Abstract

Link Layer Discovery Protocol (LLDP) or IEEE 802.1AB is implemented in networking equipment from many vendors. It is natural for IETF protocols to avail this protocol for simple discovery tasks. This document describes how BGP would use LLDP to discover directly connected and 2-hop peers when peering is based on loopback addresses.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 9, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Introduction

Link Layer Discovery Protocol (LLDP) [LLDP] or IEEE 802.1AB is implemented in networking equipment from many vendors. It is natural for IETF protocols to avail this protocol for simple discovery tasks. This document describes how BGP [RFC4271] would use LLDP to discover directly connected and 2-hop peers when peering is based on loopback addresses.
1.1. Requirements Notation

1.1.1. Requirements Language

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.

2. LLDP Extensions

2.1. LLDP Organizationally Specific TLV Format

The format of the LLDP Basic Organizationally Specific TLV (OS-TLV) is defined in [LLDP]. It is shown below for completeness.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type (127)  |       Length    |  OUI (3 Octets) 00-00-5E      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| OUI Continued |  Subtype      |     Value                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             ...   (Up to 507 Octets)          |
```

Type    Organizationally Specific TLV type value, 127.

Length The length of the remainder of the TLV.

OUI     Organizationally unique identifier for the organization’s OUI. For IANA, this is value is 00-00-5E as specified in [IEEE-802-IANA].

Subtype IETF specific subtype

Value   Value for organizationally specific TLV. The Length of the value is 4 octets less than the TLV length.

LLDP Organizationally Specific TLV

The OUI for IANA was allocated in section 1.4 of [RFC7042]. This document requests creation of a registry for IETF specific sub-types for LLDP Organizationally Specific TLVs.
2.2. BGP Config OS-TLV Format

The BGP Config Organizationally Specific TLV (OS-TLV) will be used to advertise BGP configuration information. The configuration information will be composed of Sub-TLVs. Since the length is limited to 507 octets, multiple BGP Config OS-TLVs could be included in a single LLDP advertisement.

```
<table>
<thead>
<tr>
<th>Type (127)</th>
<th>Length</th>
<th>OUI (3 Octets) 00-00-5E</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>1</td>
<td>BGP Config Sub-TLVs ...</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>... (Up to 507 Octets)</td>
</tr>
</tbody>
</table>
```

Length   The length of the BGP TLV.
Subtype  IETF specific subtype for BGP Config OS-TLV. The value shall be 1.
Value    BGP Config Sub-TLVs each with a 1 byte Type and Length. The Length will include solely the value portion of the TLV and not the Type and Length fields themselves.
2.2.1. BGP Config OS-TLV – Peering Address Sub-TLV

The BGP OS-TLV Peering Address Sub-TLV will be used to advertise the local IP addresses used for BGP sessions and the associated address families specified by AFI/SAFI tuples. The AFI/SAFI tuple, 0/0, indicates to use the associated peering address for all locally configured address families without an explicit peering address specification. As always, the address families supported for a given BGP session will be determined during capabilities negotiation [RFC4760]. It is RECOMMENDED that the wildcard AFI/SAFI be used in deployments with fairly homogenous address family usage.

The format of the BGP Peering Address Sub-TLV is shown below.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type (1)    |     Length    | Address Family| IPv4/IPv6     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
- IPv4/IPv6 Peering Address ...
- IPv4/IPv6 Peering Address ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         AFI                   |     SAFI      |   o o o
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type  The Sub-TLV Type value shall be 1.

Length  The Sub-TLV length in octets will be 4 for IPv4 or 16 for IPv6 plus 3 times the number of AFI/SAFI tuples.

Address Family  IANA Address family (1 for IPv4 or 2 for IPv6)

Peering Address  An IPv4 address (4 octets) or an IPv6 address (16 octets)

AFI/SAFI Pairs  One or more AFI/SAFI tuples for BGP session using this peering address. The AFI/SAFI tuple, 0/0, is a wildcard indicating to attempt negotiation for all AFI/SAFIs.
2.2.2. BGP Config OS-TLV - BGP Local AS Sub-TLV

The BGP Config OS-TLV Local AS Sub-TLV will be used to advertise the 4-octet local Autonomous System (AS) number(s). For AS transitions, a second local AS number may be specified. The format of the BGP Local AS Sub-TLV is shown below.

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type (2) | Length (4 or 8) |          Local AS              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Local AS                        | Optional Second Local AS      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Optional Second Local AS                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type          The Sub-TLV Type value shall be 2.
Length        The Sub-TLV Length will be 4 or 8 octets.
Local AS      Local Autonomous System (AS)
Second Local AS Local Autonomous System (AS)
2.2.3. BGP Config OS-TLV - BGP Identifier Sub-TLV

The BGP Config OS-TLV BGP Identifier Sub-TLV will be used to advertise the 4-octet local BGP Identifier. The BGP Identifier is used for debugging purposes and possibly to reduce the likelihood of BGP connection collisions. The format of the BGP Identifier Sub-TLV is shown below.

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type (3)      | Length (4)    |       BGP Identifier          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      BGP Identifier           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

**Type**  The Sub-TLV Type value shall be 3.

**Length**  The Sub-TLV Length will be 4 octets.

**BGP Identifier**  Local BGP Identifier (aka, BGP Router ID)
2.2.4. BGP Config OS-TLV - Session Group-ID Sub-TLV

The BGP Config OS-TLV Session Group-ID Sub-TLV is an opaque 4-octet value that is used to represent a category of BGP session that is supported on the interface. The format of the Session Group-ID Sub-TLV is shown below.

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type (4) | Length (4) |       Session Group-ID        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Session Group-ID            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- **Type**
  - The Sub-TLV Type value shall be 4.

- **Length**
  - The Sub-TLV Length will be 4 octets.

- **Session Group-ID**
  - The session group-id used to indicate a class or category of BGP session supported on the interface.
2.2.5. BGP Config OS-TLV - BGP Session Capabilities Sub-TLV

The BGP Config OS-TLV Session Capabilities Sub-TLV will be used to advertise an 8-octet Session Capabilities field. The session capabilities are represented as bit flags identifying the supported BGP session capabilities. The format of the BGP Session Capabilities Sub-TLV is shown below.

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type (5)      | Length (8)    |   Session Capabilities        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Session Capabilities                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Session Capabilities         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: The Sub-TLV Type value shall be 5.
Length: The Sub-TLV Length will be 8 octets.
Session Capabilities: Bit fields identify BGP session capabilities.

The BGP Session Capabilities is an 8-octet bit field. The most significant bit is the first bit (Bit 1) of the Session Capabilities. The following bits are defined:

- **Bit 1:** This bit indicates support for TCP MD5 authentication [TCP-MD5].
- **Bit 2:** This bit indicates support for TCP-AO authentication [TCP-AO].
- **Bit 3:** This bit indicates support for Generalized TTL Security Mechanism (GTSM) [GTSM] with a configured TTL range of 254-255.

TCP MD5 authentication is described in [RFC2385]. The TCP Authentication Option (TCP-AO) is described in [RFC5925]. The Generalized TTL Security Mechanism (GTSM) is described in [RFC5082]. If both TCP MD5 authentication and TCP-AO authentication are specified and TCP-AO is supported, it will take precedence.
2.2.6. BGP Config OS-TLV - Key Chain Sub-TLV

The BGP Config OS-TLV Key Chain Sub-TLV is a string specifying the name for the key chain used for session authentication. Key chains [RFC8177] are commonly used for protocol authentication and encryption key specification. Given the limited length of all BGP configuration information, the key chain name will be limited to 64 characters and will not include a trailing string delimiter. The format of the Session Group-ID Sub-TLV is shown below.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type (6)      |   Length (1 - 64)    |       Key Chain Name          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Key Chain Name (Up to 64 Octets)          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Key Chain Name                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: The Sub-TLV Type value shall be 6.
Length: The Sub-TLV Length will be 1 – 64 octets.
Key Chain Name: The name of a key chain to be used for MD5 or TCP-AO authentication.
3. BGP LLDP Peer Discovery Operations

The simple use case is to just use the peer address advertised in the LLDP Packet Data Unit (PDU) to establish a 1-hop BGP peer session. This can be used in data centers using BGP as described in [RFC7938]. The use case where a loopback address or other local address is advertised as the peering address is also supported. However, reachability to a peering address other than the interface address is beyond the scope of this document.

3.1. Advertising BGP Speaker

A BGP speaker MAY advertise its BGP peering address in an LLDP PDU for a link using the BGP Local Address Sub-TLV of the BGP-OS TLV. This can be an IPv4 or IPv6 local address associated with the LLDP link for 1-hop peering. For 2-hop peering, it could be a loopback address or any other address that is local to the node but not the LLDP link. As noted above, reachability to the loopback address is beyond the scope of this document.

A BGP speaker MAY advertise its local AS number using the BGP Local AS Sub-TLV of the BGP-OS TLV. During AS transitions, a second local AS number may be included in the Local AS Sub-TLV. The local BGP identifier may also be advertised using the BGP Identifier Sub-TLV of the BGP-OS TLV. While not specifically required for session establishment, the values may be used for validation, troubleshooting, and connection collision avoidance. A BGP speaker may also announce a Session Group-ID indicating the class or category of session(s) supported and/or mapping to a set of session parameters. Additionally, a BGP speaker MAY also announce relevant capabilities using BGP Session Capabilities Sub-TLV of the BGP-OS TLV.

If TCP MD5 authentication [RFC2385] or TCP Authentication Option (TCP-AO) [RFC5925] is to be used on the session, the Key Chain Sub-TLV of the BGP-OS TLV MAY be used to specify the key chain name.

3.2. Receiving BGP Speaker

A BGP speaker configured for LLDP peer discovery WILL attempt to establish BGP sessions using the address in the BGP Local Address Sub-TLV of BGP-OS TLV format. If the peering address is directly accessible over the link on which the LLDP PDU is received, the BGP speaker will attempt to establish a 1-hop BGP session with the peer.

If the received BGP Peering Address is not directly accessible over the link, the peer must be reachable for the session to be established and the mechanisms for establishing reachability are beyond the scope of this specification. If the BGP speaker receives
the same BGP peering address in LLDP PDUs received on multiple links, it will not establish multiple sessions. Rather, a single 2-hop session will be established.

When the deployment of address families is fairly homogenous across the deployment, the wildcard AFI/SAFI can be utilized to simplify LLDP advertisement. When there is variance in the address families supported, usage of the wildcard could result in session establishment delay due to capabilities negotiation [RFC5492].

A BGP speaker MAY receive a remote neighbor’s local AS number(s) in an LLDP PDU in the BGP Local AS Sub-TLV of the BGP-OS TLV. A BGP speaker MAY use the received local AS number(s) to perform validation checking of the AS received in the OPEN message. A BGP speaker MAY receive a remote neighbor’s BGP Identifier in the BGP Identifier Sub-TLV of the BGP-OS TLV. This can be used to avoid connection collisions by delaying session establishment if the remote BGP Identifier is greater than the receiving speaker’s BGP Identifier.

A BGP speaker MAY receive a Session Group-ID Sub-TLV in the LLDP BGP-OS TLV. This Session Group-ID may be used for validation and/or mapping the session to a particular set of session parameters. For example, the Session Group-ID could be mapped to a spine, leaf, or Top-of-Rack (ToR) session in a data center deployment and can be used to detect cabling problems when an unexpected Session Group-ID is received.

Additionally, A BGP speaker MAY receive a remote neighbor’s capabilities in LLDP in the BGP Session Capabilities Sub-TLV of the BGP-OS TLV. A BGP speaker MAY use the received capabilities to ensure appropriate local neighbor configuration in order to facilitate session establishment.

If TCP MD5 authentication [RFC2385], or TCP Authentication Option (TCP-AO) [RFC5925] is to be used on the session as determined either via the Session Capabilities Sub-TLV, Session Group-ID, or local policy, the key chain name in the Key Chain Sub-TLV of the BGP-OS TLV MAY be used to identify the correct key chain [RFC8177].

4. Security Considerations

This security considerations for BGP [RFC4271] apply equally to this extension.

Additionally, BGP peering address discovery should only be done on trusted links (e.g., in a data center network) since LLDP packets are not authenticated or encrypted [LLDP].
5. IANA Considerations

5.1. IANA Assigned LLDP Subtype

IANA is requested to create a registry for IANA assigned subtypes in the Organizationally Specific TLV assigned to IANA (OUI of 000-00-53 [RFC7042]). Assignment is requested for 1 for the BGP Config OS-TLV.

<table>
<thead>
<tr>
<th>Range</th>
<th>Assignment Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Reserved (not to be assigned)</td>
</tr>
<tr>
<td>1</td>
<td>BGP Configuration</td>
</tr>
<tr>
<td>2-127</td>
<td>Unassigned (IETF Review)</td>
</tr>
<tr>
<td>128-254</td>
<td>Reserved (Not to be assigned now)</td>
</tr>
<tr>
<td>255</td>
<td>Reserved (not to be assigned)</td>
</tr>
</tbody>
</table>

IANA LLDP Organizationally Specific TLV Sub-Types

- Types in the range 2-127 are to be assigned subject to IETF Review. New values are assigned only through RFCs that have been shepherded through the IESG as AD-Sponsored or IETF WG Documents [RFC5226].

- Types in the range 128-254 are reserved and not to be assigned at this time. Before any assignments can be made in this range, there MUST be a Standards Track RFC that specifies IANA Considerations that covers the range being assigned.

5.2. BGP Config LLDP OS-TLV Sub-TLVs

IANA is requested to create a registry for Sub-TLVs of the BGP Config LLDP OS-TLV. Assignment is requested for 1 for the BGP Peering Address Sub-TLV. Assignment is also requested for 2 for the Local AS Sub-TLV. Additionally, assignment is requested for 3 for the BGP Identifier Sub-TLV, 4 for the BGP Session Group-ID, 5 for the Session Capabilities Sub-TLV, and 6 for the Key Chain Name.
<table>
<thead>
<tr>
<th>Range</th>
<th>Assignment Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Reserved (not to be assigned)</td>
</tr>
<tr>
<td>1</td>
<td>Peering Address</td>
</tr>
<tr>
<td>2</td>
<td>Local AS</td>
</tr>
<tr>
<td>3</td>
<td>BGP Identifier</td>
</tr>
<tr>
<td>4</td>
<td>Session Group-ID</td>
</tr>
<tr>
<td>5</td>
<td>Session Capabilities</td>
</tr>
<tr>
<td>6</td>
<td>Key Chain Name</td>
</tr>
<tr>
<td>7-127</td>
<td>Unassigned (IETF Review)</td>
</tr>
<tr>
<td>128-254</td>
<td>Reserved (Not to be assigned now)</td>
</tr>
<tr>
<td>255</td>
<td>Reserved (not to be assigned)</td>
</tr>
</tbody>
</table>

**LLDP BGP Config OS-TLV Types**

- Types in the range 7-127 are to be assigned subject to IETF Review. New values are assigned only through RFCs that have been shepherded through the IESG as AD-Sponsored or IETF WG Documents [RFC5226].

- Types in the range 128-254 are reserved and not to be assigned at this time. Before any assignments can be made in this range, there MUST be a Standards Track RFC that specifies IANA Considerations that covers the range being assigned.

6. Contributors

Contributors’ Addresses

7. References

7.1. Normative References


7.2. Informative References


Appendix A. Acknowledgments

Thanks to Sujay Gupta for review and comments.

The RFC text was produced using Marshall Rose’s xml2rfc tool.

Authors’ Addresses

Acee Lindem
Cisco Systems
301 Midenhall Way
Cary, NC 27513
USA

Email: acee@cisco.com

Keyur Patel
Arrcus, Inc

Email: keyur@arrcus.com

Shawn Zandi
LinkedIn
222 2nd Street
San Francisco, CA 94105
USA

Email: szandi@linkedin.com
Route Leaks are the propagation of BGP prefixes which violate assumptions of BGP topology relationships; e.g. passing a route learned from one peer to another peer or to a transit provider, passing a route learned from one transit provider to another transit provider or to a peer. Today, approaches to leak prevention rely on marking routes by operator configuration, with no check that the configuration corresponds to that of the BGP neighbor, or enforcement that the two BGP speakers agree on the relationship. This document enhances BGP OPEN to establish agreement of the (peer, customer, provider, Route Server, Route Server client) relationship of two neighboring BGP speakers to enforce appropriate configuration on both sides. Propagated routes are then marked with an OTC attribute according to the agreed relationship, allowing both prevention and detection of route leaks.

Requirements Language

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.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute
Internet-Draft            Route Leak Prevention                July 2019

working documents as Internet-Drafts. The list of current Internet-
Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months
and may be updated, replaced, or obsoleted by other documents at any
time. It is inappropriate to use Internet-Drafts as reference
material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 9, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the
document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal
Provisions Relating to IETF Documents
(https://trustee.ietf.org/license-info) in effect on the date of
publication of this document. Please review these documents
carefully, as they describe your rights and restrictions with respect
to this document. Code Components extracted from this document must
include Simplified BSD License text as described in Section 4.e of
the Trust Legal Provisions and are provided without warranty as
described in the Simplified BSD License.

Table of Contents

1. Introduction ................................................. 3
2. Peering Relationships ....................................... 3
3. BGP Role .................................................. 4
4. Role capability ............................................. 4
5. Role correctness ............................................ 5
   5.1. Strict mode ........................................... 5
6. BGP Only To Customer attribute ............................... 6
7. Enforcement ................................................. 6
8. Additional Considerations .................................... 7
9. IANA Considerations ......................................... 7
10. Security Considerations ...................................... 8
11. Acknowledgments ............................................ 8
12. References ............................................... 8
   12.1. Normative References ................................ 8
   12.2. Informative References ............................... 9
Authors’ Addresses ........................................... 9

1. Introduction

BGP route leaks are BGP route(s) which were learned from transit provider or peer and then announced to another provider or peer. See [RFC7908]. These are usually the result of misconfigured or absent BGP route filtering or lack of coordination between two BGP speakers.

The mechanism proposed in [I-D.ietf-idr-route-leak-detection-mitigation] uses large-communities to attempt detection of route leaks. While signaling using communities is easy to implement, it relies on operator maintained policy configuration which is too easily, and too often, misconfigured. Another problem may occur if the community signal is stripped, accidentally or maliciously.

This document provides configuration automation using ‘BGP roles’, which are negotiated using a new BGP Capability Code in OPEN message, [RFC5492] Sec 4. Either or both BGP speakers MAY be configured to require that this capability be agreed for the BGP OPEN to succeed.

A new BGP Path Attribute is specified that SHOULD be automatically configured using BGP roles. This attribute prevents networks from creating leaks, and detects leaks created by third-parties.

2. Peering Relationships

Despite uses of words such as "Customer," "Peer." etc.; these are not business relationships, who pays whom, etc. These are common terms to represent restrictions on BGP route propagation, sometimes known as the Gao-Rexford model [cite].

A Provider: MAY send to a customer all available prefixes.

A Customer: MAY send to a provider their own prefixes and prefixes learned from any of their customers. A customer MUST NOT send to a provider prefixes learned from its peers, from other providers, or from Route Servers.

A Route Server (RS) MAY send to a RS Client all available prefixes.

A Route Server Client (RS-client) MAY send to an RS its own prefixes and prefixes learned from its customers. A RS-client MUST NOT send to an RS prefixes learned from peers, from its providers, or from other RS(s).

A Peer: MAY send to a peer its own prefixes and prefixes learned from its customers. A peer MUST NOT send to a peer prefixes learned from other peers, from its providers, or from RS(s).
Of course, any BGP speaker may apply policy to reduce what is announced, and a recipient may apply policy to reduce the set of routes they accept. Violation of the above rules may result in route leaks so MUST not be allowed. Automatic enforcement of these rules should significantly reduce configuration mistakes. While these enforcing the above rules will address most BGP peering scenarios, their configuration isn’t part of BGP itself; therefore requiring configuration of ingress and egress prefix filters is still strongly advised.

3. BGP Role

BGP Role is a new configuration option that SHOULD be configured on each BGP session. It reflects the real-world agreement between two BGP speakers about their relationship.

Allowed Role values for eBGP sessions are:

- Provider - sender is a transit provider to neighbor;
- Customer - sender is transit customer of neighbor;
- RS - sender is a Route Server, usually at internet exchange point (IX)
- RS-Client - sender is client of RS
- Peer - sender and neighbor are peers;

Since BGP Role reflects the relationship between two BGP speakers, it could also be used for more than route leak mitigation.

4. Role capability

The TLV (type, length, value) of the BGP Role capability are:

- Type - <TBD1>;
- Length - 1 (octet);
- Value - integer corresponding to speaker’s BGP Role.
Table 1: Predefined BGP Role Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Role name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Sender is Provider</td>
</tr>
<tr>
<td>1</td>
<td>Sender is RS</td>
</tr>
<tr>
<td>2</td>
<td>Sender is RS-Client</td>
</tr>
<tr>
<td>3</td>
<td>Sender is Customer</td>
</tr>
<tr>
<td>4</td>
<td>Sender is Peer</td>
</tr>
</tbody>
</table>

5. Role correctness

Section 3 described how BGP Role encodes the relationship between two BGP speakers. But the mere presence of BGP Role doesn’t automatically guarantee role agreement between two BGP peers.

To enforce correctness, the BGP Role check is used with a set of constrains on how speakers’ BGP Roles MUST correspond. Of course, each speaker MUST announce and accept the BGP Role capability in the BGP OPEN message exchange.

If a speaker receives a BGP Role capability, it MUST check the value of the received capability with its own BGP Role (if it is set). The allowed pairings are (first a sender’s Role, second the receiver’s Role):

<table>
<thead>
<tr>
<th>Sender Role</th>
<th>Receiver Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provider</td>
<td>Customer</td>
</tr>
<tr>
<td>Customer</td>
<td>Provider</td>
</tr>
<tr>
<td>RS</td>
<td>RS-Client</td>
</tr>
<tr>
<td>RS-Client</td>
<td>RS</td>
</tr>
<tr>
<td>Peer</td>
<td>Peer</td>
</tr>
</tbody>
</table>

Table 2: Allowed Role Capabilities

If the Role pair is not in the above table, a speaker MUST send a Role Mismatch Notification (code 2, sub-code <TBD2>).

5.1. Strict mode

A new BGP configuration option "strict mode" is defined with values of true or false. If set to true, then the speaker MUST refuse to establish a BGP session with a neighbor which does not announce the
BGP Role capability in the OPEN message. If a speaker rejects a connection, it MUST send a Connection Rejected Notification [RFC4486] (Notification with error code 6, subcode 5). By default, strict mode SHOULD be set to false for backward compatibility with BGP speakers that do not yet support this mechanism.

6. BGP Only To Customer attribute

The Only To Customer (OTC) BGP Attribute is a new optional, transitive BGP Path attribute with the Type Code <TBD3>.

This four byte attribute MUST apply the following policy:

1. If a route with OTC attribute is received from Customer or RS-client – it’s a route leak and MUST be rejected.
2. If a route with OTC attribute is received from Peer and its value isn’t equal to the neighbor’s ASN – it’s a route leak and MUST be rejected.
3. If a route is received from a Provider, Peer or RS and the OTC attribute has not been set it MUST be added with value equal to AS number of the neighbor (sender).

The egress policy MUST be:

1. A route with the OTC attribute set MUST NOT be sent to providers, peers, or RS(s).
2. If route is sent to customer or peer and the OTC attribute is not set it MUST be added with value equal to AS number of the sender.

Once the OTC attribute has been set, it MUST be preserved unchanged.

7. Enforcement

Having the relationship unequivocally agreed between the two peers in BGP OPEN is critical; the BGP implementations enforce the relationship irrespective of operator policy configuration errors.

Similarly, the application of that relationship on prefix propagation using OTC MUST BE enforced by the BGP implementations, and not exposed to user mis-configuration.

As opposed to communities, BGP attributes may not be generally modified or filtered by the operator. The router(s) enforce them. This is the desired property for the OTC marking. Hence, this document specifies OTC as an attribute.
8. Additional Considerations

As the BGP Role reflects the peering relationship between neighbors, it might have other uses. For example, BGP Role might affect route priority, or be used to distinguish borders of a network if a network consists of multiple ASs. Though such uses may be worthwhile, they are not the goal of this document. Note that such uses would require local policy control.

As BGP role configuration results in automatic creation of inbound/outbound filters, existence of roles should be treated as existence of Import and Export policy. [RFC8212]

There are peering relationships which are ‘complex’; e.g. when both parties are intentionally sending prefixes received from each other to their peers and/or upstreams. If multiple BGP peerings can segregate the ‘complex’ parts of the relationship, the complex peering roles can be segregated into different BGP sessions, and normal BGP Roles MUST be used on the non-complex sessions. No Roles SHOULD be configured on ‘complex’ BGP sessions, and OTC MUST be set by configuration on a per-prefix basis. There can be no measures to check correctness of OTC use if Role is not configured.

9. IANA Considerations

This document defines a new Capability Codes option [to be removed upon publication: http://www.iana.org/assignments/capability-codes/capability-codes.xhtml] [RFC5492], named "BGP Role", assigned value <TBD1>. The length of this capability is 1.

The BGP Role capability includes a Value field, for which IANA is requested to create and maintain a new sub-registry called "BGP Role Value". Assignments consist of Value and corresponding Role name. Initially this registry is to be populated with the data in Table 1. Future assignments may be made by a standard action procedure[RFC5226].

This document defines new subcode, "Role Mismatch", assigned value <TBD2> in the OPEN Message Error subcodes registry [to be removed upon publication: http://www.iana.org/assignments/bgp-parameters/bgp-parameters.xhtml#bgp-parameters-6] [RFC4271].

This document defines a new optional, transitive BGP Path Attributes option, named "Only To Customer", assigned value <TBD3> [To be removed upon publication: http://www.iana.org/assignments/bgp-parameters/bgp-parameters.xhtml#bgp-parameters-2] [RFC4271]. The length of this attribute is 0.
10. Security Considerations

This document proposes a mechanism for prevention of route leaks that are the result of BGP policy mis-configuration.

Deliberate sending of a known conflicting BGP Role could be used to sabotage a BGP connection. This is easily detectable.

A misconfiguration in OTC setup may affect prefix propagation. But the automation that is provided by BGP roles should make such misconfiguration unlikely.

11. Acknowledgments

The authors wish to thank Douglas Montgomery, Brian Dickson, Andrei Robachevsky, and Daniel Ginsburg for their contributions to a variant of this work.

12. References

12.1. Normative References


12.2. Informative References

[I-D.ietf-idr-route-leak-detection-mitigation]


Authors' Addresses

Alexander Azimov
Qrator Labs
Email: a.e.azimov@gmail.com

Eugene Bogomazov
Qrator Labs
Email: eb@qrator.net

Randy Bush
Internet Initiative Japan & Arrcus
Email: randy@psg.com

Keyur Patel
Arrcus, Inc.
Email: keyur@arrcus.com
Abstract

This document defines a Border Gateway Protocol Network Layer Reachability Information (BGP NLRI) encoding format that can be used to distribute traffic Flow Specifications. This allows the routing system to propagate information regarding more specific components of the traffic aggregate defined by an IP destination prefix.

It specifies IPv4 traffic Flow Specifications via a BGP NLRI which carries traffic Flow Specification filter, and an Extended community value which encodes actions a routing system can take if the packet matches the traffic flow filters. The flow filters and the actions are processed in a fixed order. Other drafts specify IPv6, MPLS addresses, L2VPN addresses, and NV03 encapsulation of IP addresses.

This document obsoletes RFC5575 and RFC7674 to correct unclear specifications in the flow filters.

Applications which use the bgp Flow Specification are: 1) application which automate inter-domain coordination of traffic filtering, such as what is required in order to mitigate (distributed) denial-of-service attacks; 2) applications which control traffic filtering in the context of a BGP/MPLS VPN service, and 3) applications with centralized control of traffic in a SDN or NFV context. Some deployments of these three applications can be handled by the strict ordering of the BGP NLRI traffic flow filters, and the strict actions encoded in the extended community Flow Specification actions.
Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 20, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction ................................................. 3
2. Definitions of Terms Used in This Memo .................... 5
3. Flow Specifications ......................................... 6
   4.1. Length Encoding ....................................... 7
   4.2. NLRI Value Encoding .................................... 8
       4.2.1. Type 1 - Destination Prefix .................... 8
       4.2.2. Type 2 - Source Prefix ........................... 8
       4.2.3. Type 3 - IP Protocol .............................. 8
       4.2.4. Type 4 - Port ................................... 10
       4.2.5. Type 5 - Destination Port ....................... 10
       4.2.6. Type 6 - Source Port .............................. 10
       4.2.7. Type 7 - ICMP type ............................... 11
       4.2.8. Type 8 - ICMP code ............................... 11
4.2.9. Type 9 - TCP flags ..................................... 11
4.2.10. Type 10 - Packet length ............................... 12
4.2.11. Type 11 - DSCP (Diffserv Code Point) .................. 12
4.2.12. Type 12 - Fragment ..................................... 12
4.3. Examples of Encodings ....................................... 13
5. Traffic Filtering .............................................. 14
5.1. Ordering of Traffic Filtering Rules ......................... 15
6. Validation Procedure .......................................... 17
7. Traffic Filtering Actions ....................................... 18
7.1. Traffic Rate in Bytes (traffic-rate-bytes) sub-type 0x06 19
7.2. Traffic Rate in Packets (traffic-rate-packets) sub-type TBD ...................................................... 20
7.3. Traffic-action (traffic-action) sub-type 0x07 ............... 20
7.4. RT Redirect (rt-redirect) sub-type 0x08 .................... 21
7.5. Traffic Marking (traffic-marking) sub-type 0x09 ............ 21
7.6. Considerations on Traffic Action Interference ............... 21
8. Dissemination of Traffic Filtering in BGP/MPLS VPN Networks . 22
8.1. Validation Procedures for BGP/MPLS VPNs .................... 23
8.2. Traffic Actions Rules ....................................... 23
9. Limitations of Previous Traffic Filtering Efforts ............. 23
9.1. Limitations in Previous DDoS Traffic Filtering Efforts .... 23
9.2. Limitations in Previous BGP/MPLS Traffic Filtering Efforts .......................................................... 24
10. Traffic Monitoring ............................................. 24
11. Error-Handling and Future NLRI Extensions ..................... 24
12. IANA Considerations ............................................ 25
12.1. AFI/SAFI Definitions ....................................... 25
12.2. Flow Component Definitions ................................ 25
12.3. Extended Community Flow Specification Actions .......... 26
13. Security Considerations ....................................... 29
14. Operational Security Considerations ........................ 30
15. Original authors .............................................. 30
16. Acknowledgements ............................................. 30
17. References .................................................... 30
17.1. Normative References ...................................... 30
17.2. Informative References .................................... 32
17.3. URIs ....................................................... 32
Appendix A. Comparison with RFC 5575 .......................... 32
Authors’ Addresses ................................................. 33

1. Introduction

Modern IP routers contain both the capability to forward traffic according to IP prefixes as well as to classify, shape, rate limit, filter, or redirect packets based on administratively defined policies.
These traffic policy mechanisms allow the router to define match rules that operate on multiple fields of the packet header. Actions such as the ones described above can be associated with each rule.

The n-tuple consisting of the matching criteria defines an aggregate traffic Flow Specification. The matching criteria can include elements such as source and destination address prefixes, IP protocol, and transport protocol port numbers.

This document defines a general procedure to encode flow specification rules for aggregated traffic flows so that they can be distributed as a BGP [RFC4271] NLRI. Additionally, we define the required mechanisms to utilize this definition to the problem of immediate concern to the authors: intra- and inter-provider distribution of traffic filtering rules to filter (distributed) denial-of-service (DoS) attacks.

By expanding routing information with Flow Specifications, the routing system can take advantage of the ACL (Access Control List) or firewall capabilities in the router’s forwarding path. Flow specifications can be seen as more specific routing entries to a unicast prefix and are expected to depend upon the existing unicast data information.

A Flow Specification received from an external autonomous system will need to be validated against unicast routing before being accepted. If the aggregate traffic flow defined by the unicast destination prefix is forwarded to a given BGP peer, then the local system can install more specific flow rules that may result in different forwarding behavior, as requested by this system.

The key technology components required to address the class of problems targeted by this document are:

1. Efficient point-to-multipoint distribution of control plane information.
2. Inter-domain capabilities and routing policy support.
3. Tight integration with unicast routing, for verification purposes.

Items 1 and 2 have already been addressed using BGP for other types of control plane information. Close integration with BGP also makes it feasible to specify a mechanism to automatically verify flow information against unicast routing. These factors are behind the choice of BGP as the carrier of Flow Specification information.
As with previous extensions to BGP, this specification makes it possible to add additional information to Internet routers. These are limited in terms of the maximum number of data elements they can hold as well as the number of events they are able to process in a given unit of time. The authors believe that, as with previous extensions, service providers will be careful to keep information levels below the maximum capacity of their devices.

Experience with previous BGP extensions has also shown that the maximum capacity of BGP speakers has been gradually increased according to expected loads. For example Internet unicast routing as well as other BGP applications increased their maximum capacity as they gain popularity.

From an operational perspective, the utilization of BGP as the carrier for this information allows a network service provider to reuse both internal route distribution infrastructure (e.g., route reflector or confederation design) and existing external relationships (e.g., inter-domain BGP sessions to a customer network).

While it is certainly possible to address this problem using other mechanisms, this solution has been utilized in deployments because of the substantial advantage of being an incremental addition to already deployed mechanisms.

In current deployments, the information distributed by the flow-spec extension is originated both manually as well as automatically. The latter by systems that are able to detect malicious flows. When automated systems are used, care should be taken to ensure their correctness as well as to limit the number and advertisement rate of flow routes.

This specification defines required protocol extensions to address most common applications of IPv4 unicast and VPNv4 unicast filtering. The same mechanism can be reused and new match criteria added to address similar filtering needs for other BGP address families such as IPv6 families [I-D.ietf-idr-flow-spec-v6],

2. Definitions of Terms Used in This Memo

NLRI - Network Layer Reachability Information.

RIB - Routing Information Base.

Loc-RIB - Local RIB.

AS - Autonomous System.
VRF - Virtual Routing and Forwarding instance.

PE - Provider Edge router

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.

3. Flow Specifications

A Flow Specification is an n-tuple consisting of several matching criteria that can be applied to IP traffic. A given IP packet is said to match the defined flow if it matches all the specified criteria. This n-tuple is encoded into a BGP NLRI defined below.

A given flow may be associated with a set of attributes, depending on the particular application; such attributes may or may not include reachability information (i.e., NEXT_HOP). Well-known or AS-specific community attributes can be used to encode a set of predetermined actions.

A particular application is identified by a specific (Address Family Identifier, Subsequent Address Family Identifier (AFI, SAFI)) pair [RFC4760] and corresponds to a distinct set of RIBs. Those RIBs should be treated independently from each other in order to assure non-interference between distinct applications.

BGP itself treats the NLRI as an key to an entry in its databases. Entries that are placed in the Loc-RIB are then associated with a given set of semantics, which is application dependent. This is consistent with existing BGP applications. For instance, IP unicast routing (AFI=1, SAFI=1) and IP multicast reverse-path information (AFI=1, SAFI=2) are handled by BGP without any particular semantics being associated with them until installed in the Loc-RIB.

Standard BGP policy mechanisms, such as UPDATE filtering by NLRI prefix as well as community matching and manipulation, MUST apply to the Flow Specification defined NLRI-type, especially in an inter-domain environment. Network operators can also control propagation of such routing updates by enabling or disabling the exchange of a particular (AFI, SAFI) pair on a given BGP peering session.
4. Dissemination of IPv4 Flow Specification Information

We define a "Flow Specification" NLRI type (Figure 1) that may include several components such as destination prefix, source prefix, protocol, ports, and others (see Section 4.2 below).

This NLRI information is encoded using MP_REACH_NLRI and MP_UNREACH_NLRI attributes as defined in [RFC4760]. Whenever the corresponding application does not require Next-Hop information, this shall be encoded as a 0-octet length Next Hop in the MP_REACH_NLRI attribute and ignored on receipt.

The NLRI field of the MP_REACH_NLRI and MP_UNREACH_NLRI is encoded as a 1- or 2-octet NLRI length field followed by a variable-length NLRI value. The NLRI length is expressed in octets.

\[
\begin{array}{c}
\text{length (0xnn or 0xfn nn)} \\
\text{NLRI value (variable)} \\
\end{array}
\]

Figure 1: Flow-spec NLRI for IPv4

Implementations wishing to exchange Flow Specification rules MUST use BGP’s Capability Advertisement facility to exchange the Multiprotocol Extension Capability Code (Code 1) as defined in [RFC4760]. The (AFI, SAFI) pair carried in the Multiprotocol Extension Capability MUST be (AFI=1, SAFI=133) for IPv4 Flow Specification, and (AFI=1, SAFI=134) for VPNv4 Flow Specification.

4.1. Length Encoding

- If the NLRI length value is smaller than 240 (0xf0 hex), the length field can be encoded as a single octet.

- Otherwise, it is encoded as an extended-length 2-octet value in which the most significant nibble of the first byte is all ones.

In figure 1 above, values less-than 240 are encoded using two hex digits (0xnn). Values above 239 are encoded using 3 hex digits (0xfnnn). The highest value that can be represented with this encoding is 4095. The value 241 is encoded as 0xf0f1.
4.2. NLRI Value Encoding

The Flow Specification NLRI-type consists of several optional subcomponents. A specific packet is considered to match the flow specification when it matches the intersection (AND) of all the components present in the specification.

The encoding of each of the NLRI components begins with a type field (1 octet) followed by a variable length parameter. Section 4.2.1 to Section 4.2.12 define component types and parameter encodings for the IPv4 IP layer and transport layer headers. IPv6 NLRI component types are described in [I-D.ietf-idr-flow-spec-v6].

Flow Specification components must follow strict type ordering by increasing numerical order. A given component type may (exactly once) or may not be present in the specification. If present, it MUST precede any component of higher numeric type value.

All combinations of component types within a single NLRI are allowed, even if the combination makes no sense from a semantical perspective. If a given component type within a prefix in unknown, the prefix in question cannot be used for traffic filtering purposes by the receiver. Since a Flow Specification has the semantics of a logical AND of all components, if a component is FALSE, by definition it cannot be applied. However, for the purposes of BGP route propagation, this prefix should still be transmitted since BGP route distribution is independent on NLRI semantics.

4.2.1. Type 1 - Destination Prefix

Encoding: <type (1 octet), prefix length (1 octet), prefix>

Defines: the destination prefix to match. Prefixes are encoded as in BGP UPDATE messages, a length in bits is followed by enough octets to contain the prefix information.

4.2.2. Type 2 - Source Prefix

Encoding: <type (1 octet), prefix-length (1 octet), prefix>

Defines the source prefix to match.

4.2.3. Type 3 - IP Protocol

Encoding: <type (1 octet), [op, value]+>

Contains a set of {operator, value} pairs that are used to match the IP protocol value byte in IP packets.
The operator byte is encoded as:

```
+---+---+---+---+---+---+---+---+
| e | a |  len  | 0 | lt | gt | eq |
+---+---+---+---+---+---+---+---+
```

Numeric operator

e - end-of-list bit. Set in the last (op, value) pair in the list.

a - AND bit. If unset, the previous term is logically ORed with the current one. If set, the operation is a logical AND. In the first operator byte of a sequence it SHOULD be encoded as unset and and MUST be treated as always unset on decoding. The AND operator has higher priority than OR for the purposes of evaluating logical expressions.

len - length of the value field for this operator given as (1 << len). This encodes 1 (00) - 8 (11) bytes. Type 3 flow component values SHOULD be encoded as single byte (len = 00).

0 - SHOULD be set to 0 on NLRI encoding, and MUST be ignored during decoding

lt - less than comparison between data and value.

gt - greater than comparison between data and value.

eq - equality between data and value.

The bits lt, gt, and eq can be combined to produce common relational operators such as "less or equal", "greater or equal", and "not equal to".
<table>
<thead>
<tr>
<th>lt</th>
<th>gt</th>
<th>eq</th>
<th>Resulting operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>false (independent of the value)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>== (equal)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>&gt; (greater than)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>&gt;= (greater than or equal)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>&lt; (less than)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>&lt;= (less than or equal)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>!= (not equal value)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>true (independent of the value)</td>
</tr>
</tbody>
</table>

Table 1: Comparison operation combinations

4.2.4. Type 4 – Port

Encoding:<type (1 octet), [op, value]>+

Defines a list of {operator, value} pairs that matches source OR destination TCP/UDP ports. This list is encoded using the numeric operator format defined in Section 4.2.3. Values SHOULD be encoded as 1- or 2-byte quantities.

Port, source port, and destination port components evaluate to FALSE if the IP protocol field of the packet has a value other than TCP or UDP, if the packet is fragmented and this is not the first fragment, or if the system is unable to locate the transport header. Different implementations may or may not be able to decode the transport header in the presence of IP options or Encapsulating Security Payload (ESP) NULL [RFC4303] encryption.

4.2.5. Type 5 – Destination Port

Encoding:<type (1 octet), [op, value]>+

Defines a list of {operator, value} pairs used to match the destination port of a TCP or UDP packet. This list is encoded using the numeric operator format defined in Section 4.2.3. Values SHOULD be encoded as 1- or 2-byte quantities.

4.2.6. Type 6 – Source Port

Encoding:<type (1 octet), [op, value]>+

Defines a list of {operator, value} pairs used to match the source port of a TCP or UDP packet. This list is encoded using the
numeric operator format defined in Section 4.2.3. Values SHOULD be encoded as 1- or 2-byte quantities.

4.2.7. Type 7 - ICMP type

Encoding:<type (1 octet), [op, value]+>

Defines a list of {operator, value} pairs used to match the type field of an ICMP packet. This list is encoded using the numeric operator format defined in Section 4.2.3. Values SHOULD be encoded using a single byte.

The ICMP type specifiers evaluate to FALSE whenever the protocol value is not ICMP.

4.2.8. Type 8 - ICMP code

Encoding:<type (1 octet), [op, value]+>

Defines a list of {operator, value} pairs used to match the code field of an ICMP packet. This list is encoded using the numeric operator format defined in Section 4.2.3. Values SHOULD be encoded using a single byte.

The ICMP code specifiers evaluate to FALSE whenever the protocol value is not ICMP.

4.2.9. Type 9 - TCP flags

Encoding:<type (1 octet), [op, bitmask]+>

Bitmask values can be encoded as a 1- or 2-byte bitmask. When a single byte is specified, it matches byte 13 of the TCP header [RFC0793], which contains bits 8 through 15 of the 4th 32-bit word. When a 2-byte encoding is used, it matches bytes 12 and 13 of the TCP header with the data offset field having a "don't care" value.

This component evaluates to FALSE for packets that are not TCP packets.

This type uses the bitmask operator format, which differs from the numeric operator format in the lower nibble.
0 1 2 3 4 5 6 7
+---+---+---+---+---+---+---+---+
| e | a | len | 0 | 0 | not | m |
+---+---+---+---+---+---+---+---+

Bitmask operator

e, a, len - Most significant nibble: (end-of-list bit, AND bit, and length field), as defined for in the numeric operator format in Section 4.2.3.

not - NOT bit. If set, logical negation of operation.

m - Match bit. If set, this is a bitwise match operation defined as "(data AND value) == value"; if unset, (data AND value) evaluates to TRUE if any of the bits in the value mask are set in the data

0 - all 0 bits SHOULD be set to 0 on NLRI encoding, and MUST be ignored during decoding

4.2.10. Type 10 - Packet length

Encoding:<type (1 octet), [op, value]+>

Defines a list of {operator, value} pairs used to match on the total IP packet length (excluding Layer 2 but including IP header). This list is encoded using the numeric operator format defined in Section 4.2.3. Values SHOULD be encoded using 1- or 2-byte quantities.

4.2.11. Type 11 - DSCP (Diffserv Code Point)

Encoding:<type (1 octet), [op, value]+>

Defines a list of {operator, value} pairs used to match the 6-bit DSCP field [RFC2474]. This list is encoded using the numeric operator format defined in Section 4.2.3. Values SHOULD be encoded using a single byte. The six least significant bits contain the DSCP value. All other bits SHOULD be encoded as zero and ignored on decoding.

4.2.12. Type 12 - Fragment

Encoding:<type (1 octet), [op, bitmask]+>

Uses bitmask operator format defined in Section 4.2.9.
Bitmask values:

- Bit 7 - Don’t fragment (DF)
- Bit 6 - Is a fragment (IsF)
- Bit 5 - First fragment (FF)
- Bit 4 - Last fragment (LF)
- Bit 0-3 - SHOULD be set to 0 on NLRI encoding, and MUST be ignored during decoding

4.3. Examples of Encodings

An example of a Flow Specification encoding for: "all packets to 10.0.1/24 and TCP port 25".

<table>
<thead>
<tr>
<th>destination</th>
<th>proto</th>
<th>port</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01 18 0a 00 01</td>
<td>03 81 06</td>
<td>04 81 19</td>
</tr>
</tbody>
</table>

Decode for protocol:

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x03 0x81 0x06</td>
</tr>
</tbody>
</table>

An example of a Flow Specification encoding for: "all packets to 10.1.1/24 from 192/8 and port (range [137, 139] or 8080)".

<table>
<thead>
<tr>
<th>destination</th>
<th>source</th>
<th>port</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01 18 0a 01 01</td>
<td>02 08 c0</td>
<td>04 03 89 45 8b 91 1f 90</td>
</tr>
</tbody>
</table>

Decode for port:

<table>
<thead>
<tr>
<th>Value</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0x04</td>
<td>type</td>
</tr>
<tr>
<td>0x03</td>
<td>operator</td>
</tr>
<tr>
<td>0x89</td>
<td>value</td>
</tr>
<tr>
<td>0x45</td>
<td>operator</td>
</tr>
<tr>
<td>0x8b</td>
<td>value</td>
</tr>
<tr>
<td>0x91</td>
<td>operator</td>
</tr>
<tr>
<td>0x1f90</td>
<td>value</td>
</tr>
<tr>
<td>0x04</td>
<td>type</td>
</tr>
<tr>
<td>0x03</td>
<td>operator</td>
</tr>
<tr>
<td>0x89</td>
<td>value</td>
</tr>
<tr>
<td>0x45</td>
<td>operator</td>
</tr>
<tr>
<td>0x8b</td>
<td>value</td>
</tr>
<tr>
<td>0x91</td>
<td>operator</td>
</tr>
<tr>
<td>0x1f90</td>
<td>value</td>
</tr>
</tbody>
</table>

This constitutes an NLRI with an NLRI length of 16 octets.

5. Traffic Filtering

Traffic filtering policies have been traditionally considered to be relatively static. Limitations of the static mechanisms caused this mechanism to be designed for the three new applications of traffic filtering (prevention of traffic-based, denial-of-service (DOS) attacks, traffic filtering in the context of BGP/MPLS VPN service, and centralized traffic control for SDN/NFV networks) requires coordination among service providers and/or coordination among the AS within a service provider. Section 9 has details on the limitation of previous mechanisms and why BGP Flow Specification provides a solution for to prevent DOS and aid BGP/MPLS VPN filtering rules.

This Flow Specification NLRI defined above to convey information about traffic filtering rules for traffic that should be discarded or handled in manner specified by a set of pre-defined actions (which are defined in BGP Extended Communities). This mechanism is primarily designed to allow an upstream autonomous system to perform inbound filtering in their ingress routers of traffic that a given downstream AS wishes to drop.

In order to achieve this goal, this draft specifies two application specific NLRI identifiers that provide traffic filters, and a set of actions encoding in BGP Extended Communities. The two application specific NLRI identifiers are:

- IPv4 Flow Specification identifier (AFI=1, SAFI=133) along with specific semantic rules for IPv4 routes, and
VPNv4 Flow Specification identifier (AFI=1, SAFI=134) value, which can be used to propagate traffic filtering information in a BGP/MPLS VPN environment.

Distribution of the IPv4 Flow Specification is described in Section 6, and distribution of BGP/MPLS traffic Flow Specification is described in Section 8. The traffic filtering actions are described in Section 7.

5.1. Ordering of Traffic Filtering Rules

With traffic filtering rules, more than one rule may match a particular traffic flow. Thus, it is necessary to define the order at which rules get matched and applied to a particular traffic flow. This ordering function must be such that it must not depend on the arrival order of the Flow Specification’s rules and must be consistent in the network.

The relative order of two Flow Specification rules is determined by comparing their respective components. The algorithm starts by comparing the left-most components of the rules. If the types differ, the rule with lowest numeric type value has higher precedence (and thus will match before) than the rule that doesn’t contain that component type. If the component types are the same, then a type-specific comparison is performed (see below) if the types are equal the algorithm continues with the next component.

For IP prefix values (IP destination or source prefix): If the prefixes overlap, the one with the longer prefix-length has higher precedence. If they do not overlap the one with the lowest IP value has higher precedence.

For all other component types, unless otherwise specified, the comparison is performed by comparing the component data as a binary string using the memcmp() function as defined by the ISO C standard. For strings with equal lengths the lowest string (memcmp) has higher precedence. For strings of different lengths, the common prefix is compared. If the common prefix is not equal the string with the lowest prefix has higher precedence. If the common prefix is equal, the longest string is considered to have higher precedence than the shorter one.

The code below shows a Python3 implementation of the comparison algorithm. The full code was tested with Python 3.6.3 and can be obtained at https://github.com/stoffi92/flowspec-cmp [1].

```python
import itertools

<CODE BEGINS>
import itertools

import ipaddress

def flow_rule_cmp(a, b):
    for comp_a, comp_b in itertools.zip_longest(a.components, b.components):
        # If a component type does not exist in one rule
        # this rule has lower precedence
        if not comp_a:
            return B_HAS_PRECEDENCE
        if not comp_b:
            return A_HAS_PRECEDENCE
        # higher precedence for lower component type
        if comp_a.component_type < comp_b.component_type:
            return A_HAS_PRECEDENCE
        if comp_a.component_type > comp_b.component_type:
            return B_HAS_PRECEDENCE
        # component types are equal -> type specific comparison
        if comp_a.component_type in (IP_DESTINATION, IP_SOURCE):
            # assuming comp_a.value, comp_b.value of type ipaddress
            if comp_a.value.overlaps(comp_b.value):
                # longest prefixlen has precedence
                if comp_a.value.prefixlen > comp_b.value.prefixlen:
                    return A_HAS_PRECEDENCE
                if comp_a.value.prefixlen < comp_b.value.prefixlen:
                    return B_HAS_PRECEDENCE
                # components equal -> continue with next component
                elif comp_a.value > comp_b.value:
                    return B_HAS_PRECEDENCE
                elif comp_a.value < comp_b.value:
                    return A_HAS_PRECEDENCE
            else:
                # assuming comp_a.value, comp_b.value of type bytearray
                if len(comp_a.value) == len(comp_b.value):
                    if comp_a.value > comp_b.value:
                        return B_HAS_PRECEDENCE
                    if comp_a.value < comp_b.value:
                        return A_HAS_PRECEDENCE
                    # components equal -> continue with next component
                else:
                    common = min(len(comp_a.value), len(comp_b.value))
                    if comp_a.value[:common] > comp_b.value[:common]:
                        return B_HAS_PRECEDENCE
                    elif comp_a.value[:common] < comp_b.value[:common]:
                        return A_HAS_PRECEDENCE
                    # the first common bytes match
                    elif len(comp_a.value) > len(comp_b.value):
                        return A_HAS_PRECEDENCE
                    else:
                        return A_HAS_PRECEDENCE
6. Validation Procedure

Flow Specifications received from a BGP peer that are accepted in the respective Adj-RIB-In are used as input to the route selection process. Although the forwarding attributes of two routes for the same Flow Specification prefix may be the same, BGP is still required to perform its path selection algorithm in order to select the correct set of attributes to advertise.

The first step of the BGP Route Selection procedure (Section 9.1.2 of [RFC4271]) is to exclude from the selection procedure routes that are considered non-feasible. In the context of IP routing information, this step is used to validate that the NEXT_HOP attribute of a given route is resolvable.

The concept can be extended, in the case of Flow Specification NLRI, to allow other validation procedures.

A Flow Specification NLRI must be validated such that it is considered feasible if and only if all of the below is true:

a) A destination prefix component is embedded in the Flow Specification.


c) There are no more specific unicast routes, when compared with the flow destination prefix, that has been received from a different neighboring AS than the best-match unicast route, which has been determined in rule b).

Rule a) MAY be relaxed by configuration, permitting Flow Specifications that include no destination prefix component. If such is the case, rules b) and c) are moot and MUST be disregarded.

By originator of a BGP route, we mean either the BGP originator path attribute, as used by route reflection, or the transport address of the BGP peer, if this path attribute is not present.

BGP implementations MUST also enforce that the AS_PATH attribute of a route received via the External Border Gateway Protocol (eBGP) contains the neighboring AS in the left-most position of the AS_PATH
attribute. While this rule is optional in the BGP specification, it
becomes necessary to enforce it for security reasons.

The best-match unicast route may change over the time independently
of the Flow Specification NLRI. Therefore, a revalidation of the
Flow Specification NLRI MUST be performed whenever unicast routes
change. Revalidation is defined as retesting that clause a and
clause b above are true.

Explanation:

The underlying concept is that the neighboring AS that advertises the
best unicast route for a destination is allowed to advertise flow-
spec information that conveys a more or equally specific destination
prefix. Thus, as long as there are no more specific unicast routes,
received from a different neighboring AS, which would be affected by
that filtering rule.

The neighboring AS is the immediate destination of the traffic
described by the Flow Specification. If it requests these flows to
be dropped, that request can be honored without concern that it
represents a denial of service in itself. Supposedly, the traffic is
being dropped by the downstream autonomous system, and there is no
added value in carrying the traffic to it.

7. Traffic Filtering Actions

This specification defines a minimum set of filtering actions that it
standardizes as BGP extended community values [RFC4360]. This is not
meant to be an inclusive list of all the possible actions, but only a
subset that can be interpreted consistently across the network.
Additional actions can be defined as either requiring standards or as
vendor specific.

Implementations SHOULD provide mechanisms that map an arbitrary BGP
community value (normal or extended) to filtering actions that
require different mappings in different systems in the network. For
instance, providing packets with a worse-than-best-effort, per-hop
behavior is a functionality that is likely to be implemented
differently in different systems and for which no standard behavior
is currently known. Rather than attempting to define it here, this
can be accomplished by mapping a user-defined community value to
platform-/network-specific behavior via user configuration.

The default action for a traffic filtering Flow Specification is to
accept IP traffic that matches that particular rule.
This document defines the following extended communities values shown in Table 2 in the form 0x8xnn where nn indicates the sub-type. Encodings for these extended communities are described below.

<table>
<thead>
<tr>
<th>Community</th>
<th>Action</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x8006</td>
<td>traffic-rate-bytes</td>
<td>2-byte ASN, 4-byte float</td>
</tr>
<tr>
<td>TBD</td>
<td>traffic-rate-packets</td>
<td>2-byte ASN, 4-byte float</td>
</tr>
<tr>
<td>0x8007</td>
<td>traffic-action</td>
<td>bitmask</td>
</tr>
<tr>
<td>0x8008</td>
<td>rt-redirect AS-2byte</td>
<td>2-octet AS, 4-octet value</td>
</tr>
<tr>
<td>0x8108</td>
<td>rt-redirect IPv4</td>
<td>4-octet IPv4 address, 2-octet</td>
</tr>
<tr>
<td>0x8208</td>
<td>rt-redirect AS-4byte</td>
<td>4-octet AS, 2-octet value</td>
</tr>
<tr>
<td>0x8009</td>
<td>traffic-marking</td>
<td>DSCP value</td>
</tr>
</tbody>
</table>

Table 2: Traffic Action Extended Communities

Some traffic action communities may interfere with each other. Section 7.6 of this specification provides general considerations on such traffic action interference. Any additional definition of a traffic actions specified by additional standards documents or vendor documents MUST specify if the traffic action interacts with an existing traffic actions, and provide error handling per [RFC7606].

Multiple traffic actions may be present for a single NLRI. The traffic actions are processed in ascending order of the sub-type found in the BGP Extended Communities. If not all of them can be processed the filter SHALL NOT be applied at all (for example: if for a given flow there are the action communities rate-limit-bytes and traffic-marking attached, and the platform does not support one of them also the other shall not be applied for that flow).

All traffic actions are specified as transitive BGP Extended Communities.

7.1. Traffic Rate in Bytes (traffic-rate-bytes) sub-type 0x06

The traffic-rate-bytes extended community uses the following extended community encoding:

The first two octets carry the 2-octet id, which can be assigned from a 2-byte AS number. When a 4-byte AS number is locally present, the 2 least significant bytes of such an AS number can be used. This value is purely informational and SHOULD NOT be interpreted by the implementation.
The remaining 4 octets carry the maximum rate information in IEEE floating point [IEEE.754.1985] format, units being bytes per second. A traffic-rate of 0 should result on all traffic for the particular flow to be discarded. On encoding the traffic-rate MUST NOT be negative. On decoding negative values MUST be treated as zero (discard all traffic).

Interferes with: No other BGP Flow Specification traffic action in this document.

7.2. Traffic Rate in Packets (traffic-rate-packets) sub-type TBD

The traffic-rate-packets extended community uses the same encoding as the traffic-rate-bytes extended community. The floating point value carries the maximum packet rate in packets per second. A traffic-rate-packets of 0 should result in all traffic for the particular flow to be discarded. On encoding the traffic-rate-packets MUST NOT be negative. On decoding negative values MUST be treated as zero (discard all traffic).

Interferes with: No other BGP Flow Specification traffic action in this document.

7.3. Traffic-action (traffic-action) sub-type 0x07

The traffic-action extended community consists of 6 bytes of which only the 2 least significant bits of the 6th byte (from left to right) are currently defined.

```
+---+---+---+---+---+---+---+---+
|        reserved       | S | T |
+---+---+---+---+---+---+---+---+
```

where S and T are defined as:

- T: Terminal Action (bit 47): When this bit is set, the traffic filtering engine will apply any subsequent filtering rules (as defined by the ordering procedure). If not set, the evaluation of the traffic filter stops when this rule is applied.

- S: Sample (bit 46): Enables traffic sampling and logging for this Flow Specification.

- reserved: should always be set to 0 by the originator and not be evaluated by the receiving BGP speaker.
The use of the Terminal Action (bit 47) may result in more than one filter-rule matching a particular flow. All the flow actions from these rules shall be collected and applied. In case of interfering traffic actions it is an implementation decision which actions are selected. See also Section 7.6.

Interferes with: No other BGP Flow Specification traffic action in this document.

7.4. RT Redirect (rt-redirect) sub-type 0x08

The redirect extended community allows the traffic to be redirected to a VRF routing instance that lists the specified route-target in its import policy. If several local instances match this criteria, the choice between them is a local matter (for example, the instance with the lowest Route Distinguisher value can be elected). This extended community allows 3 different encodings formats for the route-target (type 0x80, 0x81, 0x82). It uses the same encoding as the Route Target extended community [RFC4360].

It should be noted that the low-order nibble of the Redirect’s Type field corresponds to the Route Target Extended Community format field (Type). (See Sections 3.1, 3.2, and 4 of [RFC4360] plus Section 2 of [RFC5668].) The low-order octet (Sub-Type) of the Redirect Extended Community remains 0x08 for all three encodings of the BGP Extended Communities (AS 2-byte, AS 4-byte, and IPv4 address).

Interferes with: All other redirect functions.

7.5. Traffic Marking (traffic-marking) sub-type 0x09

The traffic marking extended community instructs a system to modify the DSCP bits of a transiting IP packet to the corresponding value. This extended community is encoded as a sequence of 5 zero bytes followed by the DSCP value encoded in the 6 least significant bits of 6th byte.

Interferes with: No other BGP Flow Specification traffic action in this document.

7.6. Considerations on Traffic Action Interference

Since traffic actions are represented as BGP extended community values, traffic actions may interfere with each other (i.e. there may be more than one conflicting traffic-rate action associated with a single flow-filter). Traffic action interference has no impact on BGP propagation of flow filters (all communities are propagated according to policies).
If a flow filter associated with interfering flow actions is selected for packet forwarding, it is a implementation decision which of the interfering traffic actions are selected. Implementors of this specification SHOULD document the behaviour of their implementation in such cases.

If required, operators are encouraged to make use of the BGP policy framework supported by their implementation in order to achieve a predictable behaviour (ie. match - replace - delete communities on administrative boundaries).

8. Dissemination of Traffic Filtering in BGP/MPLS VPN Networks

Provider-based Layer 3 VPN networks, such as the ones using a BGP/MPLS IP VPN [RFC4364] control plane, may have different traffic filtering requirements than Internet service providers. But also Internet service providers may use those VPNs for scenarios like having the Internet routing table in a VRF, resulting in the same traffic filtering requirements as defined for the global routing table environment within this document. This document proposes an additional BGP NLRI type (AFI=1, SAFI=134) value, which can be used to propagate traffic filtering information in a BGP/MPLS VPN environment.

The NLRI format for this address family consists of a fixed-length Route Distinguisher field (8 bytes) followed by a Flow Specification, following the encoding defined above in Section 4.2 of this document. The NLRI length field shall include both the 8 bytes of the Route Distinguisher as well as the subsequent Flow Specification.

```
+------------------------------+
| length (0xnn or 0xfn nn)    |
+------------------------------+
| Route Distinguisher (8 bytes) |
+------------------------------+
|     NLRI value (variable)    |
+------------------------------+
```

Flow-spec NLRI for MPLS

Propagation of this NLRI is controlled by matching Route Target extended communities associated with the BGP path advertisement with the VRF import policy, using the same mechanism as described in "BGP/MPLS IP VPNs" [RFC4364].

Flow Specification rules received via this NLRI apply only to traffic that belongs to the VRF(s) in which it is imported. By default,
traffic received from a remote PE is switched via an MPLS forwarding
decision and is not subject to filtering.

Contrary to the behavior specified for the non-VPN NLRI, flow rules
are accepted by default, when received from remote PE routers.

8.1. Validation Procedures for BGP/MPLS VPNs

The validation procedures are the same as for IPv4.

8.2. Traffic Actions Rules

The traffic action rules are the same as for IPv4.

9. Limitations of Previous Traffic Filtering Efforts

9.1. Limitations in Previous DDoS Traffic Filtering Efforts

The popularity of traffic-based, denial-of-service (DoS) attacks,
which often requires the network operator to be able to use traffic
filters for detection and mitigation, brings with it requirements
that are not fully satisfied by existing tools.

Increasingly, DoS mitigation requires coordination among several
service providers in order to be able to identify traffic source(s)
and because the volumes of traffic may be such that they will
otherwise significantly affect the performance of the network.

Several techniques are currently used to control traffic filtering of
DoS attacks. Among those, one of the most common is to inject
unicast route advertisements corresponding to a destination prefix
being attacked (commonly known as remote triggered blackhole RTBH).
One variant of this technique marks such route advertisements with a
community that gets translated into a discard Next-Hop by the
receiving router. Other variants attract traffic to a particular
node that serves as a deterministic drop point.

Using unicast routing advertisements to distribute traffic filtering
information has the advantage of using the existing infrastructure
and inter-AS communication channels. This can allow, for instance, a
service provider to accept filtering requests from customers for
address space they own.

There are several drawbacks, however. An issue that is immediately
apparent is the granularity of filtering control: only destination
prefixes may be specified. Another area of concern is the fact that
filtering information is intermingled with routing information.
The mechanism defined in this document is designed to address these limitations. We use the Flow Specification NLRI defined above to convey information about traffic filtering rules for traffic that is subject to modified forwarding behavior (actions). The actions are defined as extended communities and include (but are not limited to) rate-limiting (including discard), traffic redirection, packet rewriting.

9.2. Limitations in Previous BGP/MPLS Traffic Filtering Efforts

Provider-based Layer 3 VPN networks, such as the ones using a BGP/MPLS IP VPN [RFC4364] control plane, may have different traffic filtering requirements than Internet service providers.

In these environments, the VPN customer network often has traffic filtering capabilities towards their external network connections (e.g., firewall facing public network connection). Less common is the presence of traffic filtering capabilities between different VPN attachment sites. In an any-to-any connectivity model, which is the default, this means that site-to-site traffic is unfiltered.

In circumstances where a security threat does get propagated inside the VPN customer network, there may not be readily available mechanisms to provide mitigation via traffic filter.

But also Internet service providers may use those VPNs for scenarios like having the Internet routing table in a VRF. Therefore, limitations described in Section 9.1 also apply to this section.

The BGP Flow Specification addresses these limitations.

10. Traffic Monitoring

Traffic filtering applications require monitoring and traffic statistics facilities. While this is an implementation-specific choice, implementations SHOULD provide:

- A mechanism to log the packet header of filtered traffic.
- A mechanism to count the number of matches for a given flow specification rule.

11. Error-Handling and Future NLRI Extensions

In case BGP encounters an error in a Flow Specification UPDATE message it SHOULD treat this message as Treat-as-withdraw according to [RFC7606] Section 2.
Possible reasons for an error are (for more reasons see also [RFC7606]):

- Incorrect implementation of this specification - the encoding/decoding of the NLRI or traffic action extended-communities do not comply with this specification.
- Unknown Flow Specification extensions - The sending party has implemented a Flow Specification NLRI extension unknown to the receiving party.

In order to facilitate future extensions of the Flow Specification NLRI, such extensions SHOULD specify a way to encode a "always-true" match condition within the newly introduced components. This match condition can be used to propagate (and apply) certain filters only if a specific extension is known to the implementation.

12. IANA Considerations

This section complies with [RFC7153].

12.1. AFI/SAFI Definitions

IANA maintains a registry entitled "SAFI Values". For the purpose of this work, IANA updated the registry and allocated two additional SAFIs:

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>133</td>
<td>IPv4 dissemination of Flow Specification</td>
<td>[this document]</td>
</tr>
<tr>
<td></td>
<td>rules</td>
<td></td>
</tr>
<tr>
<td>134</td>
<td>VPNv4 dissemination of Flow Specification</td>
<td>[this document]</td>
</tr>
<tr>
<td></td>
<td>rules</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Registry: SAFI Values

12.2. Flow Component Definitions

A Flow Specification consists of a sequence of flow components, which are identified by an 8-bit component type. IANA has created and maintains a registry entitled "Flow Spec Component Types". This document defines the following Component Type Codes:
<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Destination Prefix</td>
<td>[this document]</td>
</tr>
<tr>
<td>2</td>
<td>Source Prefix</td>
<td>[this document]</td>
</tr>
<tr>
<td>3</td>
<td>IP Protocol</td>
<td>[this document]</td>
</tr>
<tr>
<td>4</td>
<td>Port</td>
<td>[this document]</td>
</tr>
<tr>
<td>5</td>
<td>Destination port</td>
<td>[this document]</td>
</tr>
<tr>
<td>6</td>
<td>Source port</td>
<td>[this document]</td>
</tr>
<tr>
<td>7</td>
<td>ICMP type</td>
<td>[this document]</td>
</tr>
<tr>
<td>8</td>
<td>ICMP code</td>
<td>[this document]</td>
</tr>
<tr>
<td>9</td>
<td>TCP flags</td>
<td>[this document]</td>
</tr>
<tr>
<td>10</td>
<td>Packet length</td>
<td>[this document]</td>
</tr>
<tr>
<td>11</td>
<td>DSCP</td>
<td>[this document]</td>
</tr>
<tr>
<td>12</td>
<td>Fragment</td>
<td>[this document]</td>
</tr>
</tbody>
</table>

Table 4: Registry: Flow Spec Component Types

In order to manage the limited number space and accommodate several usages, the following policies defined by [RFC8126] used:

<table>
<thead>
<tr>
<th>Range</th>
<th>Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Invalid value</td>
</tr>
<tr>
<td>[1 .. 12]</td>
<td>Defined by this specification</td>
</tr>
<tr>
<td>[13 .. 127]</td>
<td>Specification required</td>
</tr>
<tr>
<td>[128 .. 255]</td>
<td>First Come First Served</td>
</tr>
</tbody>
</table>

Table 5: Flow Spec Component Types Policies

The specification of a particular "Flow Spec Component Type" must clearly identify what the criteria used to match packets forwarded by the router is. This criteria should be meaningful across router hops and not depend on values that change hop-by-hop such as TTL or Layer 2 encapsulation.

12.3. Extended Community Flow Specification Actions

The Extended Community Flow Specification Action types defined in this document consist of two parts:

Type (BGP Transitive Extended Community Type)

Sub-Type
For the type-part, IANA maintains a registry entitled "BGP Transitive Extended Community Types". For the purpose of this work (Section 7), IANA updated the registry to contain the values listed below:

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x80</td>
<td>Generic Transitive Experimental Use Extended Community</td>
<td>[RFC7153]</td>
</tr>
<tr>
<td></td>
<td>(Sub-Types are defined in the &quot;Generic Transitive Experimental Use</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Extended Community Sub-Types&quot; registry)</td>
<td></td>
</tr>
<tr>
<td>0x81</td>
<td>Generic Transitive Experimental Use Extended Community Part 2</td>
<td>[this document]</td>
</tr>
<tr>
<td></td>
<td>(Sub-Types are defined in the &quot;Generic Transitive Experimental Use</td>
<td>[See Note-1]</td>
</tr>
<tr>
<td></td>
<td>Extended Community Part 2 Sub-Types&quot; Registry)</td>
<td></td>
</tr>
<tr>
<td>0x82</td>
<td>Generic Transitive Experimental Use Extended Community Part 3</td>
<td>[this document]</td>
</tr>
<tr>
<td></td>
<td>(Sub-Types are defined in the &quot;Generic Transitive Experimental Use</td>
<td>[See Note-1]</td>
</tr>
<tr>
<td></td>
<td>Extended Community Part 3 Sub-Types&quot; Registry)</td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Registry: Generic Transitive Experimental Use Extended Community Types

Note-1: This document obsoletes RFC7674.

For the sub-type part of the extended community actions IANA maintains and updated the following registries:
<table>
<thead>
<tr>
<th>Sub-Type Value</th>
<th>Name</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x06</td>
<td>Flow spec traffic-rate-bytes</td>
<td>[this document]</td>
</tr>
<tr>
<td>TBD</td>
<td>Flow spec traffic-rate-packets</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x07</td>
<td>Flow spec traffic-action (Use of the &quot;Value&quot; field is defined in the &quot;Traffic Action Fields&quot; registry)</td>
<td>[this document] [See Note-2]</td>
</tr>
<tr>
<td>0x08</td>
<td>Flow spec rt-redirect AS-2byte format</td>
<td>[this document]</td>
</tr>
<tr>
<td>0x09</td>
<td>Flow spec traffic-remarking</td>
<td>[this document]</td>
</tr>
</tbody>
</table>

Table 7: Registry: Generic Transitive Experimental Use Extended Community Sub-Types

Note-2: This document obsoletes both RFC7674 and RFC5575.

<table>
<thead>
<tr>
<th>Sub-Type Value</th>
<th>Name</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x08</td>
<td>Flow spec rt-redirect IPv4 format</td>
<td>[this document] [See Note-3]</td>
</tr>
</tbody>
</table>

Table 8: Registry: Generic Transitive Experimental Use Extended Community Part 2 Sub-Types

<table>
<thead>
<tr>
<th>Sub-Type Value</th>
<th>Name</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x08</td>
<td>Flow spec rt-redirect AS-4byte format</td>
<td>[this document] [See Note-3]</td>
</tr>
</tbody>
</table>

Table 9: Registry: Generic Transitive Experimental Use Extended Community Part 3 Sub-Types

Note-3: This document obsoletes RFC7674, and becomes the only reference for this table.
The "traffic-action" extended community (Section 7.3) defined in this document has 46 unused bits, which can be used to convey additional meaning. IANA created and maintains a new registry entitled: "Traffic Action Fields". These values should be assigned via IETF Review rules only. The following traffic-action fields have been allocated:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Name</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>47</td>
<td>Terminal Action</td>
<td>[this document]</td>
</tr>
<tr>
<td>46</td>
<td>Sample</td>
<td>[this document]</td>
</tr>
</tbody>
</table>

Table 10: Registry: Traffic Action Fields

13. Security Considerations

Inter-provider routing is based on a web of trust. Neighboring autonomous systems are trusted to advertise valid reachability information. If this trust model is violated, a neighboring autonomous system may cause a denial-of-service attack by advertising reachability information for a given prefix for which it does not provide service.

As long as traffic filtering rules are restricted to match the corresponding unicast routing paths for the relevant prefixes, the security characteristics of this proposal are equivalent to the existing security properties of BGP unicast routing. However, this document also specifies traffic filtering actions that may need custom additional verification on the receiver side. See Section 14.

Where it is not the case, this would open the door to further denial-of-service attacks.

Enabling firewall-like capabilities in routers without centralized management could make certain failures harder to diagnose. For example, it is possible to allow TCP packets to pass between a pair of addresses but not ICMP packets. It is also possible to permit packets smaller than 900 or greater than 1000 bytes to pass between a pair of addresses, but not packets whose length is in the range 900-1000. Such behavior may be confusing and these capabilities should be used with care whether manually configured or coordinated through the protocol extensions described in this document.
14. Operational Security Considerations

While the general verification of the traffic filter NLRI is specified in this document (Section 6) the traffic filtering actions received by a third party may need custom verification or filtering. In particular all non traffic-rate actions may allow a third party to modify packet forwarding properties and potentially gain access to other routing-tables/VPNs or undesired queues. This can be avoided by proper filtering of action communities at network borders and by mapping user-defined communities (see Section 7) to expose certain forwarding properties to third parties.

Since verification of the traffic filtering NLRI is tied to the announcement of the best unicast route, a unfiltered address space hijack (e.g. advertisement of a more specific route) may cause this verification to fail and consequently prevent Flow Specification filters from being accepted by a peer.

15. Original authors

Barry Greene, Pedro Marques, Jared Mauch, Danny McPherson, and Nischal Sheth were authors on RFC5575, and therefore are contributing authors on this document.

16. Acknowledgements

The authors would like to thank Yakov Rekhter, Dennis Ferguson, Chris Morrow, Charlie Kaufman, and David Smith for their comments for the comments on the original RFC5575. Chaitanya Kodeboyina helped design the flow validation procedure; and Steven Lin and Jim Washburn ironed out all the details necessary to produce a working implementation in the original RFC5575.

A packet rate flowspec action was also described in a flowspec extention draft and the authors like to thank Wesley Eddy, Justin Dailey and Gilbert Clark for their work.

Additional the authors would like to thank Alexander Mayrhofer, Nicolas Fevrier, Job Snijders, Jeffrey Haas and Adam Chappell for their comments and review.

17. References

17.1. Normative References

[IEEE.754.1985]
17.2. Informative References

[I-D.ietf-idr-flow-spec-v6]
McPherson, D., Raszuk, R., Pithawala, B.,
akarch@cisco.com, a., and S. Hares, "Dissemination of Flow Specification Rules for IPv6",
draft-ietf-idr-flow-spec-v6-09 (work in progress), November 2017.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)",
RFC 4303, DOI 10.17487/RFC4303, December 2005,

17.3. URIs


Appendix A. Comparison with RFC 5575

This document includes numerous editorial changes to RFC5575. It is recommended to read the entire document. The authors, however want to point out the following technical changes to RFC5575:

Section 1 introduces the Flow Specification NLRI. In RFC5575 this NLRI was defined as an opaque-key in BGP's database. This specification has removed all references to a opaque-key property. BGP is able understand the NLRI encoding. This change also resulted in a new section regarding error-handling and extensibility (Section 11).

Section 4.2.3 defines a numeric operator and comparison bit combinations. In RFC5575 the meaning of those bit combination was not explicitly defined and left open to the reader.

Section 4.2.3 - Section 4.2.8, Section 4.2.10, Section 4.2.11 make use of the above numeric operator. The allowed length of the comparison value was not consistently defined in RFC5575.

Section 7 defines all traffic action extended communities as transitive extended communities. RFC5575 defined the traffic-rate
action to be non-transitive and did not define the transitivity of the other action communities at all.

Section 7.2 introduces a new traffic filtering action (traffic-rate-packets). This action did not exist in RFC5575.

Section 7.4 contains the same redirect actions already defined in RFC5575 however, these actions have been renamed to "rt-redirect" to make it clearer that the redirection is based on route-target.

Section 7.6 contains general considerations on interfering traffic actions. Section 7.3 also cross-references this section. RFC5575 did not mention this.

Section 11 contains a modified error handling to gracefully allow future extensions of flow specification.

Authors’ Addresses

Christoph Loibl
Next Layer Communications
Mariahilfer Guertel 37/7
Vienna 1150
AT

Phone: +43 664 1176414
Email: cl@tix.at

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI  48176
USA

Email: shares@ndzh.com

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

Email: robert@raszuk.net
Danny McPherson
Verisign
USA

Email: dmcpherson@verisign.com

Martin Bacher
T-Mobile Austria
Rennweg 97-99
Vienna 1030
AT

Email: mb.ietf@gmail.com
Making Route Servers Aware of Data Link Failures at IXPs

draft-ietf-idr-rs-bfd-07

Abstract

When BGP route servers are used, the data plane is not congruent with the control plane. Therefore, peers at an Internet exchange can lose data connectivity without the control plane being aware of it, and packets are lost. This document proposes the use of a newly defined BGP Subsequent Address Family Identifier (SAFI) both to allow the route server to request its clients use BFD to track data plane connectivity to their peers' addresses, and for the clients to signal that connectivity state back to the route server.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in [RFC2119] only when they appear in all upper case. They may also appear in lower or mixed case as English words, without normative meaning.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."
1. Introduction

In configurations (typically Internet Exchange Points (IXPs)) where EBGP routing information is exchanged between client routers through the agency of a route server (RS) [RFC7947], but traffic is exchanged directly, operational issues can arise when partial data plane connectivity exists among the route server client routers. Since the
data plane is not congruent with the control plane, the client routers on the IXP can lose data connectivity without the control plane - the route server - being aware of it, resulting in significant data loss.

To remedy this, two basic problems need to be solved:

1. Client routers must have a means of verifying connectivity amongst themselves, and
2. Client routers must have a means of communicating the knowledge of the failure (and restoration) back to the route server.

The first can be solved by application of Bidirectional Forwarding Detection [RFC5880]. The second can be solved by exchanging BGP routes which use the NH-Reach Subsequent Address Family Identifier (SAFI) defined in this document.

Throughout this document, we generally assume that the route server being discussed is able to represent different RIBs towards different clients, as discussed in section 2.3.2.1 of [RFC7947]. If this is not the case, the procedures described here to allow BFD to be automatically provisioned between clients still have value; however, the procedures for signaling reachability back to the route server may not.

Throughout this document, we refer to the "route server", "RS" or just "server" and the "client" to describe the two BGP routers engaging in the exchange of information. We observe that there could be other applications for this extension. Our use of terminology is intended for clarity of description, and not to limit the future applicability of the proposal.

[I-D.ietf-idr-bgp-bestpath-selection-criteria] discusses enhancement of the route resolvability condition of section 9.1.2.1 of [RFC4271] to include next hop reachability and path availability checks. This specification represents in part an instance of such, implemented using BFD as the OAM mechanism.

2. Definitions

- Indirect peer: If a route server is configured such that routes from a given client might be sent to some other client, or vice-versa, those two clients are considered to be indirect peers.
- Indirect Peer’s Address, IPA, next hop: We refer frequently to a next hop. It should generally be clear from context what is intended, almost always an address associated with an indirect peer (the exception, when an indirect peer sends a third party next hop, is discussed in Section 3). In Section 5 we discuss the
MP-BGP [RFC4760] Next Hop field; this is distinguished by its capitalization and should also be clear from context. Later in that section we define the Indirect Peer’s Address field of the NLRI, also called "IPA". It will be clear to the reader that this refers to the "next hops" discussed elsewhere in the document, but we don’t use the name "next hop" for this field to avoid confusion with the pre-existing next hop path attribute of [RFC4271] and attribute field of [RFC4760].

- RS: Route Server. See [RFC7947].

3. Overview

As with the base BGP protocol, we model the function of this extension as the interaction between a conceptual set of databases:

- ReachAsk: The reachability request database. A database of next hops (host addresses) for which data plane reachability is being queried.
- ReachAsk-Out: A set of queries sent to the client.
- ReachAsk-In: A set of queries received from the route server.
- ReachTell: The reachability response database. A database of responses to ReachAsk queries, indicating what is known about data plane reachability.
- ReachTell-Out: The responses being sent to the route server.
- ReachTell-In: The response received from the client.
- LocReach: The local reachability database.
- NHIB: Next Hop Information Base. Stores what is known about the client’s reachability to its next hops.
In outline, the route server requests its client to track connectivity for all the potential next hops the RS might send to the client, by sending these next hops as ReachAsk "routes". The client tracks connectivity using BFD and reports its connectivity status to the RS using ReachTell "routes". Connectivity status may be that the next hop is reachable, unreachable, or unknown. Once the RS has been informed by the client of its connectivity, it uses this information to influence the route selection the RS performs on behalf of the client. Details are elaborated in the following sections.

4. Next Hop Validation

Below, we detail procedures where a route server tells its client router about other client next hops by sending it ReachAsk routes and the client router verifies connectivity to those other client routers and communicates its findings back to the RS using ReachTell routes. The RS uses the received ReachTell routes as input to the NHIB and hence the route selection process it performs on behalf of the client.
4.1. ReachAsk

The route server maintains a ReachAsk database for each client that supports this proposal, that is, for each client that has advertised support (Section 5) for the NH-Reach SAFI. This database is the union of:

- The set of next hops found in the associated per-client Loc-RIB (see section 2.3.2.1 of [RFC7947]).
- The set of addresses of this client’s indirect peers (Section 2).
- The RS MAY also add other entries, for example under configuration control.

We note that under most circumstances, the first (Loc-RIB next hops) set will be a subset of the second (indirect peers) set. For this not to be the case, a client would have to have sent a "third party" next hop [RFC4271] to the server. To cover such a case, an implementation MAY note any such next hops, and include them in its list of indirect peers. (This implies that if a third party next hop for client C is conveyed to client A, not only will C be placed in A’s ReachAsk database, but A will be placed in C’s ReachAsk database.)

The contents of the ReachAsk database are communicated to the client using the NLRI format and procedures described in Section 5.

4.2. LocReach

The client MUST attempt to track data plane connectivity to each host address depicted in the ReachAsk database. It MAY also track connectivity to other addresses. The use of BFD for this purpose is detailed in Section 6.

For each address being tracked, its state is maintained by the client in a LocReach entry. The state can be:

- Unknown. Connectivity status is unknown. This may be due to a temporary or permanent lack of feasible OAM mechanism to determine the status.
- Up. The address has been determined to be reachable.
- Down. The address has been determined to be unreachable.

The LocReach database is used as input for the ReachTell database; it MAY also be used as input to the client’s route resolvability condition (section 9.1.2.1 of [RFC4271]).
4.3. ReachTell

The ReachTell database contains an entry for every entry in the LocReach database.

The contents of the ReachTell database are communicated to the server using the NLRI format and procedures described in Section 5.

4.4. NHIB

The route server maintains a per-client Next Hop Information Base, or NHIB. This contains the information about next hop status received from ReachTell.

In computing its per-client Loc-RIB, the RS uses the content of the related per-client NHIB as input to the route resolvability condition (section 9.1.2.1 of [RFC4271]). The next hop being resolved is looked up in the NHIB and its state determined:

- Up next hops are considered resolvable.
- Unknown next hops MAY be considered resolvable. They MAY be less preferred for selection.
- Down next hops MUST NOT be considered resolvable.
- If a given next hop is not present in the NHIB, but is present in ReachAsk-Out, either the client has not responded yet (a transient condition) or an error exists. Similar to Unknown next hops, such routes MAY be considered resolvable; they MAY be less preferred.

5. Advertising NH-Reach state in BGP

A new BGP SAFI, the NH-Reach SAFI, is defined in this document. It has been assigned value TBD. A route server or a route server client using the procedures in this document MUST advertise support for this SAFI, for the IPv4 and/or IPv6 Address Family Identifier (AFI). The use of this SAFI with any other AFI is not defined by this document.

NH-Reach NLRI "routes" have a Length of Next Hop Network Address value of 0, therefore they have an empty Network Address of Next Hop field (section 3 of [RFC4760]).

Since as specified here, ReachTell "routes" from different clients populate distinct databases on the RS, there will generally be only a single path per "route"; this implies that route selection need not be performed (or equivalently, that it’s trivial to perform).

In the other direction, a client might peer with multiple route servers and receive differing sets of ReachAsk routes from them. An implementation MAY handle this situation by implementing a distinct
ReachAsk and ReachTell per server, but it MAY also handle it by placing all servers' ReachAsk "routes" into a single ReachAsk, and sending the results to all servers from a single ReachTell. This would imply some route server(s) might get ReachTell results they had not asked for, but this is permissible in any case. Again, since the contents of ReachAsk are simply a set of host routes to be tested, route selection over a combined ReachAsk MAY be omitted.

ReachAsk and ReachTell entries are exchanged using the NH-Reach NLRI encoding:

```
+---------------------------------------------+            
|T|Reserved|Sta|Indirect Peer's Address (4 or 16 octets) |
+---------------------------------------------+            
... Indirect Peer's Address (4 or 16 octets) ...
```

**NH-Reach NLRI Format**

- **T**: Type is a one-bit field that can take the value 0, meaning the NLRI is a ReachAsk entry, or 1, meaning it is a ReachTell entry.
- **Reserved**: These five bits are reserved. They MUST be sent as zero and MUST be disregarded on receipt.
- **Sta**: State is a two-bit field used to signal the LocReach (Section 4.2) state:
  - 0 or 3: Unknown.
  - 1: Up.
  - 2: Down.

Although either 0 or 3 is to be interpreted as "Unknown", the value 0 MUST be used on transmission. The value 3 MUST be accepted as an alias for 0 on receipt.

- The Indirect Peer's Address ("IPA") field is an IPv4 or IPv6 host route, depending on whether the AFI is IPv4 or IPv6.

ReachAsk and ReachTell entries MUST NOT be propagated from one BGP peering session to another; the routes are not transitive.

The IPA field is the key for the NH-Reach NLRI type; the information encoded in the top octet is non-key information. It is possible in principle (although unlikely) for two NLRI to be validly present in an UPDATE message with identical IPA fields but different types. However, two NLRI with the same IPA field and different State fields MUST NOT be encoded in the same UPDATE message. If such is
encountered, the receiver MUST behave as though the state "Unknown" was received for the IPA in question.

6. Client Procedures for NH-Reach Changes

When an entry is added to a route server client’s ReachAsk-In for a route server peering session, the client will then attempt to verify connectivity to the host depicted by that entry. The procedure described in this specification utilizes BFD.

If no existing BFD session exists to this next hop, a BFD session is provisioned to that IP address and the LocReach reachability state (Section 4.2) is set to Unknown.

If the client cannot establish a BFD session with an entry in its ReachAsk-In, the next hop remains in LocReach with its Reachable state Unknown.

Once the BFD session moves to the Up state, the LocReach reachability state is set to Up.

When the BFD session transitions out of the Up state to the Down state, the LocReach reachability state is set to Down.

If the BFD session transitions out of the Up state to the AdminDown state, the LocReach reachability state is set to Unknown.

When entries are removed from the route server client’s ReachAsk-In for a route server peering session, the client MAY delay de-provisioning the BFD peering session. If the client delays de-provisioning the session, it should remove it if the BFD session transitions to the Down or AdminDown states.

7. Recommendations for Using BFD

The RECOMMENDED way a client router can confirm the data plane connectivity to its next hops is available, is the use of BFD in asynchronous mode. Echo mode MAY be used if both client routers running a BFD session support this. The use of authentication in BFD is OPTIONAL as there is a certain level of trust between the operators of the client routers at a particular IXP. If trust cannot be assumed, it is recommended to use pair-wise keys (how this can be achieved is outside the scope of this document). The ttl/hop limit values as described in section 5 [RFC5881] MUST be obeyed in order to shield BFD sessions against packets coming from outside the IXP.

The following values of the BFD configuration of client routers (see section 6.8.1 [RFC5880]) are RECOMMENDED:
A client router administrator MAY select more appropriate values to meet the special needs of a particular deployment.

8. Other Considerations

For purposes of routing stability, implementations may wish to apply hysteresis ("holddown") to next hops that have transitioned from reachable to unreachable and back.

Implementations MAY restrict the range of addresses with which they will attempt to form BFD relationships. For example, an implementation might by default only allow BFD relationships with peers that share a subnetwork with the route server. An implementation MAY apply such restrictions by default.

In a route-server environment, use of this feature SHOULD be restricted to consider only routes that are advertised from within the IXP network. This might include checks on AS_PATH length.

9. Acknowledgments

The authors would like to thanks Thomas King for his contributions toward this work.

10. IANA Considerations

IANA is requested to allocate a value from the Subsequent Address Family Identifiers (SAFI) Parameters registry for this proposal. Its Description in that registry shall be NH-Reach with a Reference of this RFC.

11. Security Considerations

The mechanism in this document permits a route server client to influence the contents of the route server’s Adj-Ribs-Out through its reports of next hop reachability state using the NH-Reach SAFI. Since this state is per-client, if a route server client is able to inject NH-Reach routes for another route server’s BGP session to a client, it can cause the route server to select different forwarding than otherwise expected. This issue may be mitigated using transport security on the BGP sessions between the route server and its clients. See [RFC4272].
The NH-Reach SAFI enables the server to trigger creation of a BFD session on its client. A malicious or misbehaving server could trigger an unreasonable number of sessions, a potential resource exhaustion attack. The sedate default timers proposed in Section 7 mitigate this; they also mitigate concerns about use of the client as a source of packets in a flooding attack. An implementation MAY also impose limits on the number of BFD sessions it will create at the request of the server.

The reachability tests between route server clients themselves may be a target for attack. Such attacks may include forcing a BFD session Down through injecting false BFD state. A less likely attack includes forcing a BFD session to stay Up when its real state is Down. These attacks may be mitigated using the BFD security mechanisms defined in [RFC5880].

12. References

12.1. Normative References


12.2. Informative References

[I-D.chen-bfd-unsolicited]

[I-D.ietf-idr-bgp-bestpath-selection-criteria]


Appendix A. Summary of Document Changes

idr-06: Refresh -05.
idr-04 to idr-05: Added reference to "BGP Bestpath Selection Criteria Enhancement" draft. Rename "next hop" field of NLRI to "Indirect Peer’s Address". Add suggestion about AS_PATH length checks.
idr-03 to idr-04: Note other forms of connectivity checks.
idr-02 to idr-03: Substantial rewrite. Introduce NLRI format that embeds state.
idr-01 to idr-02: Move from BGP-LS to NH-Reach SAFI. Lots of editorial changes.
idr-00 to idr-01: Add BGP Capability. Move from NH-Cost to BGP-LS.
ymbk-01 to idr-00: No technical changes; adopted by IDR.
ymbk-00 to ymbk-01: Clarifications to BFD procedures. Use BFD state as an input to BGP route selection.

Appendix B. Other Forms of Connectivity Checks

RFC 5880/5881 BFD is a well-deployed feature. For this reason, it was chosen as the connectivity check utilized for nexthop reachability by this document. As other forms of BFD become more widely deployed, they may also be utilized to provide the connectivity check functionality.

Examples of other such BFD mechanisms include:
Internet-Draft Making RSes aware of IXP Data Link Failures    March 2019

- Seamless BFD [RFC7880]
- Unsolicited BFD for Sessionless Applications
  [I-D.chen-bfd-unsolicited]

Implementations MUST support RFC 5880/5881 BFD to be compliant with this specification. Implementations MAY support other forms of connectivity check, including those mechanisms listed above, so long as they provide the ability to fall-back to RFC 5880/5881 BFD.

Authors’ Addresses

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

Email: randy@psg.com

Jeffrey Haas
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA  94089
US

Email: jhaas@juniper.net

John G. Scudder
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA  94089
US

Email: jgs@juniper.net

Arnold Nipper
DE-CIX Management GmbH
Lichtstrasse 43i
Cologne  50825
Germany

Email: arnold.nipper@de-cix.net
Christoph Dietzel
DE-CIX Management GmbH
Lichtstrasse 43i
Cologne  50825
Germany

Email: christoph.dietzel@de-cix.net
The BGP Tunnel Encapsulation Attribute
draft-ietf-idr-tunnel-encaps-13.txt

Abstract

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

This document obsoletes RFC 5512.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."
This Internet-Draft will expire on January 23, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the
document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal
Provisions Relating to IETF Documents
(https://trustee.ietf.org/license-info) in effect on the date of
publication of this document. Please review these documents
carefully, as they describe your rights and restrictions with respect
to this document. Code Components extracted from this document must
include Simplified BSD License text as described in Section 4.e of
the Trust Legal Provisions and are provided without warranty as
described in the Simplified BSD License.

Table of Contents

1. Introduction .................................................. 3
   1.1. Brief Summary of RFC 5512 ............................... 4
   1.2. Deficiencies in RFC 5512 ............................... 4
   1.3. Brief Summary of Changes from RFC 5512 ................ 5
   1.4. Impact on RFC 5566 ..................................... 6
2. The Tunnel Encapsulation Attribute .......................... 6
3. Tunnel Encapsulation Attribute Sub-TLVs .................... 8
   3.1. The Tunnel Endpoint Sub-TLV ............................. 8
   3.2. Encapsulation Sub-TLVs for Particular Tunnel Types .... 10
      3.2.1. VXLAN ............................................. 10
      3.2.2. VXLAN-GPE ....................................... 12
      3.2.3. NVGRE ........................................... 13
      3.2.4. L2TPv3 .......................................... 14
      3.2.5. GRE ............................................... 15
      3.2.6. MPLS-in-GRE ..................................... 15
      3.2.7. IP-in-IP ......................................... 16
   3.3. Outer Encapsulation Sub-TLVs ........................... 16
      3.3.1. IPv4 DS Field .................................... 16
      3.3.2. UDP Destination Port .............................. 17
   3.4. Sub-TLVs for Aiding Tunnel Selection ................... 17
      3.4.1. Protocol Type Sub-TLV ............................. 17
      3.4.2. Color Sub-TLV ................................... 17
   3.5. Embedded Label Handling Sub-TLV ........................ 18
   3.6. MPLS Label Stack Sub-TLV ................................ 19
   3.7. Prefix-SID Sub-TLV .................................... 20
4. Extended Communities Related to the Tunnel Encapsulation
   Attribute ................................................... 21
   4.1. Encapsulation Extended Community ....................... 21
   4.2. Router’s MAC Extended Community ....................... 23
4.3.  Color Extended Community .................................. 23
5.  Semantics and Usage of the Tunnel Encapsulation attribute .. 23
6.  Routing Considerations ......................................... 27
   6.1.  Impact on BGP Decision Process ............................ 27
   6.2.  Looping, Infinite Stacking, Etc. .......................... 27
7.  Recursive Next Hop Resolution ................................. 28
8.  Use of Virtual Network Identifiers and Embedded Labels when
   Imposing a Tunnel Encapsulation ............................... 28
   8.1.  Tunnel Types without a Virtual Network Identifier Field . 29
   8.2.  Tunnel Types with a Virtual Network Identifier Field ... 29
      8.2.1.  Unlabeled Address Families ........................... 30
      8.2.2.  Labeled Address Families .............................. 30
9.  Applicability Restrictions .................................... 31
10. Scoping .......................................................... 32
11. Error Handling .................................................. 32
12. IANA Considerations ............................................. 34
   12.1.  Subsequent Address Family Identifiers .................... 34
   12.2.  BGP Path Attributes ....................................... 34
   12.3.  Extended Communities ..................................... 35
   12.4.  BGP Tunnel Encapsulation Attribute Sub-TLVs ............ 35
   12.5.  Tunnel Types ............................................... 36
   12.6.  Flags Field of Vxlan Encapsulation sub-TLV ............. 36
   12.7.  Flags Field of Vxlan-GPE Encapsulation sub-TLV ....... 36
   12.8.  Flags Field of NVGRE Encapsulation sub-TLV ............ 36
   12.9.  Embedded Label Handling sub-TLV ........................ 36
13. Security Considerations ........................................ 37
14. Acknowledgments ................................................ 38
15. Contributor Addresses .......................................... 38
16. References ....................................................... 38
   16.1.  Normative References ..................................... 38
   16.2.  Informative References ................................... 40
Authors’ Addresses ............................................... 41

1.  Introduction

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

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and
"OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all
capitals, as shown here.
1.1. Brief Summary of RFC 5512

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

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

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

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

1.2. Deficiencies in RFC 5512

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

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

o 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.

o 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.

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

1.3. Brief Summary of Changes from RFC 5512

In this document we address these deficiencies by:

o Deprecating the Encapsulation SAFI.

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

o 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.

o 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.

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

One of the sub-TLVs defined in [RFC5512] is the "Encapsulation sub-TLV". For a given tunnel, the encapsulation sub-TLV specifies some of the information needed to construct the encapsulation header used when sending packets through that tunnel. This document defines encapsulation sub-TLVs for a number of tunnel types not discussed in [RFC5512]: VXLAN (Virtual Extensible Local Area Network, [RFC7348]),
VXLAN-GPE (Generic Protocol Extension for VXLAN, [I-D.ietf-nvo3-vxlan-gpe]), NVGRE (Network Virtualization Using Generic Routing Encapsulation [RFC7637]), and MPLS-in-GRE (MPLS in Generic Routing Encapsulation [RFC2784], [RFC2890], [RFC4023]). MPLS-in-UDP [RFC7510] is also supported, but an Encapsulation sub-TLV for it is not needed.

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

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

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

1.4. Impact on RFC 5566

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

2. The Tunnel Encapsulation Attribute

The Tunnel Encapsulation attribute is an optional transitive BGP Path attribute. IANA has assigned the value 23 as the type code of the attribute. The attribute is composed of a set of Type-Length-Value (TLV) encodings. Each TLV contains information corresponding to a particular tunnel type. A TLV is structured as shown in Figure 1:
o Tunnel Type (2 octets): identifies a type of tunnel. The field contains values from the IANA Registry "BGP Tunnel Encapsulation Attribute Tunnel Types".

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

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

o 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:

```plaintext
+--------------------------------+
| Sub-TLV Type (1 Octet)        |
+--------------------------------+
| Sub-TLV Length (1 or 2 Octets)|
+--------------------------------+
| Sub-TLV Value (Variable)      |
+--------------------------------+
```

Table 1: Tunnel Encapsulation Sub-TLV Format

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

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

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

3. Tunnel Encapsulation Attribute Sub-TLVs

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

3.1. The Tunnel Endpoint Sub-TLV

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

1. a four-octet Autonomous System (AS) number sub-field
2. a two-octet Address Family sub-field
3. an address sub-field, whose length depends upon the Address Family.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Autonomous System Number                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Address Family           |           Address             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Tunnel Endpoint Sub-TLV Value Field

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

If the Address Family subfield contains the value for IPv4, the address subfield must contain an IPv4 address (a /32 IPv4 prefix).
In this case, the length field of Tunnel Endpoint sub-TLV must contain the value 10 (0xa).

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

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

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

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

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

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

- The sub-TLV contains the value for IPv4 in its Address Family subfield, but the length of the sub-TLV’s value field is other than 10 (0xa).
- The sub-TLV contains the value for IPv6 in its Address Family subfield, but the length of the sub-TLV’s value field is other than 22 (0x16).
- The sub-TLV contains the value zero in its Address Family field, but the length of the sub-TLV’s value field is other than 6, or the Autonomous System subfield is not set to zero.
- The IP address in the sub-TLV’s address subfield is not a valid IP address (e.g., it’s an IPv4 broadcast address).
It can be determined that the IP address in the sub-TLV’s address subfield does not belong to the non-zero AS whose number is in the its Autonomous System subfield. (See section Section 13 for discussion of one way to determine this.)

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

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

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

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

3.2. Encapsulation Sub-TLVs for Particular Tunnel Types

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

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

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

3.2.1. VXLAN

This document defines an encapsulation sub-TLV for VXLAN tunnels. When the tunnel type is VXLAN, the following is the structure of the value field in the encapsulation sub-TLV:
V: This bit is set to 1 to indicate that a "valid" VN-ID (Virtual Network Identifier) is present in the encapsulation sub-TLV. Please see Section 8.

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

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

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

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

When forming the VXLAN encapsulation header:

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

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

If the M bit is not set, and the payload being sent through the VXLAN tunnel is an ethernet frame, the Destination MAC Address field of the Inner Ethernet Header is just the Destination MAC Address field of the payload’s ethernet header.
If the M bit is not set, and the payload being sent through the VXLAN tunnel is an IP or MPLS packet, the Inner Destination MAC address field is set to a configured value; if there is no configured value, the VXLAN tunnel cannot be used.

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

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

### 3.2.2. VXLAN-GPE

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

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Ver|V|R|R|R|R|                 Reserved                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       VN-ID                   |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

**Figure 4: VXLAN GPE Encapsulation Sub-TLV**

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

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

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

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

- The values of the V and R bits are NOT copied into the flags field of the VXLAN-GPE header. However, the values of the Ver bits are copied into the VXLAN-GPE header. Other bits in the flags field of the VXLAN-GPE header are set as per [I-D.ietf-nvo3-vxlan-gpe].
- See Section 8 to see how the VNI field of the VXLAN-GPE encapsulation header is set.

### 3.2.3. NVGRE

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

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>V</td>
<td>M</td>
<td>R</td>
</tr>
<tr>
<td></td>
<td>VN-ID (3 Octets)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAC Address (4 Octets)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAC Address (2 Octets)</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAC Address (4 Octets)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAC Address (2 Octets)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAC Address (4 Octets)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAC Address (2 Octets)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAC Address (4 Octets)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAC Address (2 Octets)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAC Address (4 Octets)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAC Address (2 Octets)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAC Address (4 Octets)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAC Address (2 Octets)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAC Address (4 Octets)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAC Address (2 Octets)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 5: NVGRE Encapsulation Sub-TLV**

- **V**: This bit is set to 1 to indicate that a "valid" VN-ID is present in the encapsulation sub-TLV. Please see Section 8.
- **M**: This bit is set to 1 to indicate that a valid MAC Address is present in the encapsulation sub-TLV.
- **R**: The remaining bits in the 8-bit flags field are reserved for further use. They MUST always be set to 0 by the originator of the sub-TLV. Intermediate routers MUST propagate them without modification. Any receiving routers MUST ignore these bits upon a receipt of the sub-TLV.
- **VN-ID**: If the V bit is set, the VN-id field contains a 3 octet VN-ID value. If the V bit is not set, the VN-id field MUST be set to zero.
- **MAC Address**: If the M bit is set, this field contains a 6 octet Ethernet MAC address. If the M bit is not set, this field MUST be set to all zeroes.
When forming the NVGRE encapsulation header:

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

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

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

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

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

### 3.2.4. L2TPv3

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

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Session ID (4 octets)                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                        Cookie (Variable)                      |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

#### Figure 6: L2TPv3 Encapsulation Sub-TLV

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

Cookie: an optional, variable length (encoded in octets -- 0 to 8 octets) value used by L2TPv3 to check the association of a received data message with the session identified by the Session.
ID. Generation and usage of the cookie value is as specified in [RFC3931].

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

3.2.5. GRE

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

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      GRE Key (4 octets)                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: GRE Encapsulation Sub-TLV

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

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

3.2.6. MPLS-in-GRE

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

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       GRE-Key (4 Octets)                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

GRE-Key: 4-octet field [RFC2890] that is generated by the advertising router. The actual method by which the key is obtained is beyond the scope of this document. The key is inserted into the GRE encapsulation header of the payload packets sent by ingress routers to the advertising router. It is intended
to be used for identifying extra context information about the received payload. Note that the key is optional. Unless a key value is being advertised, the MPLS-in-GRE encapsulation sub-TLV MUST NOT be present.

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

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

3.2.7. IP-in-IP

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

3.3. Outer Encapsulation Sub-TLVs

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

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

3.3.1. IPv4 DS Field

Most of the tunnel types that can be specified in the Tunnel Encapsulation attribute require an outer IP encapsulation. The IPv4 Differentiated Services (DS) Field sub-TLV can be carried in the TLV of any such tunnel type. It specifies the setting of the one-octet
Differentiated Services field in the outer IP encapsulation (see [RFC2474]). The value field is always a single octet.

3.3.2. UDP Destination Port

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

3.4. Sub-TLVs for Aiding Tunnel Selection

3.4.1. Protocol Type Sub-TLV

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

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

3.4.2. Color Sub-TLV

The color sub-TLV MAY be encoded as a way to "color" the corresponding tunnel TLV. The value field of the sub-TLV is eight octets long, and consists of a Color Extended Community, as defined in Section 4.3. For the use of this sub-TLV and Extended Community, please see Section 7.
Note that the high-order octet of this sub-TLV’s value field MUST be set to 3, and the next octet MUST be set to 0x0b. (Otherwise the value field is not identical to a Color Extended Community.)

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

3.5. Embedded Label Handling Sub-TLV

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

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

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

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

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

1: The payload will be an MPLS packet with the embedded label at the top of its label stack.
2: The embedded label is not carried in the payload, but is carried either in the virtual network identifier field of the encapsulation header, or else is ignored entirely.

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

3.6. MPLS Label Stack Sub-TLV

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

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

Each label stack entry has the following format:

```
+-------------+-------------+-------------+-------------+
|    Label    |     TC      |       S     |       TTL   |
+-------------+-------------+-------------+-------------+
```

Figure 9: MPLS Label Stack Sub-TLV

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

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

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

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

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

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

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

3.7. Prefix-SID Sub-TLV

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

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

The Prefix-SID sub-TLV can occur in a TLV identifying any type of tunnel. If an Originator SRGB is specified in the sub-TLV, that SRGB MUST be interpreted to be the SRGB used by the tunnel’s egress endpoint. The Label-Index, if present, is the Segment Routing SID that the tunnel’s egress endpoint uses to represent the prefix.
appearing in the NLRI field of the BGP UPDATE to which the Tunnel Encapsulation attribute is attached.

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

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

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

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

4. Extended Communities Related to the Tunnel Encapsulation Attribute

4.1. Encapsulation Extended Community

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

1. it identifies the same tunnel type,
2. it has a Tunnel Endpoint sub-TLV for which one of the following two conditions holds:
   A. its "Address Family" subfield contains zero, or
   B. its "Address" subfield contains the same IP address that appears in the next hop field of the route to which the Tunnel Encapsulation attribute is attached

3. it has no other sub-TLVs.

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

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

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

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

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

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

Note that for tunnel types of the form "X-in-Y", e.g., MPLS-in-GRE, the Encapsulation Extended Community implies that only packets of the specified payload type "X" are to be carried through the tunnel of type "Y".
In the remainder of this specification, when we speak of a route as containing a Tunnel Encapsulation attribute with a TLV identifying a particular tunnel type, we are implicitly including the case where the route contains a Tunnel Encapsulation Extended Community identifying that tunnel type.

4.2. Router’s MAC Extended Community

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

4.3. Color Extended Community

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

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0x03    |     0x0b      |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Color Value                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: Color Extended Community

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

5. Semantics and Usage of the Tunnel Encapsulation attribute

[RFC5512] specifies the use of the Tunnel Encapsulation attribute in BGP UPDATE messages of AFI/SAFI 1/7 and 2/7. That document restricts the use of this attribute to UPDATE 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.
It has been suggested that it may sometimes be useful to attach a Tunnel Encapsulation attribute to a BGP UPDATE message that is also carrying a PMSI (Provider Multicast Service Interface) Tunnel attribute [RFC6514]. If the PMSI Tunnel attribute specifies an IP tunnel, the Tunnel Encapsulation attribute could be used to provide additional information about the IP tunnel. The usage of the Tunnel Encapsulation attribute in combination with the PMSI Tunnel attribute is outside the scope of this document.

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

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

Suppose that:

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

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

If the Tunnel Encapsulation attribute contains several TLVs (i.e., if it specifies several tunnels), router R may choose any one of those
tunnels, based upon local policy. If any tunnel TLV contains one or more Color sub-TLVs (Section 3.4.2) and/or the Protocol Type sub-TLV (Section 3.4.1), the choice of tunnel may be influenced by these sub-TLVs.

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

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

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

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

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

Whenever a new Tunnel Type TLV is defined, the specification of that TLV should describe (or reference) the procedures for creating the
encapsulation header used to forward packets through that tunnel
type. If a tunnel type codepoint is assigned in the IANA "BGP Tunnel
Encapsulation Tunnel Types" registry, but there is no corresponding
specification that defines an Encapsulation sub-TLV for that tunnel
type, the transmitting endpoint of such a tunnel is presumed to know
a priori how to form the encapsulation header for that tunnel type.

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

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

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

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

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

6.1. Impact on BGP Decision Process

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

6.2. Looping, Infinite Stacking, Etc.

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

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

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

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

Suppose that:

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

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

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

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

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

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

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

8.1. Tunnel Types without a Virtual Network Identifier Field

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

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

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

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

8.2. Tunnel Types with a Virtual Network Identifier Field

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

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

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

This sub-section applies when:

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

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

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

8.2.2. Labeled Address Families

This sub-section applies when:

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

8.2.2.1. When a Valid VNI has been Signaled

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

- If the TLV contains an Embedded Label Handling sub-TLV whose value is 1, then the virtual network identifier field of the encapsulation header is set to the value of the virtual network identifier field of the Encapsulation sub-TLV.
The embedded label (from the NLRI of the route that is carrying the Tunnel Encapsulation attribute) appears at the top of the MPLS label stack in the encapsulation payload.

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

8.2.2.2. When a Valid VNI has not been Signaled

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

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

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

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

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

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

9. Applicability Restrictions

In a given UPDATE of a labeled address family, the label embedded in the NLRI is generally a label that is meaningful only to the router whose address appears as the next hop. Certain of the procedures of Section 8.2.2.1 or Section 8.2.2.2 cause the embedded label to be carried by a data packet to the router whose address appears in the Tunnel Endpoint sub-TLV. If the Tunnel Endpoint sub-TLV does not identify the same router that is the next hop, sending the packet through the tunnel may cause the label to be misinterpreted at the tunnel’s egress endpoint. This may cause misdelivery of the packet.
Internet-Draft       Tunnel Encapsulation Attribute            July 2019

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

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

10. Scoping

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

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

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

11. Error Handling

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

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

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

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

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

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

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

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

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

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

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

12. IANA Considerations

12.1. Subsequent Address Family Identifiers

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

12.2. BGP Path Attributes

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

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

12.4. BGP Tunnel Encapsulation Attribute Sub-TLVs

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

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

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

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

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

- 6: Remote Endpoint
- 7: IPv4 DS Field
- 8: UDP Destination Port
- 9: Embedded Label Handling
- 10: MPLS Label Stack
11: Prefix SID

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

12.5. Tunnel Types

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

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

12.6. Flags Field of Vxlan Encapsulation sub-TLV

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

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

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

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

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

12.8. Flags Field of NVGRE Encapsulation sub-TLV

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

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

12.9. Embedded Label Handling sub-TLV

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

13. Security Considerations

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

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

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

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

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

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

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

If BGP Origin Validation is used as specified above, and the tunnel specified in a particular TLV of a Tunnel Encapsulation attribute is therefore regarded as "suspicious", that tunnel should not be used.
Other tunnels specified in (other TLVs of) the Tunnel Encapsulation attribute may still be used.

14. Acknowledgments

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

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

15. Contributor Addresses

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

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

Email: randy@psg.com

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

Email: robert@raszuk.net

16. References

16.1. Normative References

[I-D.ietf-idr-bgp-prefix-sid]


16.2. Informative References

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

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


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


[Page 41]
Abstract

To aid BGP receiver to steer the AS-outgoing traffic among the exit links, this document introduces a new BGP community, congestion status community, to carry the link bandwidth and utilization information, especially for the exit links of one AS. If accepted, this document will update RFC4271, RFC4360 and RFC7153.

The introduced congestion status community is not used to impact the decision process of BGP specified in section 9.1 of RFC4271, but can be used by route policy to impact the data forwarding behavior.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 4, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents.
1. Introduction

Knowing the congestion status (bandwidth and utilization) of the AS exit links is useful for traffic steering, especially for steering the AS outgoing traffic among the exit links. Section 7 of [I-D.gredler-idr-bGPLU-epe] explicitly specifies this kind of requirement, which is also needed in our field network.

The following figure is used to illustrate the benefits of knowing the congestion status of the AS exit links. AS A has multiple exit links connected to AS B. Both AS A and B has exit link to AS C, and AS B provides transit service for AS A. Due to cost or some other reasons, AS A prefers using AS B to transmit its’ traffic to AS C, not the directly connected link between AS A and C. If the exit routers, Router 7 and 8, in AS A tell their iBGP peers the congestion status of the exit links, the peers in turn can steer some outgoing traffic toward the less loaded exit link. If AS A knows the link between AS B and AS C is congested, it can steer some traffic towards AS C from AS B to the directly connected link by applying some route policies.
This document introduces new BGP extensions to deliver the congestion status of the exit link to other BGP speakers. The BGP receiver can then use this community to deploy route policy, thus steer AS outgoing traffic according to the congestion status of the exit links. This mechanism can be used by both iBGP and eBGP.

In this version, we provide three solution alternatives according to the discussion in the face to face meetings and email list. After adoption, one solution will be selected as the final solution based on the working group consensus.

In a network deployed SDN (Software Defined Network) controller, congestion status extended community can be used by the controller to steer the AS outgoing traffic among all the exit links from the perspective of the whole network.

For the network with Route Reflectors (RRs) [RFC4456], RRs by default only advertise the best route for a specific prefix to their clients. Thus RR clients have no opportunity to compare the congestion status among all the exit links. In this situation, to allow RR clients learning all the routes for a specific prefix from all the exit links, RRs are RECOMMENDED to enable add-path functionality [RFC7911].
To emphasize, the introduced new BGP extensions have no impact on the decision process of BGP specified in section 9.1 of [RFC4271], but can be used by route policy to impact the data forwarding behavior.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Previous Work

In [constrained-multiple-path], authors from France Telecom also specified the requirement to know the congestion status of a link.

To aid a router to perform unequal cost load balancing, experts from Cisco introduced Link Bandwidth Extended Community in [link-bandwidth-community] to carry the cost to reach the external BGP neighbor. The cost can be either configured per neighbor or derived from the bandwidth of the link that connects the router to a directly connected external neighbor. This document was accepted by the IDR working group, but expired in 2013.

Link Bandwidth Extended Community only carries the link bandwidth of the exit link. The method provided in our document can carry the link bandwidth together with the link utilization information. What the BGP receiver needs to impact its traffic steering policy is the up-to-date unused link bandwidth, which can be derived from the link bandwidth and link utilization. Since Link Bandwidth Extended Community is expired, the BGP speaker who receives update message with both Link Bandwidth Extended Community and Congestion Status Community SHOULD ignore the Link Bandwidth Extended Community and use the Congestion Status Community.

4. Solution Alternative 1: Extended Community

As described in [RFC4360], the extended community attribute is an 8-octet value with the first one or two octets to indicate the type of this attribute. Since congestion status community needs to be delivered from one AS to other ASes, and used by the BGP speakers both in other ASes and within the same AS as the sender, it MUST be a transitive extended community, i.e. the T bit in the first octet MUST be zero.

We only define the congestion status community for four-octet AS number [RFC6793], since all the BGP speakers can handle four-octet AS number now and the two-octet AS numbers can be mapped to four-octet...
AS numbers by setting the two high-order octets of the four-octet field to zero, as per [RFC6793].

Congestion status community is a sub-type allocated from Transitive Four-Octet AS-Specific Extended Community Sub-Types defined in section 5.2.4 of [RFC7153]. Its format is as Figure 1.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Type =0x02   |    Sub-Type   |        Sender AS Number       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Sender AS Number (cont.)   |    Bandwidth    | Utilization |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Congestion status extended community

Type: 1 octet. This field MUST be 0x02 to indicate this is a Transitive Four-Octet AS-Specific Extended Community.

Sub-Type: 1 octet. It is used to indicate this is a Congestion Status Extended Community. Its value is to be assigned by IANA.

Sender AS Number: 4 octets. Its value is the AS number of the BGP speaker who generates this congestion status extended community. If the generator has 2-octct AS number, it MUST encode its AS number in the last (low order) two bytes and set the first (high order) two bytes to zero, as per [RFC6793].

Bandwidth: 1 octet. Its value is the bandwidth of the exit link in unit of 10 gbps (gigabits per second). The link with bandwidth less than 10 gbps is not suitable to use this feature. To reflect the practice that sometimes the traffic is rate limited to a capacity smaller than the physical link, the value of the bandwidth can be the configured capacity of the link. The available configured capacity can be calculated from this field together with Utilization field. Zero means the bandwidth is unknown or is not advertised to other peers.

Utilization: 1 octet. Its value is the utilization of the exit link in unit of percent. A value bigger than 100 means the incoming traffic is higher than the link capacity. We can use the "Utilization" field together with the "Bandwidth" field to calculate the traffic load that we can further steer to this exit link.
5. Solution Alternative 2: Large Community

As described in [RFC8092], the BGP large community attribute is an optional transitive path attribute of variable length, consisting of 12-octet values. The BGP large community attribute is mainly used to extend the size of BGP Community [RFC1997] and Extended Community [RFC4360], thus to accommodate at least two four-octet ASNs [RFC6793]. As shown in the following figure, the format of the 12-octet BGP Large Community value is not suitable to be used to define new type for congestion status community.

0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Global Administrator                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Local Data Part 1                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Local Data Part 2                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Figure 2

Global Administrator: A four-octet namespace identifier.

Local Data Part 1: A four-octet operator-defined value.

Local Data Part 2: A four-octet operator-defined value.

6. Solution Alternative 3: Community Container

As described in [I-D.ietf-idr-wide-bgp-communities], the BGP Community Container has flexible encoding format, which we can use to define the congestion status community.

A new type of the BGP Community Container is defined for the congestion status community, which has the same common header as the BGP Community Container with the following encoding format.
Type: 2 octets. Its value is to be assigned by IANA from the registry "BGP Community Container Types" to indicate this is the Congestion Status Community.

Flags: 1 octet. C and T bits MUST be set to indicate the Congestion Status Community is transitive across confederation and AS boundaries. The other bits in Flags field MUST be set to zero when originated and SHOULD be ignored upon receipt.

Reserved: Reserved fields are reserved for future definition, which MUST be set to zero when originated and SHOULD be ignored upon receipt.

Length: 2 octets. This field represents the total length of a given container’s contents in octets.

Sender AS Number: 4 octets. Its value is the AS number of the BGP speaker who generates this congestion status community. If the generator has 2-octct AS number, it MUST encode its AS number in the last (low order) two bytes and set the first (high order) two bytes to zero, as per [RFC6793].

Bandwidth: 4 octets. Its value is the bandwidth of the exit link in IEEE floating point format (see [IEEE.754.1985]), expressed in bytes per second. Zero means the bandwidth is unknown or is not advertised to other peers. Appendix A lists some typical bandwidth values, most of which are extracted from Section 3.1.2 of [RFC3471].

To reflect the practice that sometimes the traffic is rate limited to a capacity smaller than the physical link, the value of the bandwidth can be the configured capacity of the link. The available configured capacity can be calculated from this field together with Utilization field.
Utilization: 1 octet. Its value is the utilization of the exit link in unit of percent. A value bigger than 100 means the incoming traffic is higher than the link capacity. We can use the "Utilization" field together with the "Bandwidth" field to calculate the traffic load that we can further steer to this exit link.

7. Deployment Considerations

- To avoid route oscillation
  
The exit router SHOULD set a threshold. When the utilization change reaches the threshold, the exit router SHOULD generate a BGP update message with congestion status community.

  Implementations SHOULD further reduce the BGP update messages triggered by link utilization change using the method similar to BGP Route Flap Damping [RFC2439]. When link utilization change by small amounts that fall under thresholds that would cause the announcement of BGP update message, implementations SHOULD suppress the announcement and set the penalty value accordingly.

  To reduce the update churn introduced, when one BGP router needs to re-advertise a BGP path due to attribute changes, it SHOULD update its Congestion Status Community at the same time. Supposing there are N ASes on the way from the far end egress BGP speaker to the final ingress BGP speaker, this allows reducing the update churn as the final ingress BGP speaker will receive a single UPDATE refreshing the N communities, rather than N UPDATEs, each refreshing one community.

- To avoid traffic oscillation

  Traffic oscillation means more traffic than expected is attracted to the low utilized link, and some traffic has to be steered back to other links.

  Route policy is RECOMMENDED to be set at the exit router. Congestion status community is only conveyed for some specific routes or only for some specific BGP peers.

  Congestion status community can also be used in a SDN network. The SDN controller uses the exit link utilization information to steer the Internet access traffic among all the exit links from the perspective of the whole network.

- Other Concerns
To avoid forwarding loops incremental deployment issues, complications in error handling, the reception of such community over IBGP session SHOULD NOT influence routing decision unless tunneling is used to reach the BGP Next-Hop.

8. Security Considerations

This document defines a new BGP community to carry the congestion status of the exit link. It is up to the BGP receiver to trust the congestion status communities or not. Following deployment models can be considered.

The BGP receiver may choose to only trust the congestion status communities generated by some specific ASes or containing bandwidth greater than a specific value.

You can filter the congestion status communities at the border of your trust/administrative domain. Hence all the ones you receive are trusted.

You can record the communities received over time, monitor the congestion e.g. via probing, detect inconsistency and choose to not trust anymore the ASes which advertise fake news.

9. IANA Considerations

For solution alternative 1, one sub-type is solicited to be assigned from Transitive Four-Octet AS-Specific Extended Community Sub-Types registry to indicate the Congestion Status Community defined in this document.

For solution alternative 3, one community value is solicited to be assigned from the registry "Registered Type 1 BGP Wide Community Community Types" to indicate the Congestion Status Community defined in this document.

10. Acknowledgments

We appreciate the constructive suggestions received from Bruno Decraene. Many thanks to Rudiger Volk, Susan Hares, John Scudder, Randy Bush for their review and comments to improve this document.

11. References
11.1.  Normative References

[I-D.ietf-idr-wide-bgp-communities]
Raszuk, R., Haas, J., Lange, A., Decraene, B., Amante, S.,
and P. Jakma, "BGP Community Container Attribute",
draft-ietf-idr-wide-bgp-communities-04 (work in progress),
March 2017.

[RFC2119]  Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,

Border Gateway Protocol 4 (BGP-4)", RFC 4271,
DOI 10.17487/RFC4271, January 2006,

Communities Attribute", RFC 4360, DOI 10.17487/RFC4360,

[RFC7153]  Rosen, E. and Y. Rekhter, "IANA Registries for BGP
Extended Communities", RFC 7153, DOI 10.17487/RFC7153,

I., and N. Hilliard, "BGP Large Communities Attribute",
RFC 8092, DOI 10.17487/RFC8092, February 2017,

11.2.  Informative References

[constrained-multiple-path]
Boucadair, M. and C. Jacquenet, "Constrained Multiple BGP
draft-boucadair-idr-constrained-multiple-path-00.txt>.

[I-D.gredler-idr-bgplu-epe]
Gredler, H., Vairavakkalai, K., R, C., Rajagopalan, B.,
Aries, E., and L. Fang, "Egress Peer Engineering using
BGP-LU", draft-gredler-idr-bgplu-epe-11 (work in progress),
October 2017.


Appendix A. Bandwidth Values

Some typical bandwidth values encoded in 32-bit IEEE floating point format are enumerated below.
<table>
<thead>
<tr>
<th>Link Type</th>
<th>Bit-rate (Mbps)</th>
<th>Bandwidth Value (Bytes/Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>2.048</td>
<td>0x487A0000</td>
</tr>
<tr>
<td>Ethernet</td>
<td>10.00</td>
<td>0x49989680</td>
</tr>
<tr>
<td>Fast Ethernet</td>
<td>100.00</td>
<td>0x4B3EBC20</td>
</tr>
<tr>
<td>OC-3/STM-1</td>
<td>155.52</td>
<td>0x4B9450C0</td>
</tr>
<tr>
<td>OC-12/STM-4</td>
<td>622.08</td>
<td>0x4C9450C0</td>
</tr>
<tr>
<td>GigE</td>
<td>1000.00</td>
<td>0x4CEE6B28</td>
</tr>
<tr>
<td>OC-48/STM-16</td>
<td>2488.32</td>
<td>0x4D9450C0</td>
</tr>
<tr>
<td>OC-192/STM-64</td>
<td>9953.28</td>
<td>0x4E9450C0</td>
</tr>
<tr>
<td>10GigE</td>
<td>10000.00</td>
<td>0x4E9502F9</td>
</tr>
<tr>
<td>OC-768/STM-256</td>
<td>39813.12</td>
<td>0x4F9450C0</td>
</tr>
<tr>
<td>100GigE</td>
<td>100000.00</td>
<td>0x503A43B7</td>
</tr>
</tbody>
</table>

Authors’ Addresses

Zhenqiang Li
China Mobile
No.32 Xuanwumenxi Ave., Xicheng District
Beijing  100032
P.R. China

Email: li_zhenqiang@hotmail.com

Jie Dong
Huawei Technologies
Huawei Campus, No.156 Beiqing Rd.
Beijing  100095
P.R. China

Email: jie.dong@huawei.com
Abstract

A bit, F bit, is defined in traffic action extended community, which is used by FlowSpec to indicate the associated specifications be populated in FIB (Forwarding Information Base) after appropriate process.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 4, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as
described in the Simplified BSD License.

Table of Contents

1. Introduction ......................................................... 2
2. Requirements Language ............................................. 3
3. Populate to FIB Action ............................................. 3
4. Implementation Considerations ...................................... 3
5. Security Considerations ............................................ 4
6. IANA Considerations .................................................. 4
7. Normative References ................................................ 4
8. Authors’ Addresses ................................................... 5

1. Introduction

BGP FlowSpec [RFC5575] provides a flexible mechanism to distribute
traffic flow specifications, where the matching rules are encoded in
the Border Gateway Protocol Network Layer Reachability Information
(BGP NLRI) with defined new format and the corresponding actions are
encoded in BGP Extended communities.

In routers, traffic flow specifications distributed by BGP FlowSpec
[RFC5575] are stored in distinct set of RIBs (Routing Information
Base) according to their (AFI, SAFI) pairs. These RIBs are then
populated to the dedicated hardware (most of them are TCAM based)
usually shared with ACLs (Access Control Lists). The dedicated
hardware is much more expensive and space limited when compared with
the hardware used to store the FIB (Forwarding Information Base),
which is usually sufficient to fit several millions of FIB entries.
Although in some implementations, the hardware used to populate
traffic flow specifications and FIB entries is the same, the size for
each parts is fixed at design stage. As the number of ACL rules and
FlowSpec specifications increases, especially when FlowSpec is used
for dynamic traffic flow steering, which is one of the three BGP
FlowSpec applications listed in [RFC5575] and
[I-D.ietf-idr-rfc5575bis], hardware space requirement of FlowSpec
specifications in the field network may exceed the size of the
dedicated hardware. To save the limited and expensive space of the
dedicated hardware, it is better to populate some FlowSpec
specifications to FIB if possible. The destination prefix based
FlowSpec specifications, for example, are suitable to be populated to
FIB.

However, there is no method in the current version of BGP FlowSpec
[RFC5575] and RFC5575bis [I-D.ietf-idr-rfc5575bis] to indicate the
associated specifications are suitable to be populated to FIB. This
document defines a new bit, F bit (populate to FIB), in 0x8007 traffic action extended community to satisfy the requirement.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Populate to FIB Action

F bit, populate to FIB bit, is defined in 0x8007 traffic action extended community [RFC5575] to indicate the associated BGP FlowSpec specifications are suitable to be populated to FIB. Thus the space of the dedicated hardware that is used to store the BGP FlowSpec specifications can be saved for other kinds of BGP FlowSpec specifications and ACL rules.

The encoding format of the traffic action extended community with F bit is shown below. The F bit is solicited to be assigned by IANA.

```
|       reserved    | F | S | T |
+---+---+---+---+---+---+---+---+
```

Traffic-action extended community consists of 2 bytes for type and subtype, the value of which MUST be 0x8007, and 6 bytes for value, of which only the 3 least significant bits of the 6th byte (from left to right) are currently defined. S and T are defined in BGP FlowSpec [RFC5575]. F is defined as:

- F: Populate to FIB Action (bit 45, to be asssined by IANA): When this bit is set, the associated BGP FlowSpec specifications SHOULD be populated to FIB. If not set, the associated BGP FlowSpec specifications MUST NOT be populated to FIB. If this bit is set and the associated BGP FlowSpec specifications can not be populated to FIB, the associated BGP FlowSpec specifications MUST be ignored.

4. Implementation Considerations

FlowSpec rules are ordering sensitive. After ordering processing as per section 5.1 of [RFC5575], they are searched sequentially until a matching rule is found. FIB entries, on the contrary, have no ordering implication. Longest prefix matching is the rule to choose the matching FIB entry. Only the destination prefix based, F bit tagged FlowSpec rules that pass the validation (as per section 6 of
and ordering (as per section 5.1 of [RFC5575]) processing are suitable to be populated into FIB. When populating a FlowSpec rule into FIB, the following facts have to be taken into account.

- FlowSpec rules have higher priority than corresponding IGP and BGP routing entries.
- When populating the FIB, the FlowSpec rules with F bit tagged are preferred than the corresponding IGP and BGP routing entries.
- When a FlowSpec rule is being populated into FIB, the FIB entries, including those come from IGP or BGP updates, covered by this FlowSpec rule MUST be removed or replaced by this FlowSpec rule.
- The populated FlowSpec rules in the FIB MUST not be overridden by IGP or BGP updates.

5. Security Considerations

This document defines a new bit in the traffic action extended community to indicate the associated BGP FlowSpec specifications SHOULD be populated to FIB directly. This bit does not introduce any new security issues. The same security considerations as for the BGP FlowSpec [RFC5575] applies.

6. IANA Considerations

One bit, F bit, is solicited to be assigned from Traffic Action Fields registry. This bit is used by BGP FlowSpec to indicate the associated BGP FlowSpec specifications SHOULD be populated to FIB directly.

7. Normative References

[I-D.ietf-idr-rfc5575bis]


Authors' Addresses

Zhenqiang Li
China Mobile
No.32 Xuanwumenxi Ave., Xicheng District
Beijing  100032
P.R. China
Email: li_zhenqiang@hotmail.com

Jie Dong
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing  100095
China
Email: jie.dong@huawei.com

Shunwan Zhuang
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing  100095
China
Email: zhuangshunwan@huawei.com
<BGP Support for Fast Link Status Notification>
draft-sun-idr-bgp-ls-notification-00

Abstract

This document describes the use of Border Gateway Protocol (BGP) community. This optional transitive community will instruct router to monitor itself ports. With this community, controller only needs to send route update message once and will get the feedback only if link status changes. In particular this community can help controller get the link status changing notification much faster than current method.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

Copyright and License Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.
This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1 Introduction .......................................................... 3
  1.1 Large-scale DC Routing Solution ............................... 3
  1.2 BFD protocol and Hellos Protocol ............................ 5
2 Another Centralized Link Detection Method Based on BGP .......... 5
  2.1 Basic Principle .................................................. 5
  2.2 Advantages and Benefits of this solution .................... 7
3 IANA Considerations ............................................... 7
4 References .......................................................... 8
  4.1 Normative References .......................................... 8
  4.2 Informative References ...................................... 8
Authors’ Addresses .................................................. 8
1 Introduction

With the advent of micro services application architecture and the continued advances in massively scaled distributed systems, majority of traffic traversing the data center network is within the data center (east-west). This necessitates the data center network to have deterministic latency (preferably ultra-low), high scalability, high availability and low cost. For those requirements, current large-scale data center network is mostly based on CLOS architecture, [RFC7938] shows a typical 3 layer (5 stages) CLOS architecture (in Figure 1, 3 layer means Leaf-Agg-Spine).

![3-Layer Clos Topology](image)

Note: Leaf is switching node that is connected with servers, Agg is exchange node that aggregates Leaf, and Spine is core exchange node.

Nowadays, the scale of this architecture can support 100k servers. The number of links in network is nearly up to 200k links. Managing the large number of switches and links in a data center from a Controller is a difficult scale problem.

1.1 Large-scale DC Routing Solution
[RFC7938] introduces a link detection solution based on BGP. This RFC uses ebgp to connect switches (physical link) and use ibgp to connect switches and controller (logical link). The ebgp connections are made using the local loopback addresses of the Routers/Switches. Since this solution does not have any IGP in the network to convey the local loopback addresses to form the EBGP connection, the solution uses a centralized controller to initiate the messages to convey loopback address of a Router to its neighbor. It uses a combination of ibgp and ebgp connections and messages to achieve the following as Figure 2.

```
  +----------+
inject Prefix +-----+Controller+----+
for R1 with   |     +----------+    | expect Prefix
  one-hop     |                   | for R1 from R2
community    |                   | R1--------------|R2|
            |                   | R1+--------------+-+ R2|
            |                   |  Prefix for R1   ++
            |                   | Prefix for R1    | Prefix for R1
            |                   | Prefix not relayed |     | Prefix not relayed |
|R3|                    |R1|                    |R3|
```

Figure 2 one kind of link detection method

In Figure 2, the controller periodically updates the packets to the source of the link, determines link status (status of link connecting to routers-switches) according to whether controller receives update message from destination link node. The controller sends route message to switch R1 periodically, which only contains one-hop community attribute. R1 publishes this message to its neighbor R2 through ebgp with no_export attribute in it. R2 sends this message to controller through ibgp instead of sending message to R3 because of no_export attribute. If controller receives route message from R2 within specified time, it is assumed that R1->R2 status is normal. Otherwise, R1->R2 status is down.

But when link detection packets sending frequency is high, the controller load is heavy, i.e. controller processing capacity is not enough, and firewall device does not accept this large flow of traffic. On the other hand, when link detection packets sending frequency is low, the convergence speed of network is slow, that will lead to loop or network interruption and other issues. Network reliability is unacceptable. With single controller multi-threaded
exabgp + virtual router vyatta, experimental test data shows that this solution can only support 1k links and 512 servers in non-block network.

1.2 BFD protocol and Hellos Protocol

Existing mainstream distributed link monitoring methods are Protocol Hellos [RFC 2328] and BFD protocol [RFC 5880].

Protocol Hellos: Since a protocol (ebgp) is initiated over the link, the status of the link could be inferred by receiving periodic hellos (or the lack of hellos). Protocol hellos are generally regarded as a slow link detection mechanism. Increasing the frequency of hellos only creates a scale issues at many points in the network without really providing sub-second link detection.

BFD solution configures BFD session at both ends of the link which need to be detected. Each end sends detection BFD messages and link will report failure if the detection message is not received on time. BFD needs plenty of configurations to different devices and different ports. In VRRP track, 100k servers need to configure 200k links and 200k ends. At the same time, 100k servers use BFD need to configure 200k links and 400k ends which may cause some unexpectable errors with high cost.

2. Another Centralized Link Detection Method Based on BGP

2.1 Basic Principle

Considering current large-scale DCN link detection method, there are many problems of periodical detection method. When the frequency of sending and receiving messages is high, the controller load will be too heavy. The controller processing capacity is not enough and firewall devices cannot accept this large flow of traffic. On the other hand, when the frequency is low, the convergence speed of network will decrease. This may cause network interruption and worse network reliability.

Compared with traditional link detection method, this solution propose an efficient optimization method which can monitor links automatically. This method can reduce lots of manual configuration work, avoid various types of errors and high cost. Furthermore, it also eases the collection of link status notifications for the controller.

In Figure 3, if the controller need to detect link status from R1 to R2, the process is as following.
Step 1:

a) Controller sends route update message A1 to R1 (nonperiodic, just once) then they can establish a peer. In A1, there’s instructions that can enable R1’s port (link) status monitoring function.

b) is the same as a>, only the objective is R2.

c) The A1 message only contains one-hop community attribute and its prefix is used to identify device R1.

Step 2:

When R1 receives route update message A1 from controller, it will add a no_export attribute so it can only publish to egbp neighbor R2. R2 will publish this route message to controller through ibgp instead of its egbp neighbor device R3.

a) R2 finds that message A1 comes from R1 according to the community in A1.

b) Here we need to define a dedicated bit in communities to specify that R2 should start to monitor its link when it receives this indication. Hence, start to monitor all the links from R1 to R2 in this step.

step 3

If it detects ports (links) status has changed in step 2 b), on the
one hand, if the port status switches from normal to fault, R2 will
tell controller a withdraw message through ibgp. On the other hand,
R2 will tell controller a announce message through ibgp.

step 4

When controller receives route A1 update message from R2:

a) Find corresponding link based on received A1 update message
   <prefix, srcIP>. Prefix marks network device R1 and srcIP means
device R2. The <prefix, srcIP> can tell controller this is the link
   from R1 to R2.

b) If the message is route announce type, link status is normal,
   otherwise, the withdraw type means link status is fault.

It is important to notice here that we do not prefer any link
detection mechanism and the BGP implementation on a vendor’s device
is free to activate any link detection mechanism it chooses (some
examples are BFD, either auto-sensing feature etc.).

2.2 Advantages and Benefits of this solution

Generally speaking, we need a dedicated bit of communities that can
notify R2 to start monitoring the link between R1 and R2. It’s quite
simple but there are many advantages of this solution.

1. It needs no extra configuration and can monitor corresponding
   ports (links) automatically. It helps controller know about every
   link status with existing BGP protocols. It can avoid lots of manual
   configuration and unnecessary errors and costs caused by manual
   configuration.

2. It can solve the conflict that network needs fast convergence time
   but controller capacity constraint. Using this solution, network with
   single controller can support 100k servers while other method can
   only support 512 servers.

3. The performance of real-time link failure recovery is better. With
   experiments, link failure report time reduces from 3s to less than
   50ms, link failure recovery time decreases from 1s to less than 50ms.

3 IANA Considerations

The IANA has registered Transitive Extended Community Types in
RFC7153. This registry contains values of the high-order octet (the
"Type" field) of a Transitive Extended Community.
This method only needs one unassigned type value to notify device monitoring corresponding links.(ports).

4 References

4.1 Normative References


4.2 Informative References


Authors’ Addresses

Marcus Sun
HUAWEI TECHNOLOGIES CO.,LTD
12 E. Mozhou Rd.Nanjing,Jiangsu
China
EMail: marcus.sun@huawei.com

Burjiz Pithawala
HUAWEI TECHNOLOGIES CO.,LTD
2330 Central Expressway, Santa Clara, CA 95050
US
EMail: burjiz.pithawala1@huawei.com