Impact on RFC 5566

[RFC5566] uses the mechanisms defined in [RFC5512]. While this document obsoletes [RFC5512], it does not address the issue of how to use the mechanisms of [RFC5566] without also using the Encapsulation SAFI. Those issues are considered to be outside the scope of this document.

2. The Tunnel Encapsulation Attribute

The Tunnel Encapsulation attribute is an optional transitive BGP Path attribute. IANA has assigned the value 23 as the type code of the attribute. The attribute is composed of a set of Type-Length-Value (TLV) encodings. Each TLV contains information corresponding to a particular tunnel type. A TLV is structured as shown in Figure 1:

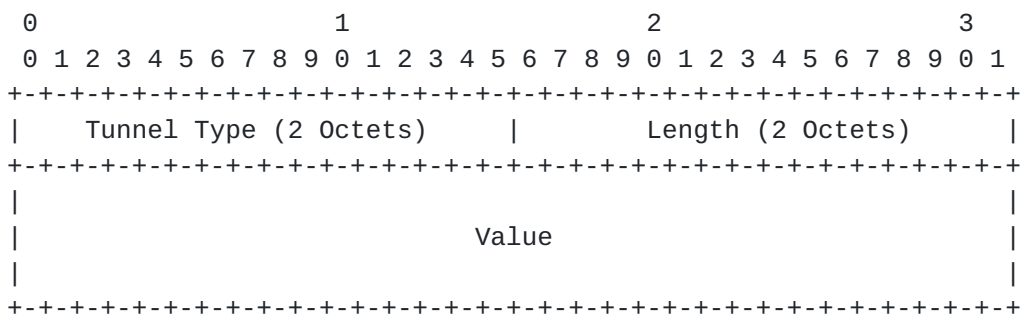


Figure 1: Tunnel Encapsulation TLV Value Field

- Tunnel Type (2 octets): identifies a type of tunnel. The field contains values from the IANA Registry "BGP Tunnel Encapsulation Attribute Tunnel Types".

  Note that for tunnel types whose names are of the form "X-in-Y", e.g., "MPLS-in-GRE", only packets of the specified payload type "X" are to be carried through the tunnel of type "Y". This is the equivalent of specifying a tunnel type "Y" and including in its TLV a Protocol Type sub-TLV (see Section 3.4.1) specifying protocol "X". If the tunnel type is "X-in-Y", it is unnecessary, though harmless, to include a Protocol Type sub-TLV specifying "X".

- Length (2 octets): the total number of octets of the value field.

- Value (variable): comprised of multiple sub-TLVs.

Each sub-TLV consists of three fields: a 1-octet type, a 1-octet or 2-octet length field (depending on the type), and zero or more octets of value. A sub-TLV is structured as shown in Figure 2:

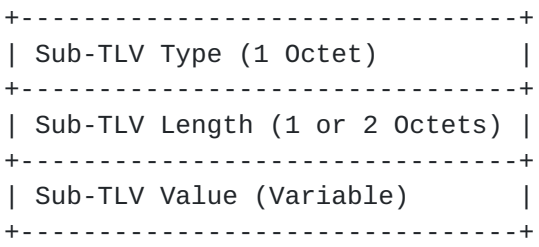


Table 1: Tunnel Encapsulation Sub-TLV Format

- Sub-TLV Type (1 octet): each sub-TLV type defines a certain property about the tunnel TLV that contains this sub-TLV.

- Sub-TLV Length (1 or 2 octets): the total number of octets of the sub-TLV value field. The Sub-TLV Length field contains 1 octet if

the Sub-TLV Type field contains a value in the range from 0-127. The Sub-TLV Length field contains two octets if the Sub-TLV Type field contains a value in the range from 128-255.

- Sub-TLV Value (variable): encodings of the value field depend on the sub-TLV type as enumerated above. The following sub-sections define the encoding in detail.

3. Tunnel Encapsulation Attribute Sub-TLVs

In this section, we specify a number of sub-TLVs. These sub-TLVs can be included in a TLV of the Tunnel Encapsulation attribute.

3.1. The Tunnel Endpoint Sub-TLV

The Tunnel Endpoint sub-TLV specifies the address of the endpoint of the tunnel, that is, the address of the router that will decapsulate the payload. It is a sub-TLV whose value field contains three sub-fields:

1. a four-octet Autonomous System (AS) number sub-field

2. a two-octet Address Family sub-field

3. an address sub-field, whose length depends upon the Address Family.

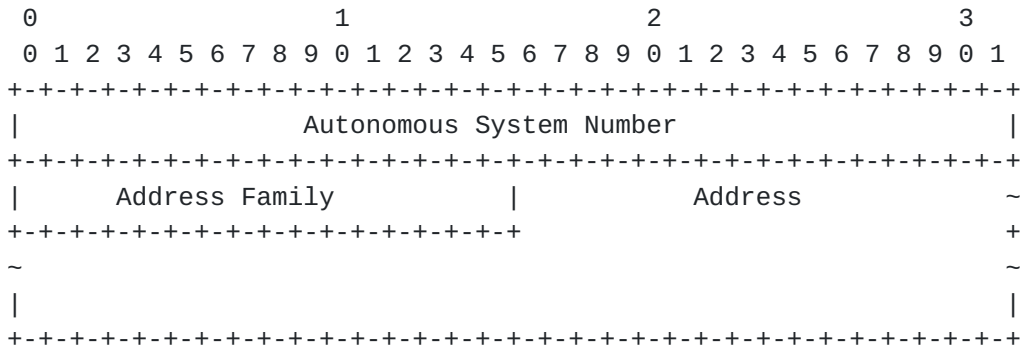


Figure 2: Tunnel Endpoint Sub-TLV Value Field

The Address Family subfield contains a value from IANA's "Address Family Numbers" registry. In this document, we assume that the Address Family is either IPv4 or IPv6; use of other address families is outside the scope of this document.

If the Address Family subfield contains the value for IPv4, the address subfield must contain an IPv4 address (a /32 IPv4 prefix).

In this case, the length field of Tunnel Endpoint sub-TLV must contain the value 10 (0xa).

If the Address Family subfield contains the value for IPv6, the address sub-field must contain an IPv6 address (a /128 IPv6 prefix). In this case, the length field of Tunnel Endpoint sub-TLV must contain the value 22 (0x16). IPv6 link local addresses are not valid values of the IP address field.

In a given BGP UPDATE, the address family (IPv4 or IPv6) of a Tunnel Endpoint sub-TLV is independent of the address family of the UPDATE itself. For example, an UPDATE whose NLRI is an IPv4 address may have a Tunnel Encapsulation attribute containing Tunnel Endpoint sub-TLVs that contain IPv6 addresses. Also, different tunnels represented in the Tunnel Encapsulation attribute may have Tunnel Endpoints of different address families.

A two-octet AS number can be carried in the AS number field by setting the two high order octets to zero, and carrying the number in the two low order octets of the field.

The AS number in the sub-TLV MUST be the number of the AS to which the IP address in the sub-TLV belongs.

There is one special case: the Tunnel Endpoint sub-TLV MAY have a value field whose Address Family subfield contains 0. This means that the tunnel's egress endpoint is the UPDATE's BGP next hop. If the Address Family subfield contains 0, the Address subfield is omitted, and the Autonomous System number field is set to 0.

If any of the following conditions hold, the Tunnel Endpoint sub-TLV is considered to be "malformed":

- The sub-TLV contains the value for IPv4 in its Address Family subfield, but the length of the sub-TLV's value field is other than 10 (0xa).

- The sub-TLV contains the value for IPv6 in its Address Family subfield, but the length of the sub-TLV's value field is other than 22 (0x16).

- The sub-TLV contains the value zero in its Address Family field, but the length of the sub-TLV's value field is other than 6, or the Autonomous System subfield is not set to zero.

- The IP address in the sub-TLV's address subfield is not a valid IP address (e.g., it's an IPv4 broadcast address).

- It can be determined that the IP address in the sub-TLV's address subfield does not belong to the non-zero AS whose number is in the its Autonomous System subfield. (See section Section 13 for discussion of one way to determine this.)

If the Tunnel Endpoint sub-TLV is malformed, the TLV containing it is also considered to be malformed, and the entire TLV MUST be ignored. However, the Tunnel Encapsulation attribute MUST NOT be considered to be malformed in this case; other TLVs in the attribute MUST be processed (if they can be parsed correctly).

When redistributing a route that is carrying a Tunnel Encapsulation attribute containing a TLV that itself contains a malformed Tunnel Endpoint sub-TLV, the TLV MUST be removed from the attribute before redistribution.

See Section 11 for further discussion of how to handle errors that are encountered when parsing the Tunnel Encapsulation attribute.

If the Tunnel Endpoint sub-TLV contains an IPv4 or IPv6 address that is valid but not reachable, the sub-TLV is NOT considered to be malformed.

3.2. Encapsulation Sub-TLVs for Particular Tunnel Types

This section defines Tunnel Encapsulation sub-TLVs for the following tunnel types: VXLAN ([RFC7348]), VXLAN-GPE ([I-D.ietf-nvo3-vxlan-gpe]), NVGRE ([RFC7637]), MPLS-in-GRE ([RFC2784], [RFC2890], [RFC4023]), L2TPv3 ([RFC3931]), and GRE ([RFC2784], [RFC2890], [RFC4023]).

Rules for forming the encapsulation based on the information in a given TLV are given in Sections 5 and 8.

There are also tunnel types for which it is not necessary to define an Encapsulation sub-TLV, because there are no fields in the encapsulation header whose values need to be signaled from the tunnel egress endpoint.

3.2.1. VXLAN

This document defines an encapsulation sub-TLV for VXLAN tunnels. When the tunnel type is VXLAN, the following is the structure of the value field in the encapsulation sub-TLV:

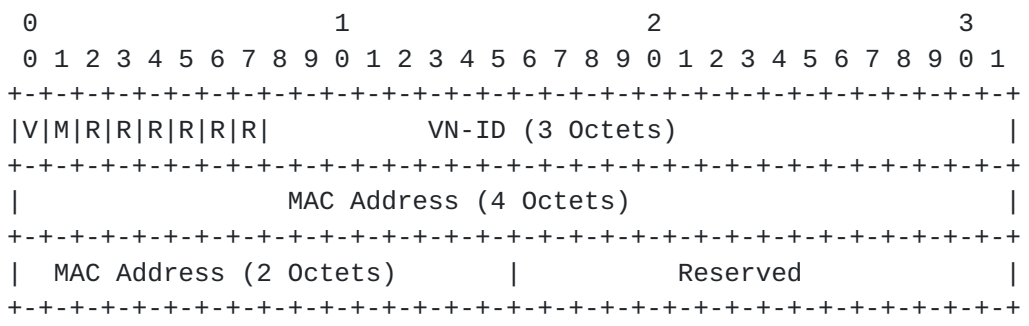


Figure 3: VXLAN Encapsulation Sub-TLV

V: This bit is set to 1 to indicate that a "valid" VN-ID (Virtual Network Identifier) is present in the encapsulation sub-TLV. Please see Section 8.

M: This bit is set to 1 to indicate that a valid MAC Address is present in the encapsulation sub-TLV.

R: The remaining bits in the 8-bit flags field are reserved for further use. They MUST always be set to 0 by the originator of the sub-TLV. Intermediate routers MUST propagate them without modification. Any receiving routers MUST ignore these bits upon a receipt of the sub-TLV.

VN-ID: If the V bit is set, the VN-id field contains a 3 octet VN-ID value. If the V bit is not set, the VN-id field MUST be set to zero.

MAC Address: If the M bit is set, this field contains a 6 octet Ethernet MAC address. If the M bit is not set, this field MUST be set to all zeroes.

When forming the VXLAN encapsulation header:

- The values of the V, M, and R bits are NOT copied into the flags field of the VXLAN header. The flags field of the VXLAN header is set as per [RFC7348].

- If the M bit is set, the MAC Address is copied into the Inner Destination MAC Address field of the Inner Ethernet Header (see section 5 of [RFC7348]).

  If the M bit is not set, and the payload being sent through the VXLAN tunnel is an ethernet frame, the Destination MAC Address field of the Inner Ethernet Header is just the Destination MAC Address field of the payload's ethernet header.

If the M bit is not set, and the payload being sent through the VXLAN tunnel is an IP or MPLS packet, the Inner Destination MAC address field is set to a configured value; if there is no configured value, the VXLAN tunnel cannot be used.

o See Section 8 to see how the VNI field of the VXLAN encapsulation header is set.

Note that in order to send an IP packet or an MPLS packet through a VXLAN tunnel, the packet must first be encapsulated in an ethernet header, which becomes the "inner ethernet header" described in [RFC7348]. The VXLAN Encapsulation sub-TLV may contain information (e.g.,the MAC address) that is used to form this ethernet header.

3.2.2. VXLAN-GPE

This document defines an encapsulation sub-TLV for VXLAN tunnels. When the tunnel type is VXLAN-GPE, the following is the structure of the value field in the encapsulation sub-TLV:

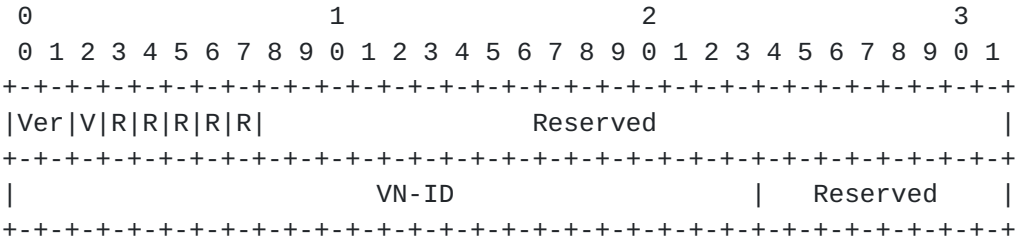


Figure 4: VXLAN GPE Encapsulation Sub-TLV

V: This bit is set to 1 to indicate that a "valid" VN-ID is present in the encapsulation sub-TLV. Please see Section 8.

R: The bits designated "R" above are reserved for future use. They MUST always be set to 0 by the originator of the sub-TLV. Intermediate routers MUST propagate them without modification. Any receiving routers MUST ignore these bits upon a receipt of the sub-TLV.

Version (Ver): Indicates VXLAN GPE protocol version. (See the "Version Bits" section of [I-D.ietf-nvo3-vxlan-gpe].) If the indicated version is not supported, the TLV that contains this Encapsulation sub-TLV MUST be treated as specifying an unsupported tunnel type. The value of this field will be copied into the corresponding field of the VXLAN encapsulation header.

VN-ID: If the V bit is set, this field contains a 3 octet VN-ID value. If the V bit is not set, this field MUST be set to zero.

When forming the VXLAN-GPE encapsulation header:

- The values of the V and R bits are NOT copied into the flags field of the VXLAN-GPE header. However, the values of the Ver bits are copied into the VXLAN-GPE header. Other bits in the flags field of the VXLAN-GPE header are set as per [I-D.ietf-nvo3-vxlan-gpe].

- See Section 8 to see how the VNI field of the VXLAN-GPE encapsulation header is set.

3.2.3. NVGRE

This document defines an encapsulation sub-TLV for NVGRE tunnels. When the tunnel type is NVGRE, the following is the structure of the value field in the encapsulation sub-TLV:

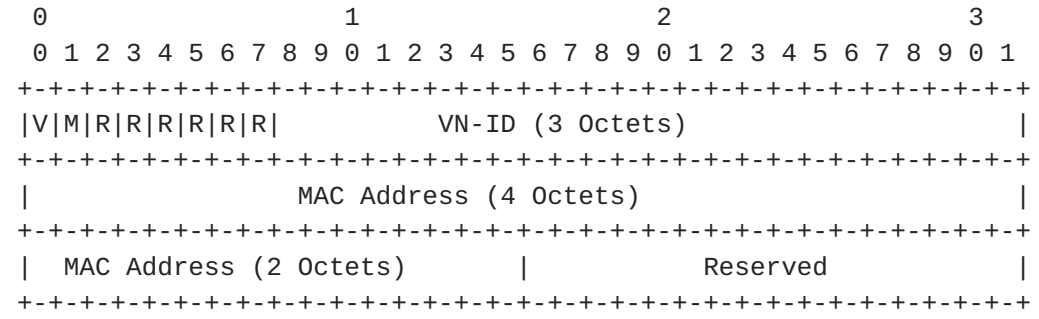


Figure 5: NVGRE Encapsulation Sub-TLV

V: This bit is set to 1 to indicate that a "valid" VN-ID is present in the encapsulation sub-TLV. Please see Section 8.

M: This bit is set to 1 to indicate that a valid MAC Address is present in the encapsulation sub-TLV.

R: The remaining bits in the 8-bit flags field are reserved for further use. They MUST always be set to 0 by the originator of the sub-TLV. Intermediate routers MUST propagate them without modification. Any receiving routers MUST ignore these bits upon a receipt of the sub-TLV.

VN-ID: If the V bit is set, the VN-id field contains a 3 octet VN-ID value. If the V bit is not set, the VN-id field MUST be set to zero.

MAC Address: If the M bit is set, this field contains a 6 octet Ethernet MAC address. If the M bit is not set, this field MUST be set to all zeroes.

When forming the NVGRE encapsulation header:

- The values of the V, M, and R bits are NOT copied into the flags field of the NVGRE header. The flags field of the VXLAN header is set as per [RFC7637].

- If the M bit is set, the MAC Address is copied into the Inner Destination MAC Address field of the Inner Ethernet Header (see section 3.2 of [RFC7637]).

  If the M bit is not set, and the payload being sent through the NVGRE tunnel is an ethernet frame, the Destination MAC Address field of the Inner Ethernet Header is just the Destination MAC Address field of the payload's ethernet header.

  If the M bit is not set, and the payload being sent through the NVGRE tunnel is an IP or MPLS packet, the Inner Destination MAC address field is set to a configured value; if there is no configured value, the NVGRE tunnel cannot be used.

- See Section 8 to see how the VSID (Virtual Subnet Identifier) field of the NVGRE encapsulation header is set.

3.2.4. L2TPv3

When the tunnel type of the TLV is L2TPv3 over IP, the following is the structure of the value field of the encapsulation sub-TLV:

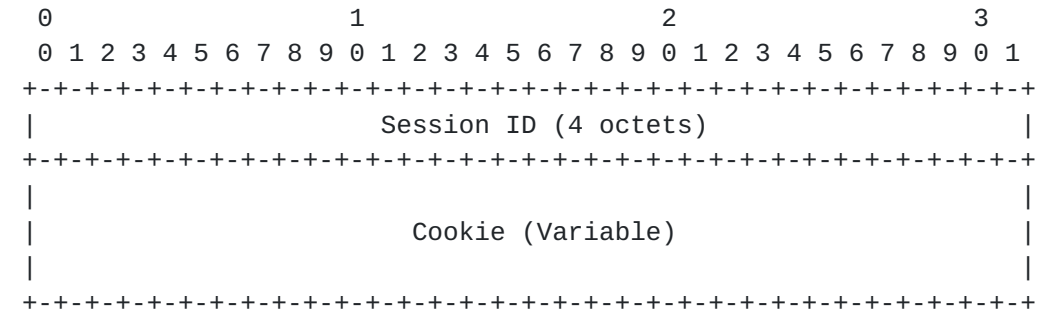


Figure 6: L2TPv3 Encapsulation Sub-TLV

Session ID: a non-zero 4-octet value locally assigned by the advertising router that serves as a lookup key in the incoming packet's context.

Cookie: an optional, variable length (encoded in octets -- 0 to 8 octets) value used by L2TPv3 to check the association of a received data message with the session identified by the Session

ID. Generation and usage of the cookie value is as specified in [RFC3931].

The length of the cookie is not encoded explicitly, but can be calculated as (sub-TLV length - 4).

3.2.5. GRE

When the tunnel type of the TLV is GRE, the following is the structure of the value field of the encapsulation sub-TLV:

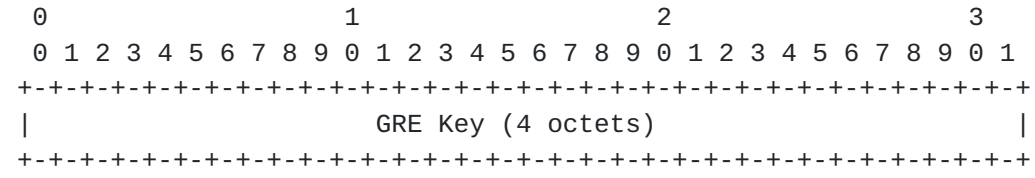


Figure 7: GRE Encapsulation Sub-TLV

GRE Key: 4-octet field [RFC2890] that is generated by the advertising router. The actual method by which the key is obtained is beyond the scope of this document. The key is inserted into the GRE encapsulation header of the payload packets sent by ingress routers to the advertising router. It is intended to be used for identifying extra context information about the received payload.

Note that the key is optional. Unless a key value is being advertised, the GRE encapsulation sub-TLV MUST NOT be present.

3.2.6. MPLS-in-GRE

When the tunnel type is MPLS-in-GRE, the following is the structure of the value field in an optional encapsulation sub-TLV:

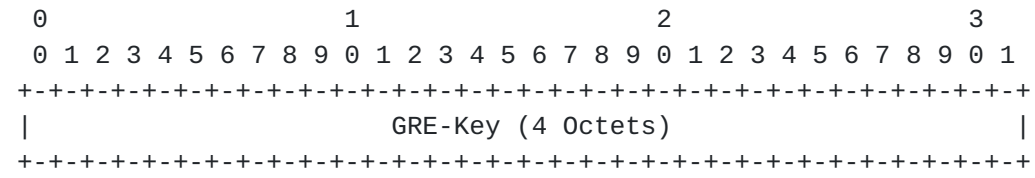


Figure 8: MPLS-in-GRE Encapsulation Sub-TLV

GRE-Key: 4-octet field [RFC2890] that is generated by the advertising router. The actual method by which the key is obtained is beyond the scope of this document. The key is inserted into the GRE encapsulation header of the payload packets sent by ingress routers to the advertising router. It is intended

to be used for identifying extra context information about the received payload. Note that the key is optional. Unless a key value is being advertised, the MPLS-in-GRE encapsulation sub-TLV MUST NOT be present.

Note that the GRE tunnel type defined in Section 3.2.5 can be used instead of the MPLS-in-GRE tunnel type when it is necessary to encapsulate MPLS in GRE. Including a TLV of the MPLS-in-GRE tunnel type is equivalent to including a TLV of the GRE tunnel type that also includes a Protocol Type sub-TLV (Section 3.4.1) specifying MPLS as the protocol to be encapsulated. That is, if a TLV specifies MPLS-in-GRE or if it includes a Protocol Type sub-TLV specifying MPLS, the GRE tunnel advertised in that TLV MUST NOT be used for carrying IP packets.

While it is not really necessary to have both the GRE and MPLS-in-GRE tunnel types, both are included for reasons of backwards compatibility.

3.2.7. IP-in-IP

When the tunnel type of the TLV is IP-in-IP, it does not have Virtual Network Identifier. See for Section 8.1 Embedded Label handling on IP-in-IP tunnels.

3.3. Outer Encapsulation Sub-TLVs

The Encapsulation sub-TLV for a particular tunnel type allows one to specify the values that are to be placed in certain fields of the encapsulation header for that tunnel type. However, some tunnel types require an outer IP encapsulation, and some also require an outer UDP encapsulation. The Encapsulation sub-TLV for a given tunnel type does not usually provide a way to specify values for fields of the outer IP and/or UDP encapsulations. If it is necessary to specify values for fields of the outer encapsulation, additional sub-TLVs must be used. This document defines two such sub-TLVs.

If an outer encapsulation sub-TLV occurs in a TLV for a tunnel type that does not use the corresponding outer encapsulation, the sub-TLV is treated as if it were an unknown type of sub-TLV.

3.3.1. IPv4 DS Field

Most of the tunnel types that can be specified in the Tunnel Encapsulation attribute require an outer IP encapsulation. The IPv4 Differentiated Services (DS) Field sub-TLV can be carried in the TLV of any such tunnel type. It specifies the setting of the one-octet

Differentiated Services field in the outer IP encapsulation (see [RFC2474]). The value field is always a single octet.

3.3.2. UDP Destination Port

Some of the tunnel types that can be specified in the Tunnel Encapsulation attribute require an outer UDP encapsulation. Generally there is a standard UDP Destination Port value for a particular tunnel type. However, sometimes it is useful to be able to use a non-standard UDP destination port. If a particular tunnel type requires an outer UDP encapsulation, and it is desired to use a UDP destination port other than the standard one, the port to be used can be specified by including a UDP Destination Port sub-TLV. The value field of this sub-TLV is always a two-octet field, containing the port value.

3.4. Sub-TLVs for Aiding Tunnel Selection

3.4.1. Protocol Type Sub-TLV

The protocol type sub-TLV MAY be included in a given TLV to indicate the type of the payload packets that may be encapsulated with the tunnel parameters that are being signaled in the TLV. The value field of the sub-TLV contains a 2-octet value from IANA's ethertype registry [Ethertypes].

For example, if we want to use three L2TPv3 sessions, one carrying IPv4 packets, one carrying IPv6 packets, and one carrying MPLS packets, the egress router will include three TLVs of L2TPv3 encapsulation type, each specifying a different Session ID and a different payload type. The protocol type sub-TLV for these will be IPv4 (protocol type = 0x0800), IPv6 (protocol type = 0x86dd), and MPLS (protocol type = 0x8847), respectively. This informs the ingress routers of the appropriate encapsulation information to use with each of the given protocol types. Insertion of the specified Session ID at the ingress routers allows the egress to process the incoming packets correctly, according to their protocol type.

3.4.2. Color Sub-TLV

The color sub-TLV MAY be encoded as a way to "color" the corresponding tunnel TLV. The value field of the sub-TLV is eight octets long, and consists of a Color Extended Community, as defined in Section 4.3. For the use of this sub-TLV and Extended Community, please see Section 7.

Note that the high-order octet of this sub-TLV's value field MUST be set to 3, and the next octet MUST be set to 0x0b. (Otherwise the value field is not identical to a Color Extended Community.)

If a Color sub-TLV is not of the proper length, or the first two octets of its value field are not 0x030b, the sub-TLV should be treated as if it were an unrecognized sub-TLV (see Section 11).

3.5. Embedded Label Handling Sub-TLV

Certain BGP address families (corresponding to particular AFI/SAFI pairs, e.g., 1/4, 2/4, 1/128, 2/128) have MPLS labels embedded in their NLRIs. We will use the term "embedded label" to refer to the MPLS label that is embedded in an NLRI, and the term "labeled address family" to refer to any AFI/SAFI that has embedded labels.

Some of the tunnel types (e.g., VXLAN, VXLAN-GPE, and NVGRE) that can be specified in the Tunnel Encapsulation attribute have an encapsulation header containing "Virtual Network" identifier of some sort. The Encapsulation sub-TLVs for these tunnel types may optionally specify a value for the virtual network identifier.

Suppose a Tunnel Encapsulation attribute is attached to an UPDATE of an embedded address family, and it is decided to use a particular tunnel (specified in one of the attribute's TLVs) for transmitting a packet that is being forwarded according to that UPDATE. When forming the encapsulation header for that packet, different deployment scenarios require different handling of the embedded label and/or the virtual network identifier. The Embedded Label Handling sub-TLV can be used to control the placement of the embedded label and/or the virtual network identifier in the encapsulation.

The Embedded Label Handling sub-TLV may be included in any TLV of the Tunnel Encapsulation attribute. If the Tunnel Encapsulation attribute is attached to an UPDATE of a non-labeled address family, the sub-TLV is treated as a no-op. If the sub-TLV is contained in a TLV whose tunnel type does not have a virtual network identifier in its encapsulation header, the sub-TLV is treated as a no-op. In those cases where the sub-TLV is treated as a no-op, it SHOULD NOT be stripped from the TLV before the UPDATE is forwarded.

The sub-TLV's Length field always contains the value 1, and its value field consists of a single octet. The following values are defined:

1: The payload will be an MPLS packet with the embedded label at the

   top of its label stack.

2: The embedded label is not carried in the payload, but is carried either in the virtual network identifier field of the encapsulation header, or else is ignored entirely.

Please see Section 8 for the details of how this sub-TLV is used when it is carried by an UPDATE of a labeled address family.

3.6. MPLS Label Stack Sub-TLV

This sub-TLV allows an MPLS label stack ([RFC3032]) to be associated with a particular tunnel.

The value field of this sub-TLV is a sequence of MPLS label stack entries. The first entry in the sequence is the "topmost" label, the final entry in the sequence is the "bottommost" label. When this label stack is pushed onto a packet, this ordering MUST be preserved.

Each label stack entry has the following format:

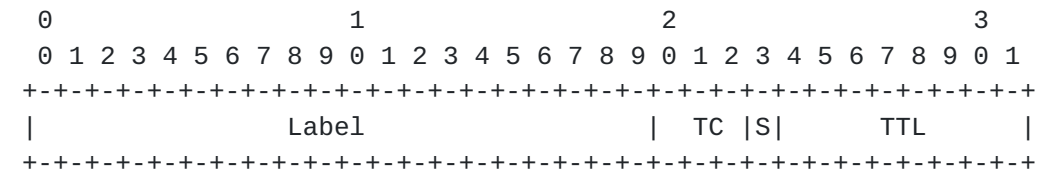


Figure 9: MPLS Label Stack Sub-TLV

If a packet is to be sent through the tunnel identified in a particular TLV, and if that TLV contains an MPLS Label Stack sub-TLV, then the label stack appearing in the sub-TLV MUST be pushed onto the packet. This label stack MUST be pushed onto the packet before any other labels are pushed onto the packet.

In particular, if the Tunnel Encapsulation attribute is attached to a BGP UPDATE of a labeled address family, the contents of the MPLS Label Stack sub-TLV MUST be pushed onto the packet before the label embedded in the NLRI is pushed onto the packet.

If the MPLS label stack sub-TLV is included in a TLV identifying a tunnel type that uses virtual network identifiers (see Section 8), the contents of the MPLS label stack sub-TLV MUST be pushed onto the packet before the procedures of Section 8 are applied.

The number of label stack entries in the sub-TLV MUST be determined from the sub-TLV length field. Thus it is not necessary to set the S bit in any of the label stack entries of the sub-TLV, and the setting of the S bit is ignored when parsing the sub-TLV. When the label stack entries are pushed onto a packet that already has a label

stack, the S bits of all the entries MUST be cleared. When the label stack entries are pushed onto a packet that does not already have a label stack, the S bit of the bottommost label stack entry MUST be set, and the S bit of all the other label stack entries MUST be cleared.

By default, the TC (Traffic Class) field ([RFC3032], [RFC5462]) of each label stack entry is set to 0. This may of course be changed by policy at the originator of the sub-TLV. When pushing the label stack onto a packet, the TC of the label stack entries is preserved by default. However, local policy at the router that is pushing on the stack MAY cause modification of the TC values.

By default, the TTL (Time to Live) field of each label stack entry is set to 255. This may be changed by policy at the originator of the sub-TLV. When pushing the label stack onto a packet, the TTL of the label stack entries is preserved by default. However, local policy at the router that is pushing on the stack MAY cause modification of the TTL values. If any label stack entry in the sub-TLV has a TTL value of zero, the router that is pushing the stack on a packet MUST change the value to a non-zero value.

Note that this sub-TLV can appear within a TLV identifying any type of tunnel, not just within a TLV identifying an MPLS tunnel. However, if this sub-TLV appears within a TLV identifying an MPLS tunnel (or an MPLS-in-X tunnel), this sub-TLV plays the same role that would be played by an MPLS Encapsulation sub-TLV. Therefore, an MPLS Encapsulation sub-TLV is not defined.

3.7. Prefix-SID Sub-TLV

[I-D.ietf-idr-bgp-prefix-sid] defines a BGP Path attribute known as the "Prefix-SID Attribute". This attribute is defined to contain a sequence of one or more TLVs, where each TLV is either a "Label-Index" TLV, an "IPv6 SID (Segment Identifier)" TLV, or an "Originator SRGB (Source Routing Global Block)" TLV.

In this document, we define a Prefix-SID sub-TLV. The value field of the Prefix-SID sub-TLV can be set to any valid value of the value field of a BGP Prefix-SID attribute, as defined in [I-D.ietf-idr-bgp-prefix-sid].

The Prefix-SID sub-TLV can occur in a TLV identifying any type of tunnel. If an Originator SRGB is specified in the sub-TLV, that SRGB MUST be interpreted to be the SRGB used by the tunnel's egress endpoint. The Label-Index, if present, is the Segment Routing SID that the tunnel's egress endpoint uses to represent the prefix

appearing in the NLRI field of the BGP UPDATE to which the Tunnel Encapsulation attribute is attached.

If a Label-Index is present in the prefix-SID sub-TLV, then when a packet is sent through the tunnel identified by the TLV, the corresponding MPLS label MUST be pushed on the packet's label stack. The corresponding MPLS label is computed from the Label-Index value and the SRGB of the route's originator.

If the Originator SRGB is not present, it is assumed that the originator's SRGB is known by other means. Such "other means" are outside the scope of this document.

The corresponding MPLS label is pushed on after the processing of the MPLS Label Stack sub-TLV, if present, as specified in Section 3.6. It is pushed on before any other labels (e.g., a label embedded in UPDATE's NLRI, or a label determined by the procedures of Section 8 are pushed on the stack.

The Prefix-SID sub-TLV has slightly different semantics than the Prefix-SID attribute. When the Prefix-SID attribute is attached to a given route, the BGP speaker that originally attached the attribute is expected to be in the same Segment Routing domain as the BGP speakers who receive the route with the attached attribute. The Label-Index tells the receiving BGP speakers that the prefix-SID is for the advertised prefix in that Segment Routing domain. When the Prefix-SID sub-TLV is used, the BGP speaker at the head end of the tunnel need even not be in the same Segment Routing Domain as the tunnel's egress endpoint, and there is no implication that the prefix-SID for the advertised prefix is the same in the Segment Routing domains of the BGP speaker that originated the sub-TLV and the BGP speaker that received it.

4. Extended Communities Related to the Tunnel Encapsulation Attribute

4.1. Encapsulation Extended Community

The Encapsulation Extended Community is a Transitive Opaque Extended Community. This Extended Community may be attached to a route of any AFI/SAFI to which the Tunnel Encapsulation attribute may be attached. Each such Extended Community identifies a particular tunnel type. If the Encapsulation Extended Community identifies a particular tunnel type, its semantics are exactly equivalent to the semantics of a Tunnel Encapsulation attribute Tunnel TLV for which the following three conditions all hold:

1. it identifies the same tunnel type,

2. it has a Tunnel Endpoint sub-TLV for which one of the following two conditions holds:

   A. its "Address Family" subfield contains zero, or

   B. its "Address" subfield contains the same IP address that appears in the next hop field of the route to which the Tunnel Encapsulation attribute is attached

3. it has no other sub-TLVs.

We will refer to such a Tunnel TLV as a "barebones" Tunnel TLV.

The Encapsulation Extended Community was first defined in [RFC5512]. While it provides only a small subset of the functionality of the Tunnel Encapsulation attribute, it is used in a number of deployed applications, and is still needed for backwards compatibility. To ensure backwards compatibility, this specification establishes the following rules:

1. If the Tunnel Encapsulation attribute of a given route contains a barebones Tunnel TLV identifying a particular tunnel type, an Encapsulation Extended Community identifying the same tunnel type SHOULD be attached to the route.

2. If the Encapsulation Extended Community identifying a particular tunnel type is attached to a given route, the corresponding barebones Tunnel TLV MAY be omitted from the Tunnel Encapsulation attribute.

3. Suppose a particular route has both (a) an Encapsulation Extended Community specifying a particular tunnel type, and (b) a Tunnel Encapsulation attribute with a barebones Tunnel TLV specifying that same tunnel type. Both (a) and (b) MUST be interpreted as denoting the same tunnel.

In short, in situations where one could use either the Encapsulation Extended Community or a barebones Tunnel TLV, one may use either or both. However, to ensure backwards compatibility with applications that do not support the Tunnel Encapsulation attribute, it is preferable to use the Encapsulation Extended Community. If the Extended Community (identifying a particular tunnel type) is present, the corresponding Tunnel TLV is optional.

Note that for tunnel types of the form "X-in-Y", e.g., MPLS-in-GRE, the Encapsulation Extended Community implies that only packets of the specified payload type "X" are to be carried through the tunnel of type "Y".

In the remainder of this specification, when we speak of a route as containing a Tunnel Encapsulation attribute with a TLV identifying a particular tunnel type, we are implicitly including the case where the route contains a Tunnel Encapsulation Extended Community identifying that tunnel type.

4.2. Router's MAC Extended Community

[I-D.ietf-bess-evpn-inter-subnet-forwarding] defines a Router's MAC Extended Community. This Extended Community provides information that may conflict with information in one or more of the Encapsulation Sub-TLVs of a Tunnel Encapsulation attribute. In case of such a conflict, the information in the Encapsulation Sub-TLV takes precedence.

4.3. Color Extended Community

The Color Extended Community is a Transitive Opaque Extended Community with the following encoding:

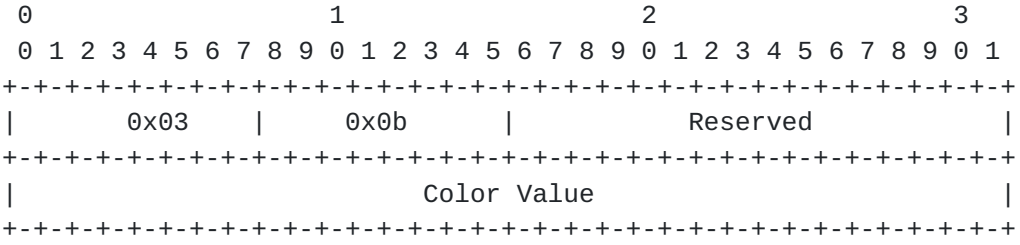


Figure 10: Color Extended Community

For the use of this Extended Community please see Section 7.

5. Semantics and Usage of the Tunnel Encapsulation attribute

[RFC5512] specifies the use of the Tunnel Encapsulation attribute in BGP UPDATE messages of AFI/SAFI 1/7 and 2/7. That document restricts the use of this attribute to UPDATE messsages of those SAFIs. This document removes that restriction.

The BGP Tunnel Encapsulation attribute MAY be carried in any BGP UPDATE message whose AFI/SAFI is 1/1 (IPv4 Unicast), 2/1 (IPv6 Unicast), 1/4 (IPv4 Labeled Unicast), 2/4 (IPv6 Labeled Unicast), 1/128 (VPN-IPv4 Labeled Unicast), 2/128 (VPN-IPv6 Labeled Unicast), or 25/70 (Ethernet VPN, usually known as EVPN)). Use of the Tunnel Encapsulation attribute in BGP UPDATE messages of other AFI/SAFIs is outside the scope of this document.

It has been suggested that it may sometimes be useful to attach a Tunnel Encapsulation attribute to a BGP UPDATE message that is also carrying a PMSI (Provider Multicast Service Interface) Tunnel attribute [RFC6514]. If the PMSI Tunnel attribute specifies an IP tunnel, the Tunnel Encapsulation attribute could be used to provide additional information about the IP tunnel. The usage of the Tunnel Encapsulation attribute in combination with the PMSI Tunnel attribute is outside the scope of this document.

The decision to attach a Tunnel Encapsulation attribute to a given BGP UPDATE is determined by policy. The set of TLVs and sub-TLVs contained in the attribute is also determined by policy.

When the Tunnel Encapsulation attribute is carried in an UPDATE of one of the AFI/SAFIs specified in the previous paragraph, each TLV MUST have a Tunnel Endpoint sub-TLV. If a TLV that does not have a Tunnel Endpoint sub-TLV, that TLV should be treated as if it had a malformed Tunnel Endpoint sub-TLV (see Section 3.1).

Suppose that:

- a given packet P must be forwarded by router R;
- the path along which P is to be forwarded is determined by BGP UPDATE U;
- UPDATE U has a Tunnel Encapsulation attribute, containing at least one TLV that identifies a "feasible tunnel" for packet P. A tunnel is considered feasible if it has the following three properties:
  - The tunnel type is supported (i.e., router R knows how to set up tunnels of that type, how to create the encapsulation header for tunnels of that type, etc.)
  - The tunnel is of a type that can be used to carry packet P (e.g., an MPLS-in-UDP tunnel would not be a feasible tunnel for carrying an IP packet, UNLESS the IP packet can first be converted to an MPLS packet).
  - The tunnel is specified in a TLV whose Tunnel Endpoint sub-TLV identifies an IP address that is reachable.

Then router R MUST send packet P through one of the feasible tunnels identified in the Tunnel Encapsulation attribute of UPDATE U.

If the Tunnel Encapsulation attribute contains several TLVs (i.e., if it specifies several tunnels), router R may choose any one of those

tunnels, based upon local policy. If any tunnel TLV contains one or more Color sub-TLVs (Section 3.4.2) and/or the Protocol Type sub-TLV (Section 3.4.1), the choice of tunnel may be influenced by these sub-TLVs.

If a particular tunnel is not feasible at some moment because its Tunnel Endpoint cannot be reached at that moment, the tunnel may become feasible at a later time (when its endpoint becomes reachable). Router R should take note of this. If router R is already using a different tunnel, it MAY switch to the tunnel that just became feasible, or it MAY decide to continue using the tunnel that it is already using. How this decision is made is outside the scope of this document.

In addition to the sub-TLVs already defined, additional sub-TLVs may be defined that affect the choice of tunnel to be used, or that affect the contents of the tunnel encapsulation header. The documents that define any such additional sub-TLVs must specify the effect that including the sub-TLV is to have.

Once it is determined to send a packet through the tunnel specified in a particular TLV of a particular Tunnel Encapsulation attribute, then the tunnel's egress endpoint address is the IP address contained in the sub-TLV. If the TLV contains a Tunnel Endpoint sub-TLV whose value field is all zeroes, then the tunnel's egress endpoint is the IP address specified as the Next Hop of the BGP Update containing the Tunnel Encapsulation attribute. The address of the tunnel egress endpoint generally appears in a "destination address" field of the encapsulation.

The full set of procedures for sending a packet through a particular tunnel type to a particular tunnel egress endpoint depends upon the tunnel type, and is outside the scope of this document. Note that some tunnel types may require the execution of an explicit tunnel setup protocol before they can be used for carrying data. Other tunnel types may not require any tunnel setup protocol.

Sending a packet through a tunnel always requires that the packet be encapsulated, with an encapsulation header that is appropriate for the tunnel type. The contents of the tunnel encapsulation header MAY be influenced by the Encapsulation sub-TLV. If there is no Encapsulation sub-TLV present, the router transmitting the packet through the tunnel must have a priori knowledge (e.g., by provisioning) of how to fill in the various fields in the encapsulation header.

Whenever a new Tunnel Type TLV is defined, the specification of that TLV should describe (or reference) the procedures for creating the

encapsulation header used to forward packets through that tunnel type. If a tunnel type codepoint is assigned in the IANA "BGP Tunnel Encapsulation Tunnel Types" registry, but there is no corresponding specification that defines an Encapsulation sub-TLV for that tunnel type, the transmitting endpoint of such a tunnel is presumed to know a priori how to form the encapsulation header for that tunnel type.

If a Tunnel Encapsulation attribute specifies several tunnels, the way in which a router chooses which one to use is a matter of policy, subject to the following constraint: if a router can determine that a given tunnel is not functional, it MUST NOT use that tunnel. In particular, if the tunnel is identified in a TLV that has a Tunnel Endpoint sub-TLV, and if the IP address specified in the sub-TLV is not reachable from router R, then the tunnel MUST be considered non-functional. Other means of determining whether a given tunnel is functional MAY be used; specification of such means is outside the scope of this specification. Of course, if a non-functional tunnel later becomes functional, router R SHOULD reevaluate its choice of tunnels.

If router R determines that it cannot use any of the tunnels specified in the Tunnel Encapsulation attribute, it MAY either drop packet P, or it MAY transmit packet P as it would had the Tunnel Encapsulation attribute not been present. This is a matter of local policy. By default, the packet SHOULD be transmitted as if the Tunnel Encapsulation attribute had not been present.

A Tunnel Encapsulation attribute may contain several TLVs that all specify the same tunnel type. Each TLV should be considered as specifying a different tunnel. Two tunnels of the same type may have different Tunnel Endpoint sub-TLVs, different Encapsulation sub-TLVs, etc. Choosing between two such tunnels is a matter of local policy.

Once router R has decided to send packet P through a particular tunnel, it encapsulates packet P appropriately and then forwards it according to the route that leads to the tunnel's egress endpoint. This route may itself be a BGP route with a Tunnel Encapsulation attribute. If so, the encapsulated packet is treated as the payload and is encapsulated according to the Tunnel Encapsulation attribute of that route. That is, tunnels may be "stacked".

Notwithstanding anything said in this document, a BGP speaker MAY have local policy that influences the choice of tunnel, and the way the encapsulation is formed. A BGP speaker MAY also have a local policy that tells it to ignore the Tunnel Encapsulation attribute entirely or in part. Of course, interoperability issues must be considered when such policies are put into place.

6. Routing Considerations

6.1. Impact on BGP Decision Process

The presence of the Tunnel Encapsulation attribute affects the BGP bestpath selection algorithm. For all the tunnels described in the Tunnel Encapsulation attribute for a path, if no Tunnel Endpoint address is feasible, then that path MUST NOT be considered resolvable for the purposes of Route Resolvability Condition [RFC4271] section 9.1.2.1.

6.2. Looping, Infinite Stacking, Etc.

Consider a packet destined for address X. Suppose a BGP UPDATE for address prefix X carries a Tunnel Encapsulation attribute that specifies a tunnel egress endpoint of Y. And suppose that a BGP UPDATE for address prefix Y carries a Tunnel Encapsulation attribute that specifies a Tunnel Endpoint of X. It is easy to see that this will cause an infinite number of encapsulation headers to be put on the given packet.

This could happen as a result of misconfiguration, either accidental or intentional. It could also happen if the Tunnel Encapsulation attribute were altered by a malicious agent. Implementations should be aware of this. This document does not specify a maximum number of recursions; that is an implementation-specific matter.

Improper setting (or malicious altering) of the Tunnel Encapsulation attribute could also cause data packets to loop. Suppose a BGP UPDATE for address prefix X carries a Tunnel Encapsulation attribute that specifies a tunnel egress endpoint of Y. Suppose router R receives and processes the update. When router R receives a packet destined for X, it will apply the encapsulation and send the encapsulated packet to Y. Y will decapsulate the packet and forward it further. If Y is further away from X than is router R, it is possible that the path from Y to X will traverse R. This would cause a long-lasting routing loop. The control plane itself cannot detect this situation, though a TTL field in the payload packets would presumably prevent any given packet from looping infinitely.

These possibilities must also be kept in mind whenever the Tunnel Endpoint for a given prefix differs from the BGP next hop for that prefix.

7. Recursive Next Hop Resolution

Suppose that:

- a given packet P must be forwarded by router R1;
- the path along which P is to be forwarded is determined by BGP UPDATE U1;
- UPDATE U1 does not have a Tunnel Encapsulation attribute;
- the next hop of UPDATE U1 is router R2;
- the best path to router R2 is a BGP route that was advertised in UPDATE U2;
- UPDATE U2 has a Tunnel Encapsulation attribute.

Then packet P MUST be sent through one of the tunnels identified in the Tunnel Encapsulation attribute of UPDATE U2. See Section 5 for further details.

However, suppose that one of the TLVs in U2's Tunnel Encapsulation attribute contains the Color Sub-TLV. In that case, packet P MUST NOT be sent through the tunnel identified in that TLV, unless U1 is carrying the Color Extended Community that is identified in U2's Color Sub-TLV.

Note that if UPDATE U1 and UPDATE U2 both have Tunnel Encapsulation attributes, packet P will be carried through a pair of nested tunnels. P will first be encapsulated based on the Tunnel Encapsulation attribute of U1. This encapsulated packet then becomes the payload, and is encapsulated based on the Tunnel Encapsulation attribute of U2. This is another way of "stacking" tunnels (see also Section 5).

The procedures in this section presuppose that U1's next hop resolves to a BGP route, and that U2's next hop resolves (perhaps after further recursion) to a non-BGP route.

8. Use of Virtual Network Identifiers and Embedded Labels when Imposing a Tunnel Encapsulation

If the TLV specifying a tunnel contains an MPLS Label Stack sub-TLV, then when sending a packet through that tunnel, the procedures of Section 3.6 are applied before the procedures of this section.

If the TLV specifying a tunnel contains a Prefix-SID sub-TLV, the procedures of Section 3.7 are applied before the procedures of this section. If the TLV also contains an MPLS Label Stack sub-TLV, the procedures of Section 3.6 are applied before the procedures of Section 3.7.

8.1. Tunnel Types without a Virtual Network Identifier Field

If a Tunnel Encapsulation attribute is attached to an UPDATE of a labeled address family, there will be one or more labels specified in the UPDATE's NLRI.

- If the TLV contains an Embedded Label Handling sub-TLV whose value is 1, the label or labels from the NLRI are pushed on the packet's label stack.

- If the TLV does not contain an Embedded Label Handling sub-TLV, or if it contains an Embedded Label Handling sub-TLV whose value is 2, the embedded label is ignored completely. The tunnel is assumed to have terminated at the corresponding VRF.

The resulting MPLS packet is then further encapsulated, as specified by the TLV.

8.2. Tunnel Types with a Virtual Network Identifier Field

Three of the tunnel types that can be specified in a Tunnel Encapsulation TLV have virtual network identifier fields in their encapsulation headers. In the VXLAN and VXLAN-GPE encapsulations, this field is called the VNI (Virtual Network Identifier) field; in the NVGRE encapsulation, this field is called the VSID (Virtual Subnet Identifier) field.

When one of these tunnel encapsulations is imposed on a packet, the setting of the virtual network identifier field in the encapsulation header depends upon the contents of the Encapsulation sub-TLV (if one is present). When the Tunnel Encapsulation attribute is being carried on a BGP UPDATE of a labeled address family, the setting of the virtual network identifier field also depends upon the contents of the Embedded Label Handling sub-TLV (if present).

This section specifies the procedures for choosing the value to set in the virtual network identifier field of the encapsulation header. These procedures apply only when the tunnel type is VXLAN, VXLAN-GPE, or NVGRE.

8.2.1. Unlabeled Address Families

This sub-section applies when:

- the Tunnel Encapsulation attribute is carried on a BGP UPDATE of an unlabeled address family, and
- at least one of the attribute's TLVs identifies a tunnel type that uses a virtual network identifier, and
- it has been determined to send a packet through one of those tunnels.

If the TLV identifying the tunnel contains an Encapsulation sub-TLV whose V bit is set, the virtual network identifier field of the encapsulation header is set to the value of the virtual network identifier field of the Encapsulation sub-TLV.

Otherwise, the virtual network identifier field of the encapsulation header is set to a configured value; if there is no configured value, the tunnel cannot be used.

8.2.2. Labeled Address Families

This sub-section applies when:

- the Tunnel Encapsulation attribute is carried on a BGP UPDATE of a labeled address family, and
- at least one of the attribute's TLVs identifies a tunnel type that uses a virtual network identifier, and
- it has been determined to send a packet through one of those tunnels.

8.2.2.1. When a Valid VNI has been Signaled

If the TLV identifying the tunnel contains an Encapsulation sub-TLV whose V bit is set, the virtual network identifier field of the encapsulation header is set as follows:

- If the TLV contains an Embedded Label Handling sub-TLV whose value is 1, then the virtual network identifier field of the encapsulation header is set to the value of the virtual network identifier field of the Encapsulation sub-TLV.

The embedded label (from the NLRI of the route that is carrying the Tunnel Encapsulation attribute) appears at the top of the MPLS label stack in the encapsulation payload.

- If the TLV does not contain an Embedded Label Handling sub-TLV, or if contains an Embedded Label Handling sub-TLV whose value is 2, the embedded label is ignored entirely, and the virtual network identifier field of the encapsulation header is set to the value of the virtual network identifier field of the Encapsulation sub-TLV.

8.2.2.2. When a Valid VNI has not been Signaled

If the TLV identifying the tunnel does not contain an Encapsulation sub-TLV whose V bit is set, the virtual network identifier field of the encapsulation header is set as follows:

- If the TLV contains an Embedded Label Handling sub-TLV whose value is 1, then the virtual network identifier field of the encapsulation header is set to a configured value.

  If there is no configured value, the tunnel cannot be used.

  The embedded label (from the NLRI of the route that is carrying the Tunnel Encapsulation attribute) appears at the top of the MPLS label stack in the encapsulation payload.

- If the TLV does not contain an Embedded Label Handling sub-TLV, or if it contains an Embedded Label Handling sub-TLV whose value is 2, the embedded label is copied into the virtual network identifier field of the encapsulation header.

  In this case, the payload may or may not contain an MPLS label stack, depending upon other factors. If the payload does contain an MPLS label stack, the embedded label does not appear in that stack.

9. Applicability Restrictions

In a given UPDATE of a labeled address family, the label embedded in the NLRI is generally a label that is meaningful only to the router whose address appears as the next hop. Certain of the procedures of Section 8.2.2.1 or Section 8.2.2.2 cause the embedded label to be carried by a data packet to the router whose address appears in the Tunnel Endpoint sub-TLV. If the Tunnel Endpoint sub-TLV does not identify the same router that is the next hop, sending the packet through the tunnel may cause the label to be misinterpreted at the tunnel's egress endpoint. This may cause misdelivery of the packet.

Therefore the embedded label MUST NOT be carried by a data packet traveling through a tunnel unless it is known that the label will be properly interpreted at the tunnel's egress endpoint. How this is known is outside the scope of this document.

Note that if the Tunnel Encapsulation attribute is attached to a VPN-IP route [RFC4364], and if Inter-AS "option b" (see section 10 of [RFC4364]) is being used, and if the Tunnel Endpoint sub-TLV contains an IP address that is not in same AS as the router receiving the route, it is very likely that the embedded label has been changed. Therefore use of the Tunnel Encapsulation attribute in an "Inter-AS option b" scenario is not supported.

10. Scoping

The Tunnel Encapsulation attribute is defined as a transitive attribute, so that it may be passed along by BGP speakers that do not recognize it. However, it is intended that the Tunnel Encapsulation attribute be used only within a well-defined scope, e.g., within a set of Autonomous Systems that belong to a single administrative entity. If the attribute is distributed beyond its intended scope, packets may be sent through tunnels in a manner that is not intended.

To prevent the Tunnel Encapsulation attribute from being distributed beyond its intended scope, any BGP speaker that understands the attribute MUST be able to filter the attribute from incoming BGP UPDATE messages. When the attribute is filtered from an incoming UPDATE, the attribute is neither processed nor redistributed. This filtering SHOULD be possible on a per-BGP-session basis. For each session, filtering of the attribute on incoming UPDATEs MUST be enabled by default.

In addition, any BGP speaker that understands the attribute MUST be able to filter the attribute from outgoing BGP UPDATE messages. This filtering SHOULD be possible on a per-BGP-session basis. For each session, filtering of the attribute on outgoing UPDATEs MUST be enabled by default.

11. Error Handling

The Tunnel Encapsulation attribute is a sequence of TLVs, each of which is a sequence of sub-TLVs. The final octet of a TLV is determined by its length field. Similarly, the final octet of a sub-TLV is determined by its length field. The final octet of a TLV MUST also be the final octet of its final sub-TLV. If this is not the case, the TLV MUST be considered to be malformed. A TLV that is found to be malformed for this reason MUST NOT be processed, and MUST be stripped from the Tunnel Encapsulation attribute before the

attribute is propagated. Subsequent TLVs in the Tunnel Encapsulation attribute may still be valid, in which case they MUST be processed and redistributed normally.

If a Tunnel Encapsulation attribute does not have any valid TLVs, or it does not have the transitive bit set, the "Attribute Discard" procedure of [RFC7606] is applied.

If a Tunnel Encapsulation attribute can be parsed correctly, but contains a TLV whose tunnel type is not recognized by a particular BGP speaker, that BGP speaker MUST NOT consider the attribute to be malformed. Rather, the TLV with the unrecognized tunnel type MUST be ignored, and the BGP speaker MUST interpret the attribute as if that TLV had not been present. If the route carrying the Tunnel Encapsulation attribute is propagated with the attribute, the unrecognized TLV MUST remain in the attribute.

If a TLV of a Tunnel Encapsulation attribute contains a sub-TLV that is not recognized by a particular BGP speaker, the BGP speaker MUST process that TLV as if the unrecognized sub-TLV had not been present. If the route carrying the Tunnel Encapsulation attribute is propagated with the attribute, the unrecognized TLV MUST remain in the attribute.

If the type code of a sub-TLV appears as "reserved" in the IANA "BGP Tunnel Encapsulation Attribute Sub-TLVs" registry, the sub-TLV MUST be treated as an unrecognized sub-TLV.

In general, if a TLV contains a sub-TLV that is malformed (e.g., contains a length field whose value is not legal for that sub-TLV), the sub-TLV should be treated as if it were an unrecognized sub-TLV. This document specifies one exception to this rule -- within a tunnel encapsulation attribute that is carried by a BGP UPDATE whose AFI/SAFI is one of those explicitly listed in the second paragraph of Section 5, if a TLV contains a malformed Tunnel Endpoint sub-TLV (as defined in Section 3.1), the entire TLV MUST be ignored, and MUST be removed from the Tunnel Encapsulation attribute before the route carrying that attribute is redistributed.

Within a tunnel encapsulation attribute that is carried by a BGP UPDATE whose AFI/SAFI is one of those explicitly listed in the second paragraph of Section 5, a TLV that does not contain exactly one Tunnel Endpoint sub-TLV MUST be treated as if it contained a malformed Tunnel Endpoint sub-TLV.

A TLV identifying a particular tunnel type may contain a sub-TLV that is meaningless for that tunnel type. For example, perhaps the TLV contains a "UDP Destination Port" sub-TLV, but the identified tunnel

type does not use UDP encapsulation at all. Sub-TLVs of this sort MUST be treated as a no-op. That is, they MUST NOT affect the creation of the encapsulation header. However, the sub-TLV MUST NOT be considered to be malformed, and MUST NOT be removed from the TLV before the route carrying the Tunnel Encapsulation attribute is redistributed. (This allows for the possibility that such sub-TLVs may be given a meaning, in the context of the specified tunnel type, in the future.)

There is no significance to the order in which the TLVs occur within the Tunnel Encapsulation attribute. Multiple TLVs may occur for a given tunnel type; each such TLV is regarded as describing a different tunnel.

The following sub-TLVs defined in this document MUST NOT occur more than once in a given Tunnel TLV: Tunnel Endpoint (discussed above), Encapsulation, IPv4 DS, UDP Destination Port, Embedded Label Handling, MPLS Label Stack, Prefix-SID. If a Tunnel TLV has more than one of any of these sub-TLVs, all but the first occurrence of each such sub-TLV type MUST be treated as a no-op. However, the Tunnel TLV containing them MUST NOT be considered to be malformed, and all the sub-TLVs MUST be propagated if the route carrying the Tunnel Encapsulation attribute is propagated.

The following sub-TLVs defined in this document may appear zero or more times in a given Tunnel TLV: Protocol Type, Color. Each occurrence of such sub-TLVs is meaningful. For example, the Color sub-TLV may appear multiple times to assign multiple colors to a tunnel.

12. IANA Considerations

12.1. Subsequent Address Family Identifiers

IANA is requested to modify the "Subsequent Address Family Identifiers" registry to indicate that the Encapsulation SAFI is deprecated. This document should be the reference.

12.2. BGP Path Attributes

IANA has previously assigned value 23 from the "BGP Path Attributes" Registry to "Tunnel Encapsulation Attribute". IANA is requested to add this document as a reference.

12.3. Extended Communities

IANA has previously assigned values from the "Transitive Opaque Extended Community" type Registry to the "Color Extended Community" (sub-type 0x0b), and to the "Encapsulation Extended Community"(0x030c). IANA is requested to add this document as a reference for both assignments.

12.4. BGP Tunnel Encapsulation Attribute Sub-TLVs

IANA is requested to add the following note to the "BGP Tunnel Encapsulation Attribute Sub-TLVs" registry:

> If the Sub-TLV Type is in the range from 0 to 127 inclusive, the Sub-TLV Length field contains one octet. If the Sub-TLV Type is in the range from 128-255 inclusive, the Sub-TLV Length field contains two octets.

IANA is requested to change the registration policy of the "BGP Tunnel Encapsulation Attribute Sub-TLVs" registry to the following:

- The values 0 and 255 are reserved.
- The values in the range 1-63 and 128-191 are to be allocated using the "Standards Action" registration procedure.
- The values in the range 64-125 and 192-252 are to be allocated using the "First Come, First Served" registration procedure.
- The values in the range 126-127 and 253-254 are reserved for experimental use; IANA shall not allocate values from this range.

IANA has assigned the following codepoints in the "BGP Tunnel Encapsulation Attribute Sub-TLVs registry:

> 6: Remote Endpoint
>
> IANA is requested to change the name of "Remote Endpoint" to "Tunnel Egress Endpoint".
>
> 7: IPv4 DS Field
>
> 8: UDP Destination Port
>
> 9: Embedded Label Handling
>
> 10: MPLS Label Stack

11: Prefix SID

IANA has previously assigned codepoints from the "BGP Tunnel Encapsulation Attribute Sub-TLVs" registry for "Encapsulation", "Protocol Type", and "Color". IANA is requested to add this document as a reference.

12.5. Tunnel Types

IANA is requested to add this document as a reference for tunnel types 8 (VXLAN), 9 (NVGRE), 11 (MPLS-in-GRE), and 12 (VXLAN-GPE) in the "BGP Tunnel Encapsulation Tunnel Types" registry.

IANA is requested to add this document as a reference for tunnel types 1 (L2TPv3), 2 (GRE), and 7 (IP in IP) in the "BGP Tunnel Encapsulation Tunnel Types" registry.

12.6. Flags Field of Vxlan Encapsulation sub-TLV

IANA is requested to add this document as a reference for creating the flags field of the Vxlan Encapsulation sub-TLV registry.

IANA is requested to add this document as a reference for flag bits V and M in the "Flags field of Vxlan Encapsulation sub-TLV" registry.

12.7. Flags Field of Vxlan-GPE Encapsulation sub-TLV

IANA is requested to add this document as a reference for creating the flags field of the Vxlan-GPE Encapsulation sub-TLV registry.

IANA is requested to add this document as a reference for flag bit V in the "Flags field of Vxlan-GPE Encapsulation sub-TLV" registry.

12.8. Flags Field of NVGRE Encapsulation sub-TLV

IANA is requested to add this document as a reference for creating the flags field of the NVGRE Encapsulation sub-TLV registry.

IANA is requested to add this document as a reference for flag bits V and M in the "Flags field of NVGRE Encapsulation sub-TLV" registry.

12.9. Embedded Label Handling sub-TLV

IANA is requested to add this document as a reference for creating the sub-TLV's value field of the Embedded Label Handling sub-TLV registry.

IANA is requested to add this document as a reference for value of 1 (Payload of MPLS with embedded label) and 2 (no embedded label in payload) in the "sub-TLV's value field of the Embedded Label Handling sub-TLV" registry.

13. Security Considerations

The Tunnel Encapsulation attribute can cause traffic to be diverted from its normal path, especially when the Tunnel Endpoint sub-TLV is used. This can have serious consequences if the attribute is added or modified illegitimately, as it enables traffic to be "hijacked".

The Tunnel Endpoint sub-TLV contains both an IP address and an AS number. BGP Origin Validation [RFC6811] can be used to obtain assurance that the given IP address belongs to the given AS. While this provides some protection against misconfiguration, it does not prevent a malicious agent from inserting a sub-TLV that will appear valid.

Before sending a packet through the tunnel identified in a particular TLV of a Tunnel Encapsulation attribute, it may be advisable to use BGP Origin Validation to obtain the following additional assurances:

- the origin AS of the route carrying the Tunnel Encapsulation attribute is correct;

- the origin AS of the route to the IP address specified in the Tunnel Endpoint sub-TLV is correct, and is the same AS that is specified in the Tunnel Endpoint sub-TLV.

One then has some level of assurance that the tunneled traffic is going to the same destination AS that it would have gone to had the Tunnel Encapsulation attribute not been present. However, this may not suit all use cases, and in any event is not very strong protection against hijacking.

For these reasons, BGP Origin Validation should not be relied upon exclusively, and the filtering procedures of Section 10 should always be in place.

Increased protection can be obtained by using BGPSEC [RFC8205] to ensure that the route carrying the Tunnel Encapsulation attribute, and the routes to the Tunnel Endpoint of each specified tunnel, have not been altered illegitimately.

If BGP Origin Validation is used as specified above, and the tunnel specified in a particular TLV of a Tunnel Encapsulation attribute is therefore regarded as "suspicious", that tunnel should not be used.

Other tunnels specified in (other TLVs of) the Tunnel Encapsulation attribute may still be used.

14. Acknowledgments

This document contains text from RFC5512, co-authored by Pradosh Mohapatra. The authors of the current document wish to thank Pradosh for his contribution. RFC5512 itself built upon prior work by Gargi Nalawade, Ruchi Kapoor, Dan Tappan, David Ward, Scott Wainner, Simon Barber, Lili Wang, and Chris Metz, whom we also thank for their contributions.

The authors wish to thank Lou Berger, Ron Bonica, Martin Djernaes, John Drake, Satoru Matsushima, Dhananjaya Rao, John Scudder, Ravi Singh, Thomas Morin, Xiaohu Xu, and Zhaohui Zhang for their review, comments, and/or helpful discussions.

15. Contributor Addresses

Below is a list of other contributing authors in alphabetical order:

Randy Bush
Internet Initiative Japan
5147 Crystal Springs
Bainbridge Island, Washington 98110
United States

Email: randy@psg.com

Robert Raszuk
Bloomberg LP
731 Lexington Ave
New York City, NY 10022
United States

Email: robert@raszuk.net

Eric C. Rosen

16. References

16.1. Normative References

[I-D.ietf-idr-bgp-prefix-sid]
Previdi, S., Filsfils, C., Lindem, A., Sreekantiah, A., and H. Gredler, "Segment Routing Prefix SID extensions for BGP", draft-ietf-idr-bgp-prefix-sid-27 (work in progress), June 2018.

[I-D.ietf-nvo3-vxlan-gpe]
Maino, F., Kreeger, L., and U. Elzur, "Generic Protocol Extension for VXLAN", draft-ietf-nvo3-vxlan-gpe-07 (work in progress), April 2019.

[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>.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <https://www.rfc-editor.org/info/rfc2784>.

[RFC2890] Dommety, G., "Key and Sequence Number Extensions to GRE", RFC 2890, DOI 10.17487/RFC2890, September 2000, <https://www.rfc-editor.org/info/rfc2890>.

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001, <https://www.rfc-editor.org/info/rfc3032>.

[RFC3931] Lau, J., Ed., Townsley, M., Ed., and I. Goyret, Ed., "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, DOI 10.17487/RFC3931, March 2005, <https://www.rfc-editor.org/info/rfc3931>.

[RFC4023] Worster, T., Rekhter, Y., and E. Rosen, Ed., "Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)", RFC 4023, DOI 10.17487/RFC4023, March 2005, <https://www.rfc-editor.org/info/rfc4023>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <https://www.rfc-editor.org/info/rfc4760>.

[RFC5512] Mohapatra, P. and E. Rosen, "The BGP Encapsulation Subsequent Address Family Identifier (SAFI) and the BGP Tunnel Encapsulation Attribute", RFC 5512, DOI 10.17487/RFC5512, April 2009, <https://www.rfc-editor.org/info/rfc5512>.

[RFC5566] Berger, L., White, R., and E. Rosen, "BGP IPsec Tunnel Encapsulation Attribute", RFC 5566, DOI 10.17487/RFC5566, June 2009, <https://www.rfc-editor.org/info/rfc5566>.

[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>.

[RFC7510] Xu, X., Sheth, N., Yong, L., Callon, R., and D. Black, "Encapsulating MPLS in UDP", RFC 7510, DOI 10.17487/RFC7510, April 2015, <https://www.rfc-editor.org/info/rfc7510>.

[RFC7606] Chen, E., Ed., Scudder, J., Ed., Mohapatra, P., and K. Patel, "Revised Error Handling for BGP UPDATE Messages", RFC 7606, DOI 10.17487/RFC7606, August 2015, <https://www.rfc-editor.org/info/rfc7606>.

[RFC7637] Garg, P., Ed. and Y. Wang, Ed., "NVGRE: Network Virtualization Using Generic Routing Encapsulation", RFC 7637, DOI 10.17487/RFC7637, September 2015, <https://www.rfc-editor.org/info/rfc7637>.

[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>.

16.2. Informative References

[Ethertypes] "IANA Ethertype Registry", <http://www.iana.org/assignments/ieee-802-numbers/ieee-802-numbers.xhtml>.

[I-D.ietf-bess-evpn-inter-subnet-forwarding]
Sajassi, A., Salam, S., Thoria, S., Drake, J., and J. Rabadan, "Integrated Routing and Bridging in EVPN", draft-ietf-bess-evpn-inter-subnet-forwarding-08 (work in progress), March 2019.

[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>.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <https://www.rfc-editor.org/info/rfc4364>.

[RFC5462] Andersson, L. and R. Asati, "Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic Class" Field", RFC 5462, DOI 10.17487/RFC5462, February 2009, <https://www.rfc-editor.org/info/rfc5462>.

[RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, DOI 10.17487/RFC6514, February 2012, <https://www.rfc-editor.org/info/rfc6514>.

[RFC6811] Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", RFC 6811, DOI 10.17487/RFC6811, January 2013, <https://www.rfc-editor.org/info/rfc6811>.

[RFC8205] Lepinski, M., Ed. and K. Sriram, Ed., "BGPsec Protocol Specification", RFC 8205, DOI 10.17487/RFC8205, September 2017, <https://www.rfc-editor.org/info/rfc8205>.

Authors' Addresses

Keyur Patel
Arrcus, Inc
2077 Gateway Pl
San Jose, CA 95110
United States

Email: keyur@arrcus.com

Gunter Van de Velde
Nokia
Copernicuslaan 50
Antwerpen 2018
Belgium

Email: gunter.van_de_velde@nokia.com

Srihari R. Sangli
Juniper Networks, Inc
10 Technology Park Drive
Westford, Massachusetts 01886
United States

Email: ssangli@juniper.net