Internet Engineering Task Force (IETF)
Request for Comments: 9012
Obsoletes: 5512, 5566
Updates: 5640
Category: Standards Track
ISSN: 2070-1721

K. Patel
Arrcus, Inc
G. Van de Velde
Nokia
S. Sangli
J. Scudder
Juniper Networks

April 2021

The BGP Tunnel Encapsulation Attribute

Abstract

This document defines a BGP path attribute known as the "Tunnel Encapsulation attribute", which can be used with BGP UPDATEs of various Subsequent Address Family Identifiers (SAFIs) to provide information needed to create tunnels and their corresponding encapsulation headers. It provides encodings for a number of tunnel types, along with procedures for choosing between alternate tunnels and routing packets into tunnels.

This document obsoletes RFC 5512, which provided an earlier definition of the Tunnel Encapsulation attribute. RFC 5512 was never deployed in production. Since RFC 5566 relies on RFC 5512, it is likewise obsoleted. This document updates RFC 5640 by indicating that the Load-Balancing Block sub-TLV may be included in any Tunnel Encapsulation attribute where load balancing is desired.

Status of This Memo

This is an Internet Standards Track document.

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

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9012.

described in the Simplified BSD License.

Table of Contents

1. Introduction

This document obsoletes [RFC5512]. The deficiencies of [RFC5512], and a summary of the changes made, are discussed in Sections 1.1-1.3. The material from [RFC5512] that is retained has been incorporated into this document. Since [RFC5566] relies on [RFC5512], it is likewise obsoleted.

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, for example, the Encapsulation sub-TLV, contain information that may be used to form the encapsulation header for the specified tunnel type. Other sub-TLVs, for example, 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 Address Family Identifier (AFI) of 1 or 2, and a SAFI of 7. In an UPDATE of the Encapsulation SAFI, the Network Layer Reachability Information (NLRI) is an address of the BGP speaker originating the UPDATE. Consider the following scenario:

* BGP speaker R1 has received and selected UPDATE U for local use;

* 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; and

* R1's best route 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 routes 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 the specification of the sub-TLVs in [RFC5512], each sub-TLV has a one-octet Length field. In some cases, where a sub-TLV may require more than 255 octets for its encoding, a two-octet Length field may be needed.

1.3. Use Case for the Tunnel Encapsulation Attribute

Consider the case of a router R1 forwarding an IP packet P. Let D be P's IP destination address. R1 must look up D in its forwarding table. Suppose that the "best match" route for D is route Q, where Q is a BGP-distributed route whose "BGP next hop" is router R2. And suppose further that the routers along the path from R1 to R2 have entries for R2 in their forwarding tables but do NOT have entries for D in their forwarding tables. For example, the path from R1 to R2 may be part of a "BGP-free core", where there are no BGP-distributed routes at all in the core. Or, as in [RFC5565], D may be an IPv4 address while the intermediate routers along the path from R1 to R2

may support only IPv6.

In cases such as this, in order for R1 to properly forward packet P, it must encapsulate P and send P "through a tunnel" to R2. For example, R1 may encapsulate P using GRE, Layer 2 Tunneling Protocol version 3 (L2TPv3), IP in IP, etc., where the destination IP address of the encapsulation header is the address of R2.

In order for R1 to encapsulate P for transport to R2, R1 must know what encapsulation protocol to use for transporting different sorts of packets to R2. R1 must also know how to fill in the various fields of the encapsulation header. With certain encapsulation types, this knowledge may be acquired by default or through manual configuration. Other encapsulation protocols have fields such as session id, key, or cookie that must be filled in. It would not be desirable to require every BGP speaker to be manually configured with the encapsulation information for every one of its BGP next hops.

This document specifies a way in which BGP itself can be used by a given BGP speaker to tell other BGP speakers, "If you need to encapsulate packets to be sent to me, here's the information you need to properly form the encapsulation header". A BGP speaker signals this information to other BGP speakers by using a new BGP attribute type value -- the BGP Tunnel Encapsulation attribute. This attribute specifies the encapsulation protocols that may be used, as well as whatever additional information (if any) is needed in order to properly use those protocols. Other attributes, for example, communities or extended communities, may also be included.

1.4. Brief Summary of Changes from RFC 5512

This document addresses the deficiencies identified in Section 1.2 by:

* Deprecating the Encapsulation SAFI.

* Defining a new "Tunnel Egress Endpoint sub-TLV" (Section 3.1) 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]: Virtual eXtensible Local Area Network (VXLAN) [RFC7348], Network Virtualization Using Generic Routing Encapsulation (NVGRE) [RFC7637], and MPLS in Generic Routing Encapsulation (MPLS-in-GRE) [RFC4023]. MPLS-in-UDP [RFC7510] is also supported, but an Encapsulation sub-TLV for it is not needed since there are no additional parameters to be signaled.

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), 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 an Encapsulation Extended Community that can be used instead of the Tunnel Encapsulation attribute under certain circumstances. This document describes how the Encapsulation Extended Community can be used in a backwards-compatible fashion (see Section 4.1). It is possible to combine Encapsulation Extended Communities and Tunnel Encapsulation attributes in the same BGP UPDATE in this manner.

1.5. Update to RFC 5640

This document updates [RFC5640] by indicating that the Load-Balancing Block sub-TLV MAY be included in any Tunnel Encapsulation attribute where load balancing is desired.

1.6. Effects of Obsoleting RFC 5566

This specification obsoletes RFC 5566. This has the effect of, in turn, deprecating a number of code points defined in that document. In the "BGP Tunnel Encapsulation Attribute Tunnel Types" registry [IANA-BGP-TUNNEL-ENCAP], the following code points have been marked as deprecated: "Transmit tunnel endpoint" (type code 3), "IPsec in Tunnel-mode" (type code 4), "IP in IP tunnel with IPsec Transport Mode" (type code 5), and "MPLS-in-IP tunnel with IPsec Transport Mode" (type code 6). In the "BGP Tunnel Encapsulation Attribute Sub-TLVs" registry [IANA-BGP-TUNNEL-ENCAP], "IPsec Tunnel Authenticator" (type code 3) has been marked as deprecated. See Section 14.2.

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 in the "BGP Path Attributes" registry [IANA-BGP-PARAMS]. The attribute is composed of a set of Type-Length-Value (TLV) encodings. Each TLV contains information corresponding to a particular tunnel type. A Tunnel Encapsulation TLV, also known as Tunnel TLV, is structured as shown in Figure 1.

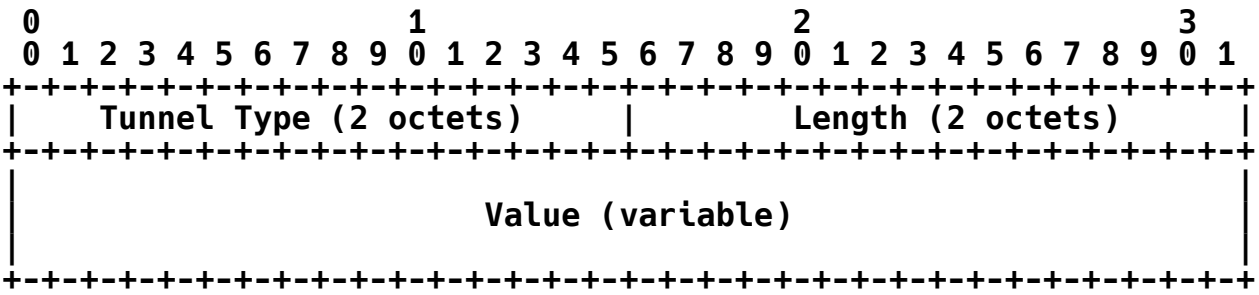


Figure 1: Tunnel Encapsulation TLV

Tunnel Type (2 octets): Identifies a type of tunnel. The field contains values from the IANA registry "BGP Tunnel Encapsulation Attribute Tunnel Types" [IANA-BGP-TUNNEL-ENCAP]. See Section 3.4.1 for discussion of special treatment of tunnel types with names of the form "X-in-Y".

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 (depending on the type), and zero or more octets of value. A sub-TLV is structured as shown in Figure 2.

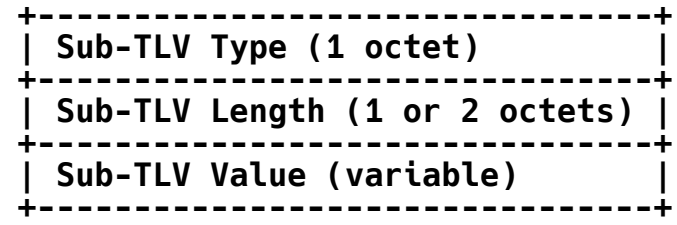


Figure 2: Encapsulation Sub-TLV

Sub-TLV Type (1 octet): Each sub-TLV type defines a certain property about the Tunnel TLV that contains this sub-TLV. The field contains values from the IANA registry "BGP Tunnel Encapsulation Attribute Sub-TLVs" [IANA-BGP-TUNNEL-ENCAP].

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. The following subsections define the encoding in detail.

3. Tunnel Encapsulation Attribute Sub-TLVs

This section specifies a number of sub-TLVs. These sub-TLVs can be included in a TLV of the Tunnel Encapsulation attribute.

3.1. The Tunnel Egress Endpoint Sub-TLV (Type Code 6)

The Tunnel Egress Endpoint sub-TLV specifies the address of the egress endpoint of the tunnel, that is, the address of the router that will decapsulate the payload. Its Value field contains three subfields:

1. a Reserved subfield
2. a two-octet Address Family subfield
3. an Address subfield, whose length depends upon the Address Family.

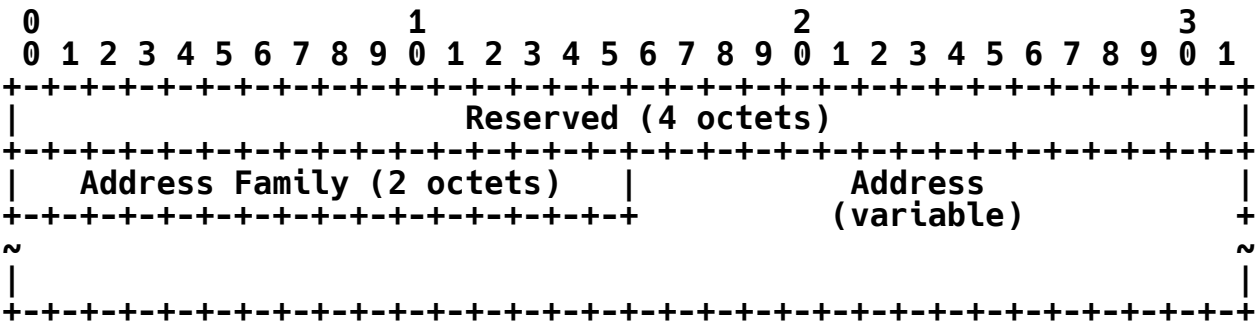


Figure 3: Tunnel Egress Endpoint Sub-TLV Value Field

The Reserved subfield SHOULD be originated as zero. It MUST be disregarded on receipt, and it MUST be propagated unchanged.

The Address Family subfield contains a value from IANA's "Address Family Numbers" registry [IANA-ADDRESS-FAM]. This document assumes 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).

If the Address Family subfield contains the value for IPv6, the Address subfield MUST contain an IPv6 address (a /128 IPv6 prefix).

In a given BGP UPDATE, the address family (IPv4 or IPv6) of a Tunnel Egress 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 Egress Endpoint sub-TLVs that contain IPv6 addresses. Also, different

tunnels represented in the Tunnel Encapsulation attribute may have tunnel egress endpoints of different address families.

There is one special case: the Tunnel Egress Endpoint sub-TLV MAY have a Value field whose Address Family subfield contains 0. This means that the tunnel's egress endpoint is the address of the next hop. If the Address Family subfield contains 0, the Address subfield is omitted. In this case, the Length field of Tunnel Egress Endpoint sub-TLV MUST contain the value 6 (0x06).

When the Tunnel Encapsulation attribute is carried in an UPDATE message of one of the AFI/SAFIs specified in this document (see the first paragraph of Section 6), each TLV MUST have one, and only one, Tunnel Egress Endpoint sub-TLV. If a TLV does not have a Tunnel Egress Endpoint sub-TLV, that TLV should be treated as if it had a malformed Tunnel Egress Endpoint sub-TLV (see below).

In the context of this specification, if the Address Family subfield has any value other than IPv4, IPv6, or the special value 0, the Tunnel Egress Endpoint sub-TLV is considered "unrecognized" (see Section 13). If any of the following conditions hold, the Tunnel Egress Endpoint sub-TLV is considered to be "malformed":

* The length of the sub-TLV's Value field is other than 6 added to the defined length for the address family given in its Address Family subfield. Therefore, for address family behaviors defined in this document, the permitted values are:

  - 10, if the Address Family subfield contains the value for IPv4.

  - 22, if the Address Family subfield contains the value for IPv6.

  - 6, if the Address Family subfield contains the value zero.

* The IP address in the sub-TLV's Address subfield lies within a block listed in the relevant Special-Purpose IP Address registry [RFC6890] with either a "destination" attribute value or a "forwardable" attribute value of "false". (Such routes are sometimes colloquially known as "Martians".) This restriction MAY be relaxed by explicit configuration.

* It can be determined that the IP address in the sub-TLV's Address subfield does not belong to the Autonomous System (AS) that originated the route that contains the attribute. Section 3.1.1 describes an optional procedure to make this determination.

Error handling is specified in Section 13.

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

3.1.1. Validating the Address Subfield

This section provides a procedure that MAY be applied to validate that the IP address in the sub-TLV's Address subfield belongs to the

AS that originated the route that contains the attribute. (The notion of "belonging to" an AS is expanded on below.) Doing this is thought to increase confidence that when traffic is sent to the IP address depicted in the Address subfield, it will go to the same AS as it would go to if the Tunnel Encapsulation attribute were not present, although of course it cannot guarantee it. See Section 15 for discussion of the limitations of this procedure. The principal applicability of this procedure is in deployments that are not strictly scoped. In deployments with strict scope, and especially those scoped to a single AS, these procedures may not add substantial benefit beyond those discussed in Section 11.

The Route Origin Autonomous System Number (ASN) of a BGP route that includes a Tunnel Encapsulation attribute can be determined by inspection of the AS_PATH attribute, according to the procedure specified in [RFC6811], Section 2. Call this value Route_AS.

In order to determine the Route Origin ASN of the address depicted in the Address subfield of the Tunnel Egress Endpoint sub-TLV, it is necessary to consider the forwarding route -- that is, the route that will be used to forward traffic toward that address. This route is determined by a recursive route-lookup operation for that address, as discussed in [RFC4271], Section 5.1.3. The relevant AS path to consider is the last one encountered while performing the recursive lookup; the procedures of [RFC6811], Section 2 are applied to that AS path to determine the Route Origin ASN. If no AS path is encountered at all, for example, if that route's source is a protocol other than BGP, the Route Origin ASN is the BGP speaker's own AS number. Call this value Egress_AS.

If Route_AS does not equal Egress_AS, then the Tunnel Egress Endpoint sub-TLV is considered not to be valid. In some cases, a network operator who controls a set of ASes might wish to allow a tunnel egress endpoint to reside in an AS other than Route_AS; configuration MAY allow for such a case, in which case the check becomes: if Egress_AS is not within the configured set of permitted AS numbers, then the Tunnel Egress Endpoint sub-TLV is considered to be "malformed".

Note that if the forwarding route changes, this procedure MUST be reapplied. As a result, a sub-TLV that was formerly considered valid might become not valid, or vice versa.

3.2. Encapsulation Sub-TLVs for Particular Tunnel Types (Type Code 1)

This section defines Encapsulation sub-TLVs for the following tunnel types: VXLAN [RFC7348], NVGRE [RFC7637], MPLS-in-GRE [RFC4023], L2TPv3 [RFC3931], and GRE [RFC2784].

Rules for forming the encapsulation based on the information in a given TLV are given in Sections 6 and 9.

Recall that the tunnel type itself is identified by the Tunnel Type field in the attribute header (Section 2); the Encapsulation sub-TLV's structure is inferred from this. Regardless of the tunnel type, the sub-TLV type of the Encapsulation sub-TLV is 1. 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 (Tunnel Type 8)

This document defines an Encapsulation sub-TLV for VXLAN [RFC7348] tunnels. When the tunnel type is VXLAN, the length of the sub-TLV is 12 octets. The structure of the Value field in the Encapsulation sub-TLV is shown in Figure 4.

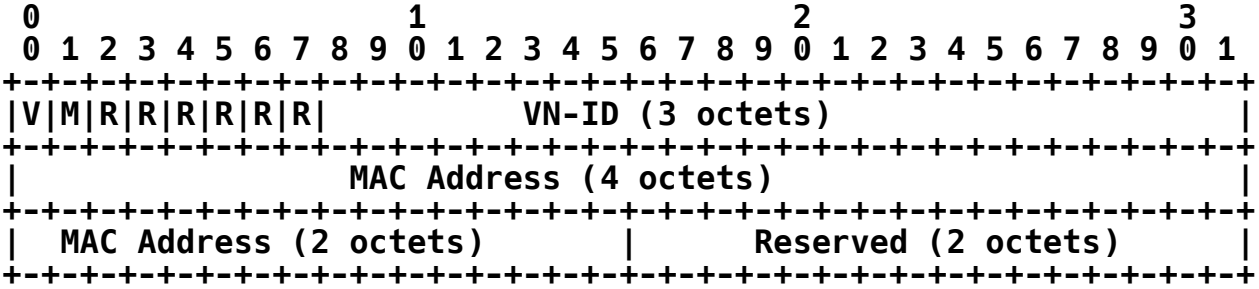


Figure 4: VXLAN Encapsulation Sub-TLV Value Field

V: This bit is set to 1 to indicate that a Virtual Network Identifier (VN-ID) is present in the Encapsulation sub-TLV. If set to 0, the VN-ID field is disregarded. Please see Section 9.

M: This bit is set to 1 to indicate that a Media Access Control (MAC) Address is present in the Encapsulation sub-TLV. If set to 0, the MAC Address field is disregarded.

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 receipt.

VN-ID: If the V bit is set to 1, the VN-ID field contains a 3-octet VN-ID value. If the V bit is set to 0, the VN-ID field MUST be set to zero on transmission and disregarded on receipt.

MAC Address: If the M bit is set to 1, this field contains a 6-octet Ethernet MAC address. If the M bit is set to 0, this field MUST be set to all zeroes on transmission and disregarded on receipt.

Reserved: MUST be set to zero on transmission and disregarded on receipt.

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 to 1, 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 set to 0, 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 set to 0, 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.

* If the V bit is set to 0, and the BGP UPDATE message has an AFI/SAFI other than Ethernet VPNs (SAFI 70, "BGP EVPNs"), then the VXLAN tunnel cannot be used.

* Section 9 describes how the VNI (VXLAN Network Identifier) 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 (for example, the MAC address) that is used to form this Ethernet header.

3.2.2. NVGRE (Tunnel Type 9)

This document defines an Encapsulation sub-TLV for NVGRE [RFC7637] tunnels. When the tunnel type is NVGRE, the length of the sub-TLV is 12 octets. The structure of the Value field in the Encapsulation sub-TLV is shown in Figure 5.

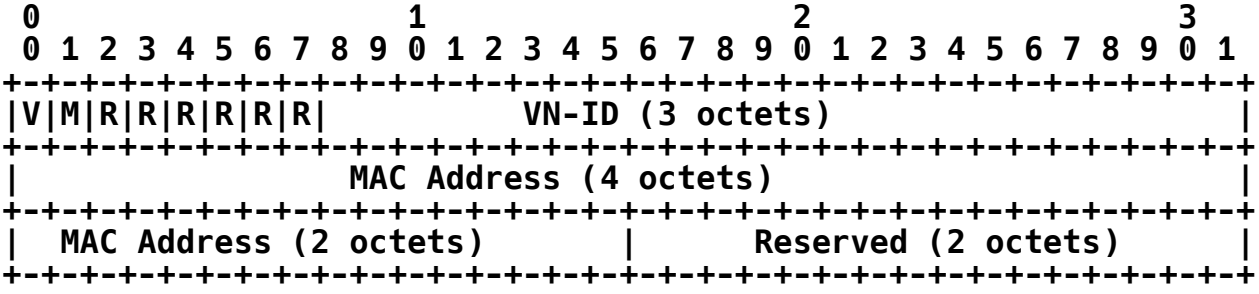


Figure 5: NVGRE Encapsulation Sub-TLV Value Field

V: This bit is set to 1 to indicate that a VN-ID is present in the Encapsulation sub-TLV. If set to 0, the VN-ID field is disregarded. Please see Section 9.

M: This bit is set to 1 to indicate that a MAC Address is present in the Encapsulation sub-TLV. If set to 0, the MAC Address field is disregarded.

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 receipt.

VN-ID: If the V bit is set to 1, the VN-ID field contains a 3-octet VN-ID value, used to set the NVGRE Virtual Subnet Identifier (VSID; see Section 9). If the V bit is set to 0, the VN-ID field MUST be set to zero on transmission and disregarded on receipt.

MAC Address: If the M bit is set to 1, this field contains a 6-octet Ethernet MAC address. If the M bit is set to 0, this field MUST be set to all zeroes on transmission and disregarded on receipt.

Reserved: MUST be set to zero on transmission and disregarded on receipt.

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 NVGRE header is set as per [RFC7637].

* If the M bit is set to 1, 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 set to 0, 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.

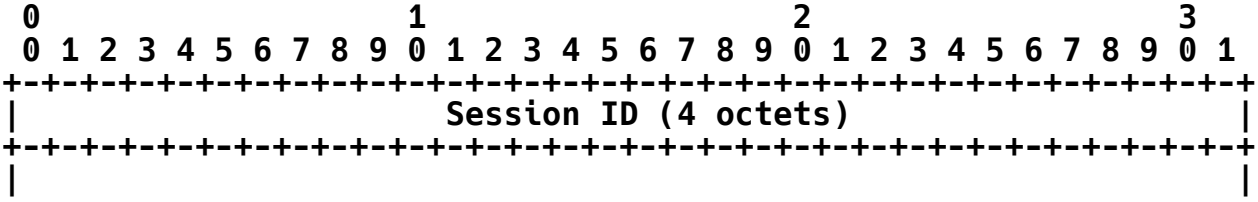
  If the M bit is set to 0, and the payload being sent through the NVGRE tunnel is an IP or MPLS packet, the Inner Destination MAC Address field is set to a configured value; if there is no configured value, the NVGRE tunnel cannot be used.

* If the V bit is set to 0, and the BGP UPDATE message has an AFI/SAFI other than Ethernet VPNs (EVPNs), then the NVGRE tunnel cannot be used.

* Section 9 describes how the VSID field of the NVGRE encapsulation header is set.

3.2.3. L2TPv3 (Tunnel Type 1)

When the tunnel type of the TLV is L2TPv3 over IP [RFC3931], the length of the sub-TLV is between 4 and 12 octets, depending on the length of the cookie. The structure of the Value field of the Encapsulation sub-TLV is shown in Figure 6.



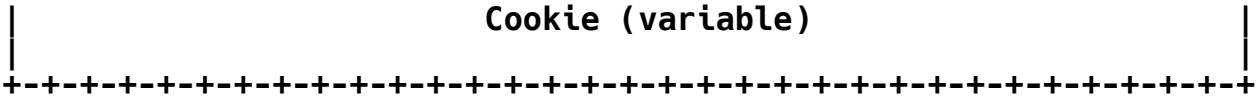


Figure 6: L2TPv3 Encapsulation Sub-TLV Value Field

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

Cookie: An optional, variable-length (encoded in 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).

3.2.4. GRE (Tunnel Type 2)

When the tunnel type of the TLV is GRE [RFC2784], the length of the sub-TLV is 4 octets. The structure of the Value field of the Encapsulation sub-TLV is shown in Figure 7.

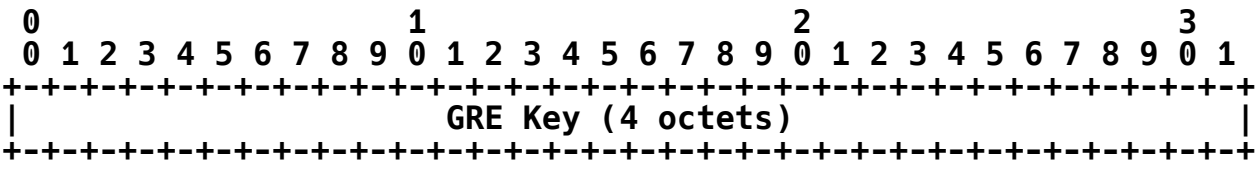


Figure 7: GRE Encapsulation Sub-TLV Value Field

GRE Key: 4-octet field [RFC2890] that is generated by the advertising router. Note that the key is optional. Unless a key value is being advertised, the GRE Encapsulation sub-TLV MUST NOT be present.

3.2.5. MPLS-in-GRE (Tunnel Type 11)

When the tunnel type is MPLS-in-GRE [RFC4023], the length of the sub-TLV is 4 octets. The structure of the Value field of the Encapsulation sub-TLV is shown in Figure 8.

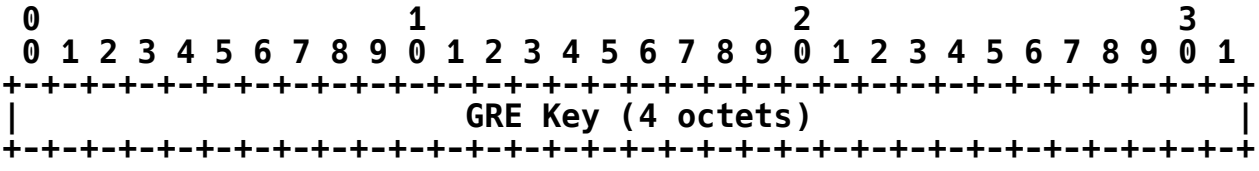


Figure 8: MPLS-in-GRE Encapsulation Sub-TLV Value Field

GRE Key: 4-octet field [RFC2890] that is generated by the advertising router. 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 3.2.4 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.

Although the MPLS-in-GRE tunnel type is just a special case of the GRE tunnel type and thus is not strictly necessary, it is included for reasons of backwards compatibility with, for example, implementations of [RFC8365].

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 MUST be treated as if it were an unrecognized type of sub-TLV.

3.3.1. DS Field (Type Code 7)

Most of the tunnel types that can be specified in the Tunnel Encapsulation attribute require an outer IP encapsulation. The 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 IPv4 or IPv6 encapsulation (see [RFC2474]). Any one-octet value can be transported; the semantics of the DSCP (Differentiated Services Code Point) field is beyond the scope of this document. The Value field is always a single octet.

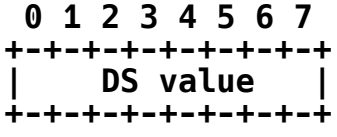


Figure 9: DS Field Sub-TLV Value Field

Because the interpretation of the DSCP field at the recipient may be different from its interpretation at the originator, an implementation MAY provide a facility to use policy to filter or modify the DS field.

3.3.2. UDP Destination Port (Type Code 8)

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 nonstandard 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. Any two-octet value other than zero can be transported. If the reserved value zero is received, the sub-TLV MUST be treated as malformed, according to the rules of Section 13.

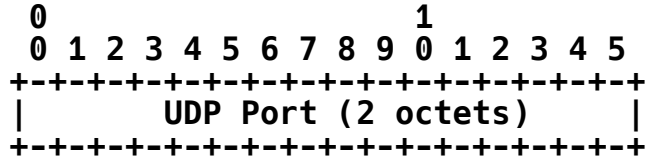


Figure 10: UDP Destination Port Sub-TLV Value Field

3.4. Sub-TLVs for Aiding Tunnel Selection

3.4.1. Protocol Type Sub-TLV (Type Code 2)

The Protocol Type sub-TLV MAY be included in a given TLV to indicate the type of the payload packets that are allowed to be encapsulated with the tunnel parameters that are being signaled in the TLV. Packets with other payload types MUST NOT be encapsulated in the relevant tunnel. The Value field of the sub-TLV contains a 2-octet value from IANA's "ETHER TYPES" registry [IANA-ETHERTYPES]. If the reserved value 0xFFFF is received, the sub-TLV MUST be treated as malformed according to the rules of Section 13.

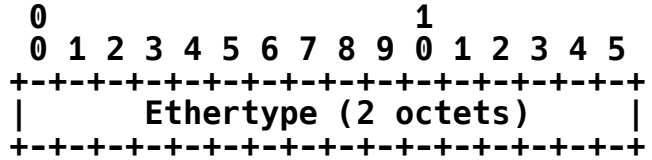


Figure 11: Protocol Type Sub-TLV Value Field

For example, if there are 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.

Note that for tunnel types whose names are of the form "X-in-Y" (for example, 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 explicitly include a Protocol Type sub-TLV specifying "X". Also, for "X-in-Y" type tunnels, a Protocol Type sub-TLV specifying anything other than "X" MUST be ignored; this is discussed further in Section 13.

3.4.2. Color Sub-TLV (Type Code 4)

The Color sub-TLV MAY be used 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 8.

The format of the Value field is depicted in Figure 15.

If the Length field of a Color sub-TLV has a value other than 8, or the first two octets of its Value field are not 0x030b, the sub-TLV MUST be treated as if it were an unrecognized sub-TLV (see Section 13).

3.5. Embedded Label Handling Sub-TLV (Type Code 9)

Certain BGP address families (corresponding to particular AFI/SAFI pairs, for example, 1/4, 2/4, 1/128, 2/128) have MPLS labels embedded in their NLRIs. The term "embedded label" is used 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 (for example, VXLAN and NVGRE) that can be specified in the Tunnel Encapsulation attribute have an encapsulation header containing a 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 a labeled 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, then the sub-TLV MUST be disregarded. 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 MUST be disregarded. In those cases where the sub-TLV is ignored, it MUST NOT be stripped from the TLV before the route is propagated.

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 either carried in the Virtual Network Identifier field of the encapsulation header or else ignored entirely.

If any value other than 1 or 2 is carried, the sub-TLV MUST be considered malformed, according to the procedures of Section 13.

Please see Section 9 for the details of how this sub-TLV is used when it is carried by an UPDATE of a labeled address family.

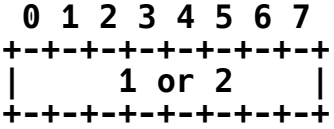


Figure 12: Embedded Label Handling Sub-TLV Value Field

3.6. MPLS Label Stack Sub-TLV (Type Code 10)

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

The length of the sub-TLV is a multiple of 4 octets, and 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, and 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 format shown in Figure 13.

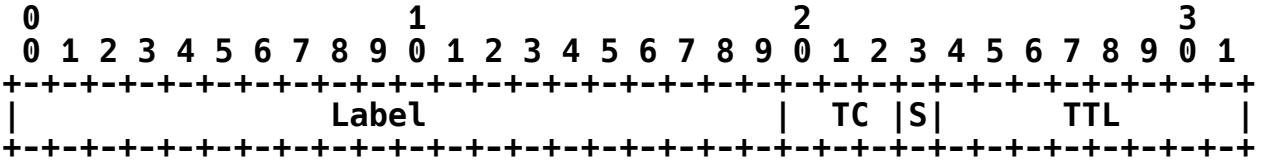


Figure 13: MPLS Label Stack Sub-TLV Value Field

The fields are as defined in [RFC3032] and [RFC5462].

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 before any other labels are pushed onto the packet. (See Section 6 for further discussion.)

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 9), the contents of the MPLS Label Stack sub-TLV MUST be pushed onto the packet before the procedures of Section 9 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 being pushed 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.

The Traffic Class (TC) field [RFC3270][RFC5129] of each label stack entry SHOULD be set to 0, unless changed by policy at the originator of the sub-TLV. When pushing the label stack onto a packet, the TC of each label stack SHOULD be preserved, unless local policy results in a modification.

The TTL (Time to Live) field of each label stack entry SHOULD be set to 255, unless changed to some other non-zero value by policy at the originator of the sub-TLV. When pushing the label stack onto a packet, the TTL of each label stack entry SHOULD be preserved, unless local policy results in a modification to some other non-zero value. If any label stack entry in the sub-TLV has a TTL value of zero, the router that is pushing the stack onto a packet MUST change the value to a non-zero value, either 255 or some other value as determined by policy as discussed above.

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.

Although this specification does not supply detailed instructions for validating the received label stack, implementations might impose restrictions on the label stack they can support. If an invalid or unsupported label stack is received, the tunnel MAY be treated as not feasible, according to the procedures of Section 6.

3.7. Prefix-SID Sub-TLV (Type Code 11)

[RFC8669] defines a BGP path attribute known as the "BGP 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 or an Originator SRGB (Source Routing Global Block) TLV.

This document defines a Prefix-SID (Prefix Segment Identifier) sub-TLV. The Value field of the Prefix-SID sub-TLV can be set to any permitted value of the Value field of a BGP Prefix-SID attribute [RFC8669].

[RFC8669] only defines behavior when the BGP Prefix-SID attribute is attached to routes of type IPv4/IPv6 Labeled Unicast [RFC4760][RFC8277], and it only defines values of the BGP Prefix-SID attribute for those cases. Therefore, similar limitations exist for the Prefix-SID sub-TLV: it SHOULD only be included in a BGP UPDATE message for one of the address families for which [RFC8669] has a defined behavior, namely BGP IPv4/IPv6 Labeled Unicast [RFC4760] [RFC8277]. If included in a BGP UPDATE for any other address family, it MUST be ignored.

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, if that tunnel is from a labeled address family, 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, as specified in Section 4.1 of [RFC8669].

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 (for example, a label embedded in an UPDATE's NLRI or a label determined by the procedures of Section 9) are pushed on the stack.

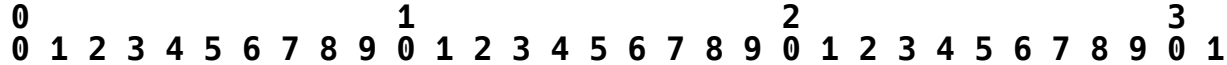
The Prefix-SID sub-TLV has slightly different semantics than the BGP Prefix-SID attribute. When the BGP Prefix-SID attribute is attached to a given route, the BGP speaker that originally attached the attribute is expected to be in the same Segment Routing domain as the BGP speakers who receive the route with the attached attribute. The Label-Index tells the receiving BGP speakers what the Prefix-SID is for the advertised prefix in that Segment Routing domain. When the Prefix-SID sub-TLV is used, 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.

The Encapsulation Extended Community encoding is as shown in Figure 14.



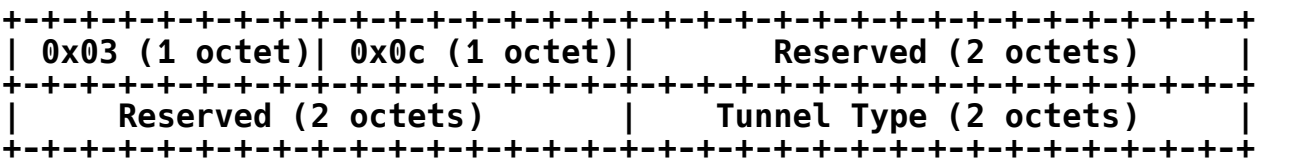


Figure 14: Encapsulation Extended Community

The value of the high-order octet of the extended Type field is 0x03, which indicates it's transitive. The value of the low-order octet of the extended Type field is 0x0c.

The last two octets of the Value field encode a tunnel type.

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; its semantics are the same as semantics of a Tunnel TLV in a Tunnel Encapsulation attribute, for which the following three conditions all hold:

1. It identifies the same tunnel type.

2. It has a Tunnel Egress 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 address of the Next Hop field of the route to which the Tunnel Encapsulation attribute is attached.

3. It has no other sub-TLVs.

Such a Tunnel TLV is called 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. In situations where a tunnel could be encoded using a barebones TLV, it MUST be encoded using the corresponding Encapsulation Extended Community. Notwithstanding, an implementation MUST be prepared to process a tunnel received encoded as a barebones TLV.

Note that for tunnel types of the form "X-in-Y" (for example, 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". Packets with other payload types MUST NOT be carried through such tunnels. See also Section 2.

In the remainder of this specification, when a route is referred to as containing a Tunnel Encapsulation attribute with a TLV identifying a particular tunnel type, it implicitly includes the case where the route contains an Encapsulation Extended Community identifying that tunnel type.

4.2. Router's MAC Extended Community

[EVPN-INTER-SUBNET] defines a router's MAC Extended Community. This extended community, as its name implies, carries the MAC address of the advertising router. Since the VXLAN and NVGRE Encapsulation sub-TLVs can also optionally carry a router's MAC, a conflict can arise if both the Router's MAC Extended Community and such an Encapsulation sub-TLV are present at the same time but have different values. In case of such a conflict, the information in the Router's MAC Extended Community MUST be used.

4.3. Color Extended Community

The Color Extended Community is a Transitive Opaque Extended Community with the encoding shown in Figure 15.

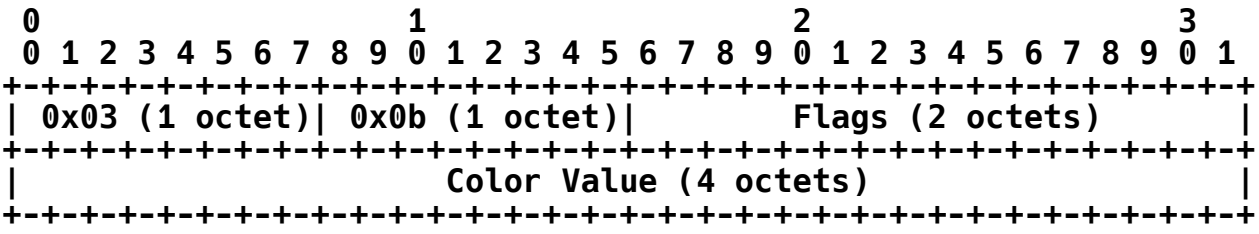


Figure 15: Color Extended Community

The value of the high-order octet of the extended Type field is 0x03, which indicates it is transitive. The value of the low-order octet of the extended Type field for this community is 0x0b. The color value is user defined and configured locally. No flags are defined in this document; this field MUST be set to zero by the originator and ignored by the receiver; the value MUST NOT be changed when propagating this extended community. The Color Value field is encoded as a 4-octet value by the administrator and is outside the scope of this document. For the use of this extended community, please see Section 8.

5. Special Considerations for IP-in-IP Tunnels

In certain situations with an IP fabric underlay, one could have a tunnel overlay with the tunnel type IP-in-IP. The egress BGP speaker can advertise the IP-in-IP tunnel endpoint address in the Tunnel Egress Endpoint sub-TLV. When the tunnel type of the TLV is IP-in-IP, it will not have a virtual network identifier. However, the tunnel egress endpoint address can be used in identifying the forwarding table to use for making the forwarding decisions to forward the payload.

6. Semantics and Usage of the Tunnel Encapsulation Attribute

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.

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. (This also applies if the Encapsulation Extended Community encoding is used.)

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.

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 four properties:

  1. The tunnel type is supported (that is, router R knows how to set up tunnels of that type, how to create the encapsulation header for tunnels of that type, etc.).

  2. The tunnel is of a type that can be used to carry packet P (for example, an MPLS-in-UDP tunnel would not be a feasible tunnel for carrying an IP packet, unless the IP packet can first be encapsulated in a MPLS packet).

  3. The tunnel is specified in a TLV whose Tunnel Egress Endpoint sub-TLV identifies an IP address that is reachable. The reachability condition is evaluated as per [RFC4271]. If the IP address is reachable via more than one forwarding table, local policy is used to determine which table to use.

  4. There is no local policy that prevents the use of the tunnel.

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 (that is, if it specifies several feasible 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. Many other factors, for example, minimization of encapsulation-header overhead, could also be used to influence selection.

The reachability to the address of the egress endpoint of the tunnel may change over time, directly impacting the feasibility of the tunnel. A tunnel that is not feasible at some moment may become

feasible at a later time when its egress endpoint address is reachable. The router may start using the newly feasible tunnel instead of an existing one. How this decision is made is outside the scope of this document.

Once it is determined to send a packet through the tunnel specified in a particular Tunnel TLV of a particular Tunnel Encapsulation attribute, then the tunnel's egress endpoint address is the IP address contained in the Tunnel Egress Endpoint sub-TLV. If the Tunnel TLV contains a Tunnel Egress Endpoint sub-TLV whose Value field is all zeroes, then the tunnel's egress endpoint is the address of the next hop of the BGP UPDATE containing the Tunnel Encapsulation attribute (that is, the Network Address of Next Hop field of the MP_REACH_NLRI attribute if the encoding of [RFC4760] is in use or the NEXT_HOP attribute otherwise). 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 (for example, by provisioning) of how to fill in the various fields in the encapsulation header.

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 Egress 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 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.

See also Section 13, which provides further specification regarding validation and exception cases.

7. Routing Considerations

7.1. Impact on the BGP Decision Process

The presence of the Tunnel Encapsulation attribute affects the BGP best route-selection algorithm. If a route includes the Tunnel Encapsulation attribute, and if that attribute includes no tunnel that is feasible, then that route MUST NOT be considered resolvable for the purposes of the route resolvability condition ([RFC4271], Section 9.1.2.1).

7.2. Looping, Mutual Recursion, 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 egress endpoint of X. It is easy to see that this can have no good outcome. [RFC4271] describes an analogous case as mutually recursive routes.

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 that such an attack will result in unresolvable BGP routes due to the mutually recursive relationship. 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 advertisement. 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 prevent any given packet from looping infinitely.

During the deployment of techniques described in this document, operators are encouraged to avoid mutually recursive route and/or tunnel dependencies. There is greater potential for such scenarios to arise when the tunnel egress endpoint for a given prefix differs from the address of the next hop for that prefix.

8. 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 address of the next hop of UPDATE U1 is router R2;

* the best route to router R2 is a BGP route that was advertised in UPDATE U2; and

* 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 6 for further details.

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

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

9. 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.

9.1. 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. When a packet is sent through a tunnel specified in one of the attribute's TLVs, and that tunnel type does not contain a Virtual Network Identifier field, the label or labels from the NLRI are pushed on the packet's label stack. The resulting MPLS packet is then further encapsulated, as specified by the TLV.

9.2. Tunnel Types with a Virtual Network Identifier Field

Two 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 encapsulation, this field is called the VNI (VXLAN 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 in 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 or NVGRE.

9.2.1. Unlabeled Address Families

This subsection applies when:

* the Tunnel Encapsulation attribute is carried in a BGP UPDATE of an unlabeled address family,

* 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 to 1, 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.

9.2.2. Labeled Address Families

This subsection applies when:

* the Tunnel Encapsulation attribute is carried in a BGP UPDATE of a labeled address family,

* 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.

9.2.2.1. When a Valid VNI Has Been Signaled

If the TLV identifying the tunnel contains an Encapsulation sub-TLV whose V bit is set to 1, 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. However, the Embedded

Label Handling sub-TLV will determine label processing as described below.

* If the TLV contains an Embedded Label Handling sub-TLV whose value is 1, 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 it contains an Embedded Label Handling sub-TLV whose value is 2, the embedded label is ignored entirely.

9.2.2.2. 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 to 1, 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 lower 3 octets of 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.

10. 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 represented by the address of the next hop. Certain of the procedures of Sections 9.2.2.1 or 9.2.2.2 cause the embedded label to be carried by a data packet to the router whose address appears in the Tunnel Egress Endpoint sub-TLV. If the Tunnel Egress Endpoint sub-TLV does not identify the same router represented by the address of 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. Avoidance of this unfortunate outcome is a matter of network planning and design and is outside the scope of this document.

Note that if the Tunnel Encapsulation attribute is attached to a VPN-IP route [RFC4364], if Inter-AS "option b" (see Section 10 of [RFC4364]) is being used, and if the Tunnel Egress Endpoint sub-TLV

contains an IP address that is not in the 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 recommended.

Other documents may define other ways to signal tunnel information in BGP. For example, [RFC6514] defines the "P-Multicast Service Interface Tunnel" (PMSI Tunnel) attribute. In this specification, we do not consider the effects of advertising the Tunnel Encapsulation attribute in conjunction with other forms of signaling tunnels. Any document specifying such joint use MUST provide details as to how interactions should be handled.

11. 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, the Tunnel Encapsulation attribute MUST be used only within a well-defined scope, for example, within a set of ASes 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 distributed. This filtering SHOULD be possible on a per-BGP-session basis; finer granularities (for example, per route and/or per attribute TLV) MAY be supported. For each external BGP (EBGP) 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 EBGP session, filtering of the attribute on outgoing UPDATEs MUST be enabled by default.

Since the Encapsulation Extended Community provides a subset of the functionality of the Tunnel Encapsulation attribute, these considerations apply equally in its case:

* Any BGP speaker that understands it MUST be able to filter it from incoming BGP UPDATE messages.

* It MUST be possible to filter the Encapsulation Extended Community from outgoing messages.

* In both cases, this filtering MUST be enabled by default for EBGP sessions.

12. Operational Considerations

A potential operational difficulty arises when tunnels are used, if the size of packets entering the tunnel exceeds the maximum

transmission unit (MTU) the tunnel is capable of supporting. This difficulty can be exacerbated by stacking multiple tunnels, since each stacked tunnel header further reduces the supportable MTU. This issue is long-standing and well-known. The tunnel signaling provided in this specification does nothing to address this issue, nor to aggravate it (except insofar as it may further increase the popularity of tunneling).

13. Validation and 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, and the "Treat-as-withdraw" procedure of [RFC7606] is applied.

If a Tunnel Encapsulation attribute does not have any valid TLVs, or it does not have the transitive bit set, the "Treat-as-withdraw" 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, it 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.

The following sub-TLVs defined in this document MUST NOT occur more than once in a given Tunnel TLV: Tunnel Egress Endpoint (discussed below), Encapsulation, DS, UDP Destination Port, Embedded Label Handling, MPLS Label Stack, and 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 disregarded. 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 and 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.

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 sub-TLV MUST remain in the attribute.

In general, if a TLV contains a sub-TLV that is malformed, the sub-TLV MUST be treated as if it were an unrecognized sub-TLV. There is one exception to this rule: if a TLV contains a malformed Tunnel

Egress 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 distributed.

Within a Tunnel Encapsulation attribute that is carried by a BGP UPDATE whose AFI/SAFI is one of those explicitly listed in the first paragraph of Section 6, a TLV that does not contain exactly one Tunnel Egress Endpoint sub-TLV MUST be treated as if it contained a malformed Tunnel Egress 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, or a tunnel of the form "X-in-Y" contains a Protocol Type sub-TLV that specifies something other than "X". Sub-TLVs of this sort MUST be disregarded. 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 distributed. An implementation MAY log a message when it encounters such a sub-TLV.

14. IANA Considerations

IANA has made the updates described in the following subsections. All registration procedures listed are per their definitions in [RFC8126].

14.1. Obsoleting RFC 5512

Because this document obsoletes RFC 5512, IANA has updated references to RFC 5512 to point to this document in the following registries:

* "Border Gateway Protocol (BGP) Extended Communities" registry [IANA-BGP-EXT-COMM]

* "Border Gateway Protocol (BGP) Parameters" registry [IANA-BGP-PARAMS]

* "Border Gateway Protocol (BGP) Tunnel Encapsulation" registry [IANA-BGP-TUNNEL-ENCAP]

* "Subsequent Address Family Identifiers (SAFI) Parameters" registry [IANA-SAFI]

14.2. Obsoleting Code Points Assigned by RFC 5566

Since this document obsoletes RFC 5566, the code points assigned by that RFC are similarly obsoleted. Specifically, the following code points have been marked as deprecated.

In the "BGP Tunnel Encapsulation Attribute Tunnel Types" registry [IANA-BGP-TUNNEL-ENCAP]:

Value	Name

3	Transmit tunnel endpoint (DEPRECATED)
4	IPsec in Tunnel-mode (DEPRECATED)
5	IP in IP tunnel with IPsec Transport Mode (DEPRECATED)
6	MPLS-in-IP tunnel with IPsec Transport Mode (DEPRECATED)

Table 1

And in the "BGP Tunnel Encapsulation Attribute Sub-TLVs" registry [IANA-BGP-TUNNEL-ENCAP]:

Value	Name
3	IPsec Tunnel Authenticator (DEPRECATED)

Table 2

14.3. Border Gateway Protocol (BGP) Tunnel Encapsulation Grouping

IANA has created a new registry grouping named "Border Gateway Protocol (BGP) Tunnel Encapsulation" [IANA-BGP-TUNNEL-ENCAP].

14.4. BGP Tunnel Encapsulation Attribute Tunnel Types

IANA has relocated the "BGP Tunnel Encapsulation Attribute Tunnel Types" registry to be under the "Border Gateway Protocol (BGP) Tunnel Encapsulation" grouping [IANA-BGP-TUNNEL-ENCAP].

14.5. Subsequent Address Family Identifiers

IANA has modified the "SAFI Values" registry [IANA-SAFI] to indicate that the Encapsulation SAFI (value 7) has been obsoleted. This document is listed as the reference for this change.

14.6. BGP Tunnel Encapsulation Attribute Sub-TLVs

IANA has relocated the "BGP Tunnel Encapsulation Attribute Sub-TLVs" registry to be under the "Border Gateway Protocol (BGP) Tunnel Encapsulation" grouping [IANA-BGP-TUNNEL-ENCAP].

IANA has included the following note to the 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 to 255 (inclusive), the Sub-TLV Length field contains two octets.

IANA has updated the registration procedures of the registry to the following:

Range	Registration Procedures
1-63	Standards Action
64-125	First Come First Served
126-127	Experimental Use
128-191	Standards Action
192-252	First Come First Served
253-254	Experimental Use

Table 3

IANA has added the following entries to this registry:

Value	Description	Reference
0	Reserved	RFC 9012
6	Tunnel Egress Endpoint	RFC 9012
7	DS Field	RFC 9012
8	UDP Destination Port	RFC 9012
9	Embedded Label Handling	RFC 9012
10	MPLS Label Stack	RFC 9012
11	Prefix-SID	RFC 9012
255	Reserved	RFC 9012

Table 4

14.7. Flags Field of VXLAN Encapsulation Sub-TLV

IANA has created a registry named "Flags Field of VXLAN Encapsulation Sub-TLVs" under the "Border Gateway Protocol (BGP) Tunnel Encapsulation" grouping [IANA-BGP-TUNNEL-ENCAP]. The registration policy for this registry is "Standards Action". The minimum possible value is 0, and the maximum is 7.

The initial values for this new registry are indicated in Table 5.

Bit Position	Description	Reference
0	V (VN-ID)	RFC 9012

1	M (MAC Address)	RFC 9012

Table 5

14.8. Flags Field of NVGRE Encapsulation Sub-TLV

IANA has created a registry named "Flags Field of NVGRE Encapsulation Sub-TLVs" under the "Border Gateway Protocol (BGP) Tunnel Encapsulation" grouping [IANA-BGP-TUNNEL-ENCAP]. The registration policy for this registry is "Standards Action". The minimum possible value is 0, and the maximum is 7.

The initial values for this new registry are indicated in Table 6.

Bit Position	Description	Reference
0	V (VN-ID)	RFC 9012
1	M (MAC Address)	RFC 9012

Table 6

14.9. Embedded Label Handling Sub-TLV

IANA has created a registry named "Embedded Label Handling Sub-TLVs" under the "Border Gateway Protocol (BGP) Tunnel Encapsulation" grouping [IANA-BGP-TUNNEL-ENCAP]. The registration policy for this registry is "Standards Action". The minimum possible value is 0, and the maximum is 255.

The initial values for this new registry are indicated in Table 7.

Value	Description	Reference
0	Reserved	RFC 9012
1	Payload of MPLS with embedded label	RFC 9012
2	No embedded label in payload	RFC 9012

Table 7

14.10. Color Extended Community Flags

IANA has created a registry named "Color Extended Community Flags" under the "Border Gateway Protocol (BGP) Tunnel Encapsulation" grouping [IANA-BGP-TUNNEL-ENCAP]. The registration policy for this registry is "Standards Action". The minimum possible value is 0, and the maximum is 15.

This new registry contains columns for "Bit Position", "Description", and "Reference". No values have currently been registered.

15. Security Considerations

As Section 11 discusses, it is intended that the Tunnel Encapsulation attribute be used only within a well-defined scope, for example, within a set of ASes that belong to a single administrative entity. As long as the filtering mechanisms discussed in that section are applied diligently, an attacker outside the scope would not be able to use the Tunnel Encapsulation attribute in an attack. This leaves open the questions of attackers within the scope (for example, a compromised router) and failures in filtering that allow an external attack to succeed.

As [RFC4272] discusses, BGP is vulnerable to traffic-diversion attacks. The Tunnel Encapsulation attribute adds a new means by which an attacker could cause traffic to be diverted from its normal path, especially when the Tunnel Egress Endpoint sub-TLV is used. Such an attack would differ from pre-existing vulnerabilities in that traffic could be tunneled to a distant target across intervening network infrastructure, allowing an attack to potentially succeed more easily, since less infrastructure would have to be subverted. Potential consequences include "hijacking" of traffic (insertion of an undesired node in the path, which allows for inspection or modification of traffic, or avoidance of security controls) or denial of service (directing traffic to a node that doesn't desire to receive it).

In order to further mitigate the risk of diversion of traffic from its intended destination, Section 3.1.1 provides an optional procedure to check that the destination given in a Tunnel Egress Endpoint sub-TLV is within the AS that was the source of the route. 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. As RFC 4272 discusses, it's possible for an attacker to announce an inaccurate AS_PATH; therefore, an attacker with the ability to inject a Tunnel Egress Endpoint sub-TLV could equally craft an AS_PATH that would pass the validation procedures of Section 3.1.1. BGP origin validation [RFC6811] and BGPsec [RFC8205] provide means to increase assurance that the origins being validated have not been falsified.

Many tunnels carry traffic that embeds a destination address that comes from a non-global namespace. One example is MPLS VPNs. If a tunnel crosses from one namespace to another, without the necessary translation being performed for the embedded address(es), there exists a risk of misdelivery of traffic. If the traffic contains confidential data that's not otherwise protected (for example, by end-to-end encryption), then confidential information could be revealed. The restriction of applicability of the Tunnel Encapsulation attribute to a well-defined scope limits the likelihood of this occurring. See the discussion of "option b" in Section 10 for further discussion of one such scenario.

RFC 8402 specifies that "SR domain boundary routers MUST filter any

external traffic" ([RFC8402], Section 8.1). For these purposes, traffic introduced into an SR domain using the Prefix-SID sub-TLV lies within the SR domain, even though the Prefix-SIDs used by the routers at the two ends of the tunnel may be different, as discussed in Section 3.7. This implies that the duty to filter external traffic extends to all routers participating in such tunnels.

16. References

16.1. Normative References

[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>.

[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>.

[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>.

[RFC3270] Le Faucheur, F., Wu, L., Davie, B., Davari, S., Vaananen, P., Krishnan, R., Cheval, P., and J. Heinanen, "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services", RFC 3270, DOI 10.17487/RFC3270, May 2002, <https://www.rfc-editor.org/info/rfc3270>.

[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>.

[RFC5129] Davie, B., Briscoe, B., and J. Tay, "Explicit Congestion Marking in MPLS", RFC 5129, DOI 10.17487/RFC5129, January 2008, <https://www.rfc-editor.org/info/rfc5129>.

[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>.

[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>.

[RFC6890] Cotton, M., Vegoda, L., Bonica, R., Ed., and B. Haberman, "Special-Purpose IP Address Registries", BCP 153, RFC 6890, DOI 10.17487/RFC6890, April 2013, <https://www.rfc-editor.org/info/rfc6890>.

[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>.

[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>.

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

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

[RFC8669] Previdi, S., Filsfils, C., Lindem, A., Ed., Sreekantiah, A., and H. Gredler, "Segment Routing Prefix Segment Identifier Extensions for BGP", RFC 8669, DOI 10.17487/RFC8669, December 2019, <https://www.rfc-editor.org/info/rfc8669>.

16.2. Informative References

[EVPN-INTER-SUBNET]
Sajassi, A., Salam, S., Thoria, S., Drake, J. E., and J. Rabadan, "Integrated Routing and Bridging in EVPN", Work in Progress, Internet-Draft, draft-ietf-bess-evpn-inter-subnet-forwarding-13, 10 February 2021, <https://tools.ietf.org/html/draft-ietf-bess-evpn-inter-subnet-forwarding-13>.

[IANA-ADDRESS-FAM]
IANA, "Address Family Numbers", <https://www.iana.org/assignments/address-family-numbers/>.

[IANA-BGP-EXT-COMM]
IANA, "Border Gateway Protocol (BGP) Extended Communities", <https://www.iana.org/assignments/bgp-extended-communities/>.

[IANA-BGP-PARAMS]
IANA, "Border Gateway Protocol (BGP) Parameters", <https://www.iana.org/assignments/bgp-parameters/>.

[IANA-BGP-TUNNEL-ENCAP]
IANA, "Border Gateway Protocol (BGP) Tunnel Encapsulation", <https://www.iana.org/assignments/bgp-tunnel-encapsulation/>.

[IANA-ETHERTYPES]
IANA, "IEEE 802 Numbers: ETHER TYPES", <https://www.iana.org/assignments/ieee-802-numbers/>.

[IANA-SAFI]
IANA, "Subsequent Address Family Identifiers (SAFI) Parameters", <https://www.iana.org/assignments/safi-namespace/>.

[RFC4272] Murphy, S., "BGP Security Vulnerabilities Analysis", RFC 4272, DOI 10.17487/RFC4272, January 2006, <https://www.rfc-editor.org/info/rfc4272>.

[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>.

[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>.

[RFC5565] Wu, J., Cui, Y., Metz, C., and E. Rosen, "Softwire Mesh Framework", RFC 5565, DOI 10.17487/RFC5565, June 2009, <https://www.rfc-editor.org/info/rfc5565>.

[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>.

[RFC5640] Filsfils, C., Mohapatra, P., and C. Pignataro, "Load-Balancing for Mesh Softwires", RFC 5640, DOI 10.17487/RFC5640, August 2009, <https://www.rfc-editor.org/info/rfc5640>.

[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>.

[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>.

[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>.

[RFC8277] Rosen, E., "Using BGP to Bind MPLS Labels to Address Prefixes", RFC 8277, DOI 10.17487/RFC8277, October 2017, <https://www.rfc-editor.org/info/rfc8277>.

[RFC8365] Sajassi, A., Ed., Drake, J., Ed., Bitar, N., Shekhar, R., Uttaro, J., and W. Henderickx, "A Network Virtualization Overlay Solution Using Ethernet VPN (EVPN)", RFC 8365, DOI 10.17487/RFC8365, March 2018, <https://www.rfc-editor.org/info/rfc8365>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

Appendix A. Impact on RFC 8365

[RFC8365] references RFC 5512 for its definition of the BGP Encapsulation Extended Community. That extended community is now defined in this document, in a way consistent with its previous definition.

Section 6 of [RFC8365] talks about the use of the Encapsulation Extended Community to allow Network Virtualization Edge (NVE) devices to signal their supported encapsulations. We note that with the introduction of this specification, the Tunnel Encapsulation attribute can also be used for this purpose. For purposes where RFC 8365 talks about "advertising supported encapsulations" (for example, in the second paragraph of Section 6), encapsulations advertised using the Tunnel Encapsulation attribute should be considered equally with those advertised using the Encapsulation Extended Community.

In particular, a review of Section 8.3.1 of [RFC8365] is called for, to consider whether the introduction of the Tunnel Encapsulation attribute creates a need for any revisions to the split-horizon procedures.

[RFC8365] also refers to a draft version of this specification in the final paragraph of Section 5.1.3. That paragraph references Section 8.2.2.2 of the draft. In this document, the correct reference would be Section 9.2.2.2. There are no substantive differences between the section in the referenced draft version and that in this document.

Acknowledgments

This document contains text from RFC 5512, authored by Pradosh Mohapatra and Eric Rosen. The authors of the current document wish to thank them for their contribution. RFC 5512 itself built upon prior work by Gargi Nalawade, Ruchi Kapoor, Dan Tappan, David Ward, Scott Wainner, Simon Barber, Lili Wang, and Chris Metz, whom the authors also thank for their contributions. Eric Rosen was the principal author of earlier versions of this document.

The authors wish to thank Lou Berger, Ron Bonica, Martin Djernaes, John Drake, Susan Hares, Satoru Matsushima, Thomas Morin, Dhananjaya Rao, Ravi Singh, Harish Sitaraman, Brian Trammell, Xiaohu Xu, and Zhaohui Zhang for their review, comments, and/or helpful discussions. Alvaro Retana provided an especially comprehensive review.

Contributors

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

Randy Bush
Internet Initiative Japan
5147 Crystal Springs
Bainbridge Island, WA 98110
United States of America

Email: randy@psg.com

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

Email: robert@raszuk.net

Eric C. Rosen

Authors' Addresses

Keyur Patel

Arrcus, Inc
2077 Gateway Pl
San Jose, CA 95110
United States of America

Email: keyur@arrcus.com

Gunter Van de Velde
Nokia
Copernicuslaan 50
2018 Antwerpen
Belgium

Email: gunter.van_de_velde@nokia.com

Srihari R. Sangli
Juniper Networks

Email: ssangli@juniper.net

John Scudder
Juniper Networks

Email: jgs@juniper.net