IDR Working Group
Internet-Draft
Obsoletes: 5512, 5566, 5640 (if approved)
Intended status: Standards Track
Expires: January 18, 2021

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

July 17, 2020

The BGP Tunnel Encapsulation Attribute
draft-ietf-idr-tunnel-encaps-17

Abstract

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

This document obsoletes RFC 5512. Since RFCs 5566 and 5640 rely on RFC 5512, they are likewise obsoleted.

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 18, 2021.

Table of Contents

1. Introduction

This document obsoletes RFC 5512. The deficiencies of RFC 5512, and a summary of the changes made, are discussed in Sections 1.1-1.3. The material from RFC 5512 that is retained has been incorporated into this document. Since [RFC5566] and [RFC5640] rely on RFC 5512, they are 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, e.g., the "Encapsulation sub-TLV", contain information that may be used to form the encapsulation header for the specified Tunnel Type. Other sub-TLVs, e.g., the "color sub-TLV" and the "protocol sub-TLV", contain information that aids in determining whether particular packets should be sent through the tunnel that the TLV identifies.

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

- BGP speaker R1 has received and 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;
- 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 paths to two different address prefixes have the same next hop, [RFC5512] does not provide a straightforward method to associate each prefix with a different tunnel.

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

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

1.3. Brief Summary of Changes from RFC 5512

This document addresses these deficiencies 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]: VXLAN (Virtual Extensible Local Area Network, [RFC7348]), VXLAN GPE (Generic Protocol Extension for VXLAN, [I-D.ietf-nvo3-vxlan-gpe]), NVGRE (Network Virtualization Using Generic Routing Encapsulation [RFC7637]), and MPLS-in-GRE (MPLS in Generic Routing Encapsulation [RFC4023]). MPLS-in-UDP [RFC7510] is also supported, but an Encapsulation sub-TLV for it is not needed.

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

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

[RFC5512] defines a Tunnel Encapsulation Extended Community that can be used instead of the Tunnel Encapsulation attribute under certain circumstances. This document describes (Section 4.1) how the Tunnel Encapsulation Extended Community can be used in a backwards-compatible fashion. It is possible to combine Tunnel Encapsulation Extended Communities and Tunnel Encapsulation attributes in the same BGP UPDATE in this manner.

1.4. 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, 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. The Tunnel Encapsulation attribute MAY be used 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)).

In a given BGP update, the encapsulation information is specified in 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, e.g., communities or extended communities, may also be included.

2. The Tunnel Encapsulation Attribute

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

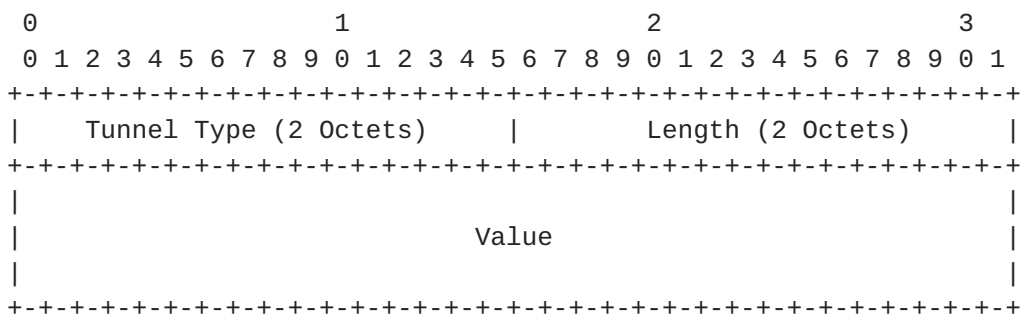

Figure 1: Tunnel Encapsulation TLV Value Field

- Tunnel Type (2 octets): identifies a type of tunnel. The field contains values from the IANA Registry "BGP Tunnel Encapsulation Attribute Tunnel Types". 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 field (depending on the type), and zero or more octets of value. A sub-TLV is structured as shown in Figure 2:

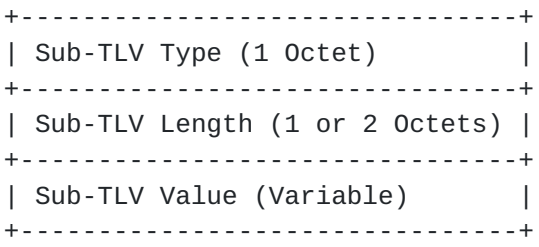

Figure 2: Encapsulation Sub-TLV Value Field

- 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".

- Sub-TLV Length (1 or 2 octets): the total number of octets of the sub-TLV value field. The Sub-TLV Length field contains 1 octet if the Sub-TLV Type field contains a value in the range from 0-127. The Sub-TLV Length field contains two octets if the Sub-TLV Type field contains a value in the range from 128-255.

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

3. Tunnel Encapsulation Attribute Sub-TLVs

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

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.

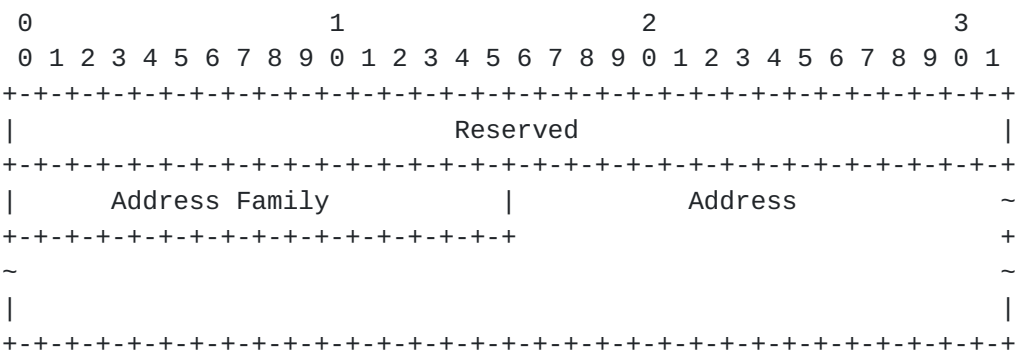

Figure 3: Tunnel Egress Endpoint Sub-TLV Value Field

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. 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 above, each TLV MUST have one, and one only, 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).

If the Address Family subfield has any value other than IPv4 or IPv6, the Tunnel Egress Endpoint sub-TLV is considered "unrecognized" (see Section 12). 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 plus 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.
  - 0, if the Address Family subfield contains the value zero.
- The IP address in the sub-TLV's address subfield is listed in the relevant Special-Purpose IP Address Registry [RFC6890] as either not a valid destination, or not forwardable.
- It can be determined according to the procedures below (Section 3.1.1) 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.

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

Error Handling is detailed in Section 12.

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 Field

This section details a procedure that MAY be applied to validate that when traffic is sent to the IP address depicted in the Address Field, it will go to the same AS as it would go to if the Tunnel Encapsulation Attribute were not present. See Section 13 for discussion of the limitations of this procedure.

The Route Origin ASN (Autonomous System Number) 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 Field of the Tunnel Egress Endpoint sub-TLV, it is necessary to determine the forwarding route, that is, the route installed in the Forwarding Information Base that will be used to forward traffic toward that address. The Address Field's Route Origin ASN is the Route Origin ASN of that route, or the distinguished value "NONE2" if the forwarding route has no AS Path, for example if that route's source is a protocol other than BGP. (Note that this is a distinct case from a route that has an empty AS Path.) 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 Autonomous Systems 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 not to be valid.

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

This section defines Encapsulation sub-TLVs for the following tunnel types: VXLAN ([RFC7348]), VXLAN GPE ([I-D.ietf-nvo3-vxlan-gpe]), 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 Section 6 and Section 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

This document defines an Encapsulation sub-TLV for VXLAN tunnels. When the Tunnel Type is VXLAN (value 8), the length of the sub-TLV is 12 octets. The following is the structure of the value field in the Encapsulation sub-TLV:

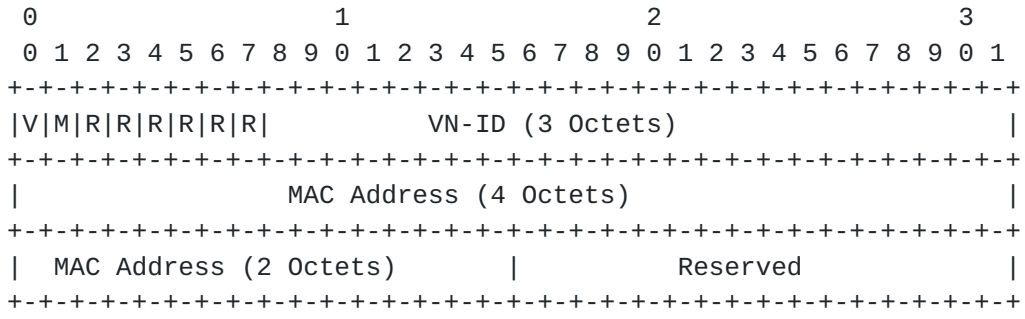

Figure 4: VXLAN Encapsulation Sub-TLV

V: This bit is set to 1 to indicate that a VN-ID (Virtual Network Identifier) is present in the Encapsulation sub-TLV. If set to 0, the VN-ID field is disregarded. Please see Section 8.

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 a receipt of the sub-TLV.

VN-ID: If the V bit is set, the VN-ID field contains a 3 octet VN-ID value. If the V bit is not set, the VN-ID field MUST be set to zero on transmission and disregarded on receipt.

MAC Address: If the M bit is set, this field contains a 6 octet Ethernet MAC address. If the M bit is not set, this field MUST be set to all zeroes 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, the MAC Address is copied into the Inner Destination MAC Address field of the Inner Ethernet Header (see section 5 of [RFC7348]).

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

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

- If the V bit is not set, and the BGP UPDATE message has AFI/SAFI other than Ethernet VPNs (EVPN) then the VXLAN tunnel cannot be used.

- Section 8 describes how the VNI field of the VXLAN encapsulation header is set.

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

3.2.2. VXLAN GPE

This document defines an Encapsulation sub-TLV for VXLAN GPE tunnels. When the Tunnel Type is VXLAN GPE (value 12), the length of the sub-TLV is 8 octets and following is the structure of the value field in the Encapsulation sub-TLV:

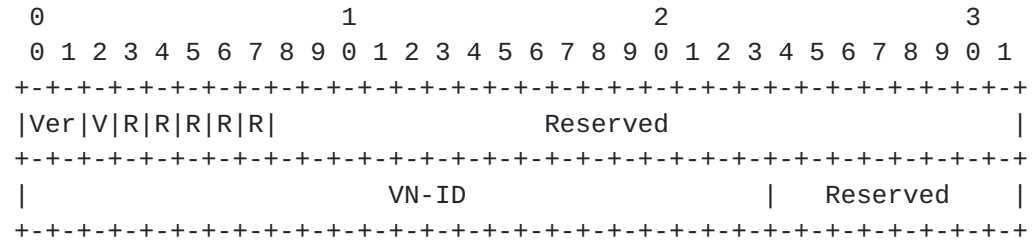

Figure 5: VXLAN GPE Encapsulation Sub-TLV

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

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 8.

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

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

Reserved (two fields): MUST be set to zero on transmission and disregarded on receipt.

When forming the VXLAN GPE encapsulation header:

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

- Section 8 describes how the VNI field of the VXLAN GPE encapsulation header is set.

3.2.3. NVGRE

This document defines an Encapsulation sub-TLV for NVGRE tunnels. When the Tunnel Type is NVGRE (value 9), the length of the sub-TLV is 12 octets. The following is the structure of the value field in the Encapsulation sub-TLV:

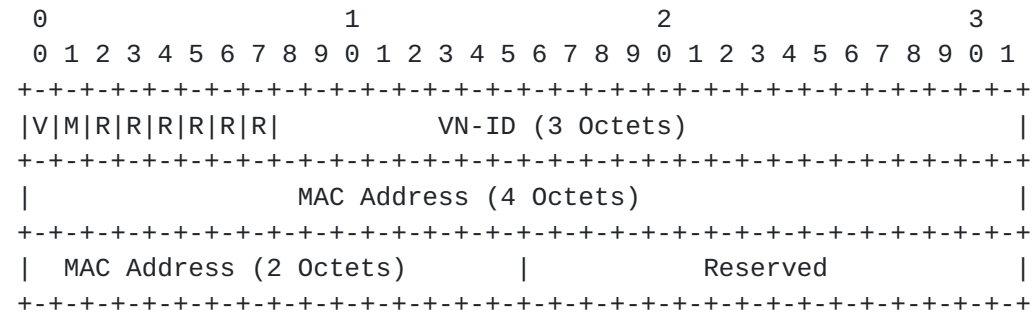

Figure 6: NVGRE Encapsulation Sub-TLV

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 8.

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, the VN-ID field contains a 3 octet VN-ID value. If the V bit is not set, the VN-ID field MUST be set to zero on transmission and disregarded on receipt.

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

Reserved (two fields): 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 VXLAN header is set as per [RFC7637].

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

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

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

- If the V bit is not set, and the BGP UPDATE message has AFI/SAFI other than Ethernet VPNs (EVPN) then the NVGRE tunnel cannot be used.

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

3.2.4. L2TPv3

When the Tunnel Type of the TLV is L2TPv3 over IP (value 1), the length of the sub-TLV is between 4 and 12 octets, depending on the length of the cookie. The following is the structure of the value field of the Encapsulation sub-TLV:

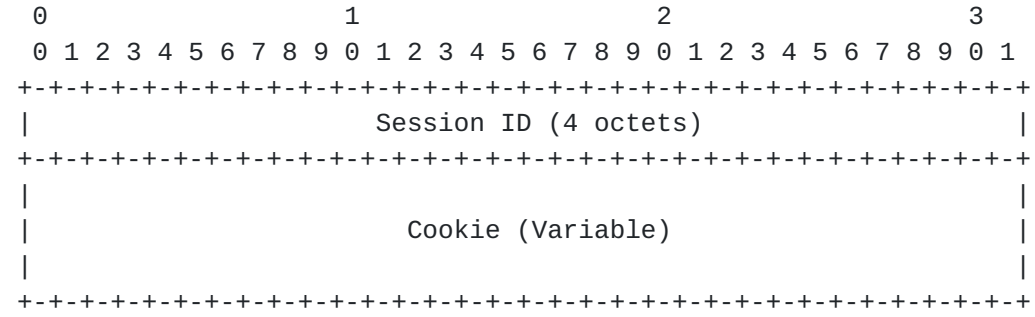

Figure 7: L2TPv3 Encapsulation Sub-TLV

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 octets -- 0 to 8 octets) value used by L2TPv3 to check the association of a received data message with the session identified by the Session ID. Generation and usage of the cookie value is as specified in [RFC3931].

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

3.2.5. GRE

When the Tunnel Type of the TLV is GRE (value 2), the length of the sub-TLV is 4 octets. The following is the structure of the value field of the Encapsulation sub-TLV:

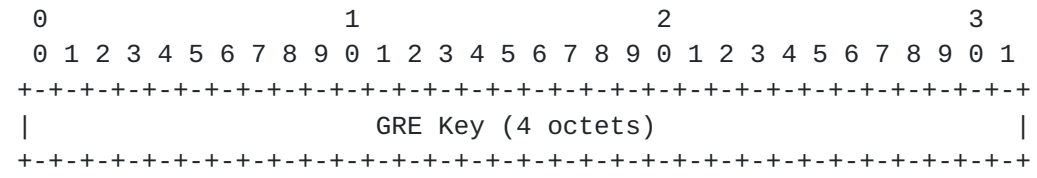

Figure 8: GRE Encapsulation Sub-TLV

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.

3.2.6. MPLS-in-GRE

When the Tunnel Type is MPLS-in-GRE (value 11), the length of the sub-TLV is 4 octets. The following is the structure of the value field of the Encapsulation sub-TLV:

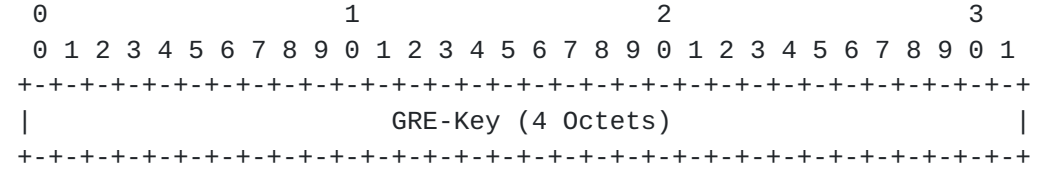

Figure 9: MPLS-in-GRE Encapsulation Sub-TLV

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 3.2.5 can be used instead of the MPLS-in-GRE Tunnel Type when it is necessary to encapsulate MPLS in GRE. Including a TLV of the MPLS-in-GRE tunnel type is equivalent to including a TLV of the GRE Tunnel Type that also includes a Protocol Type sub-TLV (Section 3.4.1) specifying MPLS as the protocol to be encapsulated.

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

3.3. 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 unknown type of sub-TLV.

3.3.1. DS Field

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, whose type code is 7, 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]). The value field is always a single octet.

3.3.2. UDP Destination Port

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

type code is 8. The value field of this sub-TLV is always a two-octet field, containing the port value.

3.4. Sub-TLVs for Aiding Tunnel Selection

3.4.1. Protocol Type Sub-TLV

The Protocol Type sub-TLV, whose type code is 2, 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 [Ethertypes].

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", e.g., "MPLS-in-GRE", only packets of the specified payload type "X" are to be carried through the tunnel of type "Y". This is the equivalent of specifying a Tunnel Type "Y" and including in its TLV a Protocol Type sub-TLV (see Section 3.4.1) specifying protocol "X". If the Tunnel Type is "X-in-Y", it is unnecessary, though harmless, to 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 12.

3.4.2. Color Sub-TLV

The Color sub-TLV, whose type code is 4, 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 7.

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

should be treated as if it were an unrecognized sub-TLV (see Section 12).

3.5. Embedded Label Handling Sub-TLV

Certain BGP address families (corresponding to particular AFI/SAFI pairs, e.g., 1/4, 2/4, 1/128, 2/128) have MPLS labels embedded in their NLRIs. 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 (e.g., VXLAN, VXLAN GPE, 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, whose type code is 9, 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 disregared. In those cases where the sub-TLV is ignored, it SHOULD 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 carried either in the virtual network identifier field of the encapsulation header, or else is ignored entirely.

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

3.6. MPLS Label Stack Sub-TLV

This sub-TLV, whose type code is 10, 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, the final entry in the sequence is the "bottommost" label. When this label stack is pushed onto a packet, this ordering MUST be preserved.

Each label stack entry has the following format:

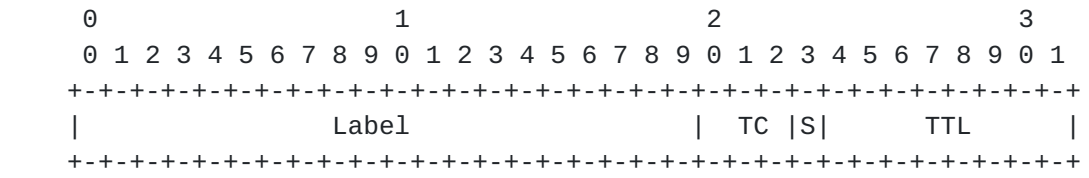

Figure 10: MPLS Label Stack Sub-TLV

The fields are as defined in [RFC3032], [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.

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

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

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

3.7. Prefix-SID Sub-TLV

[RFC8669] defines a BGP Path attribute known as the "Prefix-SID Attribute". This attribute is defined to contain a sequence of one or more TLVs, where each TLV is either a "Label-Index" TLV, or an "Originator SRGB (Source Routing Global Block)" TLV.

This document defines a Prefix-SID sub-TLV, whose type code is 11. 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 Prefix-SID Attribute is attached to routes of type IPv4/IPv6 Labeled Unicast ([RFC4760], [RFC8277]), and it only defines values of the Prefix-SID Attribute when attached to routes of those types. Therefore, similar limitations exist for the Prefix-SID sub-TLV: although it MAY be encoded in any BGP UPDATE message where the Tunnel Encapsulation attribute is allowed (see Section 5), the encoded information MUST be ignored just as the base specification that defines the encoding requires. So, in the case of the values specified in [RFC8669], they MUST be ignored if received with routes of type other than IPv4/IPv6 Labeled Unicast.

The Prefix-SID sub-TLV can occur in a TLV identifying any type of tunnel. If an Originator SRGB is specified in the sub-TLV, that SRGB

MUST be interpreted to be the SRGB used by the tunnel's egress endpoint. The Label-Index, if present, is the Segment Routing SID that the tunnel's egress endpoint uses to represent the prefix appearing in the NLRI field of the BGP UPDATE to which the Tunnel Encapsulation attribute is attached.

If a Label-Index is present in the Prefix-SID sub-TLV, then when a packet is sent through the tunnel identified by the TLV, the corresponding MPLS label MUST be pushed on the packet's label stack. The corresponding MPLS label is computed from the Label-Index value and the SRGB of the route's originator, 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 (e.g., a label embedded in UPDATE's NLRI, or a label determined by the procedures of Section 8, are pushed on the stack.

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

4. Extended Communities Related to the Tunnel Encapsulation Attribute

4.1. Encapsulation Extended Community

The Encapsulation Extended Community is a Transitive Opaque Extended Community.

The Encapsulation Extended Community encoding is as shown below

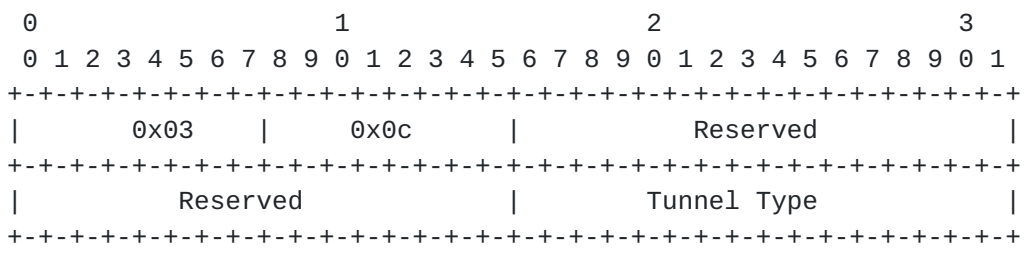

Figure 11: Encapsulation Extended Community

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 Encapsulation attribute Tunnel TLV 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.

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

type "Y". 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 a Tunnel Encapsulation Extended Community identifying that Tunnel Type.

4.2. Router's MAC Extended Community

[I-D.ietf-bess-evpn-inter-subnet-forwarding] defines a Router's MAC Extended Community. This Extended Community, 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 Encapsulation Sub-TLV MUST be used.

4.3. Color Extended Community

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

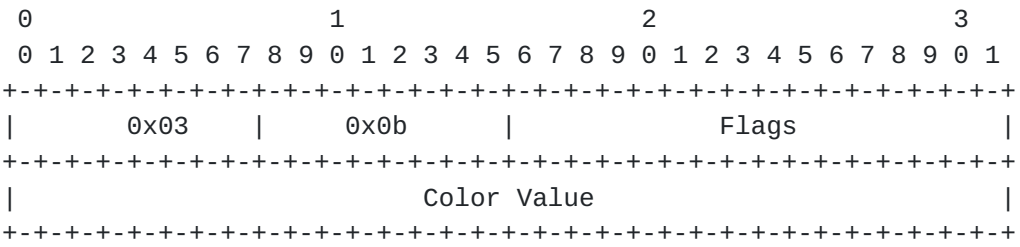

Figure 12: Color Extended Community

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 4 octet value by the administrator and is outside the scope of this document. For the use of this Extended Community please see Section 7.

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. See the second bullet point of Section 9.1 for further discussion.

6. Semantics and Usage of the Tunnel Encapsulation attribute

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

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

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

The 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 three properties:

* The Tunnel Type is supported (i.e., router R knows how to set up tunnels of that type, how to create the encapsulation header for tunnels of that type, etc.)

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

* The tunnel is specified in a TLV whose Tunnel Egress Endpoint sub-TLV identifies an IP address that is reachable. This IP address may be reachable via one or more forwarding tables. Local policy may determine these forwarding tables and is outside the scope of this document. The reachability condition is evaluated as per [RFC4271].

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

If the Tunnel Encapsulation attribute contains several TLVs (i.e., if it specifies several feasibile 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.

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 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. The address of the tunnel egress endpoint generally appears in a "destination address" field of the encapsulation.

The full set of procedures for sending a packet through a particular Tunnel Type to a particular tunnel egress endpoint depends upon the tunnel type, and is outside the scope of this document. Note that some tunnel types may require the execution of an explicit tunnel

setup protocol before they can be used for carrying data. Other tunnel types may not require any tunnel setup protocol.

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

If a Tunnel Encapsulation attribute specifies several tunnels, the way in which a router chooses which one to use is a matter of policy, In addition to the reachability to the address of the egress endpoint of the tunnel, other policy factors MAY be used to determine the feasibility of the tunnel. The policy factors are beyond the scope of this document.

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 is encapsulated according to the Tunnel Encapsulation attribute of that route. That is, tunnels may be "stacked".

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

See also Section 12, 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 which is feasible, then that route MUST NOT be considered resolvable for the purposes of 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 as 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 path to router R2 is a BGP route that was advertised in UPDATE U2;
- UPDATE U2 has a Tunnel Encapsulation attribute.

Then packet P MUST be sent through one of the tunnels identified in the Tunnel Encapsulation attribute of UPDATE U2. See Section 6 for further details.

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

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.

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

- If the TLV does not contain an Embedded Label Handling sub-TLV, or if it contains an Embedded Label Handling sub-TLV whose value is 2, the embedded label is ignored completely. In this case the tunnel encapsulation is presumed to provide complete information regarding the forwarding context required.

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

9.2. Tunnel Types with a Virtual Network Identifier Field

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

When one of these tunnel encapsulations is imposed on a packet, the setting of the virtual network identifier field in the encapsulation header depends upon the contents of the Encapsulation sub-TLV (if one is present). When the Tunnel Encapsulation attribute is being carried 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, VXLAN GPE, or NVGRE.

9.2.1. Unlabeled Address Families

This sub-section applies when:

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

- at least one of the attribute's TLVs identifies a Tunnel Type that uses a virtual network identifier, and
- it has been determined to send a packet through one of those tunnels.

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

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

9.2.2. Labeled Address Families

This sub-section applies when:

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

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, 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, 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 Section 9.2.2.1 or Section 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], and 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 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.

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, it is intended that the Tunnel Encapsulation attribute be used only within a well-defined scope, e.g., within a set of Autonomous Systems that belong to a single administrative entity. If the attribute is distributed beyond its intended scope, packets may be sent through tunnels in a manner that is not intended.

To prevent the Tunnel Encapsulation attribute from being distributed beyond its intended scope, any BGP speaker that understands the attribute MUST be able to filter the attribute from incoming BGP UPDATE messages. When the attribute is filtered from an incoming UPDATE, the attribute is neither processed nor 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.

12. 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, 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, 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. This document specifies 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 second 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.

13. IANA Considerations

This document makes the following requests of IANA. (All registration procedures listed below are per their definitions in [RFC8126].)

13.1. BGP Tunnel Encapsulation Parameters Grouping

Create a new registry grouping, to be named "BGP Tunnel Encapsulation Parameters".

13.2. Subsequent Address Family Identifiers

Modify the "Subsequent Address Family Identifiers" registry to indicate that the Encapsulation SAFI (value 7) is obsoleted. This document should be the reference.

Because this document obsoletes RFC 5512, change all registration information that references [RFC5512] to instead reference this document.

13.3. BGP Tunnel Encapsulation Attribute Sub-TLVs

Relocate the "BGP Tunnel Encapsulation Attribute Sub-TLVs" registry to be under the "BGP Tunnel Encapsulation Parameters" grouping.

Add 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 128-255 inclusive, the Sub-TLV Length field contains two octets.

Change the registration policy of the registry to the following:

Value(s)	Registration Procedure
0	Reserved
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
255	Reserved

Rename the following entries within the registry:

Value	Old Name	New Name
6	Remote Endpoint	Tunnel Egress Endpoint
7	IPv4 DS Field	DS Field

13.4. Flags Field of VXLAN Encapsulation sub-TLV

Create a registry named "Flags Field of VXLAN Encapsulation sub-TLV" under the "BGP Tunnel Encapsulation Parameters" grouping. The registration policy for this registry is "Standards Action".

The initial values for this new registry are indicated below.

Bit Position	Description	Reference
0	V (Virtual Network Identifier)	(this document)
1	M (MAC Address)	(this document)

13.5. Flags Field of VXLAN GPE Encapsulation sub-TLV

Create a registry named "Flags Field of VXLAN GPE Encapsulation sub-TLV" under the "BGP Tunnel Encapsulation Parameters" grouping. The registration policy for this registry is "Standards Action".

The initial value for this new registry is indicated below.

Bit Position	Description	Reference
0	V (VN-ID)	(this document)

13.6. Flags Field of NVGRE Encapsulation sub-TLV

Create a registry named "Flags Field of NVGRE Encapsulation sub-TLV" under the "BGP Tunnel Encapsulation Parameters" grouping. The registration policy for this registry is "Standards Action".

The initial values for this new registry are indicated below.

Bit Position	Description	Reference
0	V (VN-ID)	(this document)
1	M (MAC Address)	(this document)

13.7. Embedded Label Handling sub-TLV

Create a registry named "Embedded Label Handling sub-TLV" under the "BGP Tunnel Encapsulation Parameters" grouping. The registration policy for this registry is "Standards Action".

The initial values for this new registry are indicated below.

Value	Description	Reference
1	Payload of MPLS with embedded label	(this document)
2	no embedded label in payload	(this document)

13.8. Color Extended Community

Add this document as a reference for the "Color Extended Community" entry in the Transitive Opaque Extended Community Sub-Types registry.

13.9. Color Extended Community Flags

Create a registry named "Color Extended Community Flags" under the "BGP Tunnel Encapsulation Parameters" grouping. The registration policy for this registry is "Standards Action".

No initial values are to be registered. The format of the registry is shown below.

Bit Position	Description	Reference

14. Security Considerations

As Section 11 discusses, it is intended that the Tunnel Encapsulation attribute be used only within a well-defined scope, e.g., within a set of Autonomous Systems that belong to a single administrative entity. 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) 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.

15. 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, Satoru Matsushima, Dhananjaya Rao, Ravi Singh, Thomas Morin, Xiaohu Xu, and Zhaohui Zhang for their review, comments, and/or helpful discussions. Alvaro Retana provided an especially comprehensive review.

16. Contributor Addresses

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

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

Email: randy@psg.com

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

Email: robert@raszuk.net

Eric C. Rosen

17. References

17.1. Normative References

[I-D.ietf-nvo3-vxlan-gpe]
Maino, F., Kreeger, L., and U. Elzur, "Generic Protocol Extension for VXLAN", draft-ietf-nvo3-vxlan-gpe-09 (work in progress), December 2019.

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

[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>.

[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>.

[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>.

17.2. Informative References

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

[I-D.ietf-bess-evpn-inter-subnet-forwarding] Sajassi, A., Salam, S., Thoria, S., Drake, J., and J. Rabadan, "Integrated Routing and Bridging in EVPN", draft-ietf-bess-evpn-inter-subnet-forwarding-09 (work in progress), June 2020.

[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>.

[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>.

[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>.

[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>.

[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>.

Authors' Addresses

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

Email: keyur@arrcus.com

Gunter Van de Velde
Nokia
Copernicuslaan 50
Antwerpen 2018
Belgium

Email: gunter.van_de_velde@nokia.com

Srihari R. Sangli
Juniper Networks

Email: ssangli@juniper.net

John Scudder
Juniper Networks

Email: jgs@juniper.net