

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 19 August 2022

A. Lindem
Cisco Systems
K. Patel
Arrcus, Inc
S. Zandi
LinkedIn
J. Haas
Juniper Networks, Inc
X. Xu
Capitalonline
15 February 2022

BGP Logical Link Discovery Protocol (LLDP) Peer Discovery
draft-acee-idr-lldp-peer-discovery-11

Abstract

Link Layer Discovery Protocol (LLDP) or IEEE Std 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 19 August 2022.

Copyright Notice

Copyright (c) 2022 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
1.1. Requirements Notation	3
1.1.1. Requirements Language	3
2. LLDP Extensions	3
2.1. LLDP IETF Organizationally Specific TLV Format	3
2.2. BGP Config OS-TLV Format	4
2.2.1. BGP Config OS-TLV - Peering Address Sub-TLV	4
2.2.2. BGP Config OS-TLV - BGP Local AS Sub-TLV	5
2.2.3. BGP Config OS-TLV - BGP Identifier Sub-TLV	6
2.2.4. BGP Config OS-TLV - Session Group-ID Sub-TLV	7
2.2.5. BGP Config OS-TLV - BGP Session Capabilities Sub-TLV	7
2.2.6. BGP Config OS-TLV - Key Chain Sub-TLV	8
2.2.7. BGP Config OS-TLV - Local Address Sub-TLV	9
2.2.8. BGP Config OS-TLV - BGP State Version Sub-TLV	10
3. BGP LLDP Peer Discovery Operations	11
3.1. Advertising BGP Speaker	11
3.2. Receiving BGP Speaker	12
3.3. Updating or Deleting Auto-Discovery Parameters	13
4. LLDP Authentication/Encryption	13
5. Security Considerations	14
6. IANA Considerations	14
6.1. IANA Assigned LLDP Subtype	14
6.2. BGP Config LLDP OS-TLV Sub-TLVs	15
7. Contributors	16
8. References	16
8.1. Normative References	16
8.2. Informative References	16
Appendix A. Acknowledgments	17
Authors' Addresses	17

1. Introduction

Link Layer Discovery Protocol (LLDP) [LLDP] or IEEE Std 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 IETF Organizationally Specific TLV Format

The format of the LLDP IETF 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	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	IETF Organizationally Specific TLV type value, 127.
------	---

Length The length of the remainder of the TLV.

OUI	IETF 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

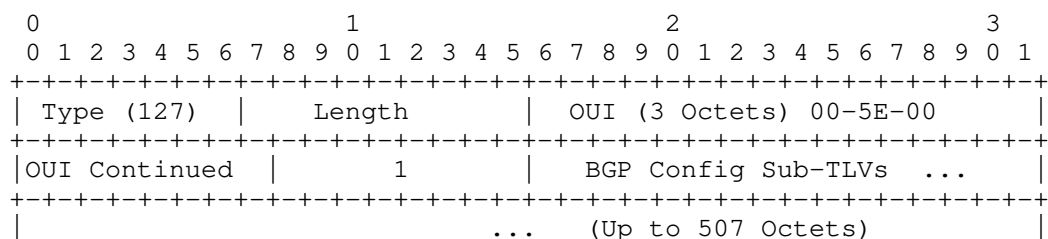
Value	Value for organizationally specific TLV. The Length of the value is 4 octets less than the TLV length.
-------	--

Figure 1: LLDP IETF 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 IETF Organizationally Specific TLVs.

2.2. BGP Config OS-TLV Format

The BGP Config IETF 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.



Length The length of the BGP TLV.

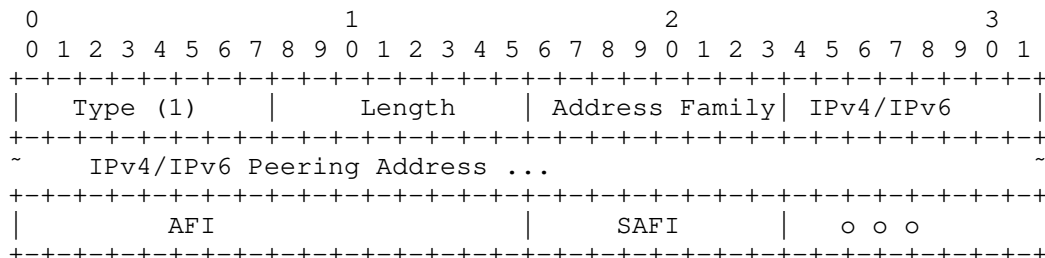
Subtype IETF specific subtype for BGP Config OS-TLV. The value shall be 1.

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.



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.

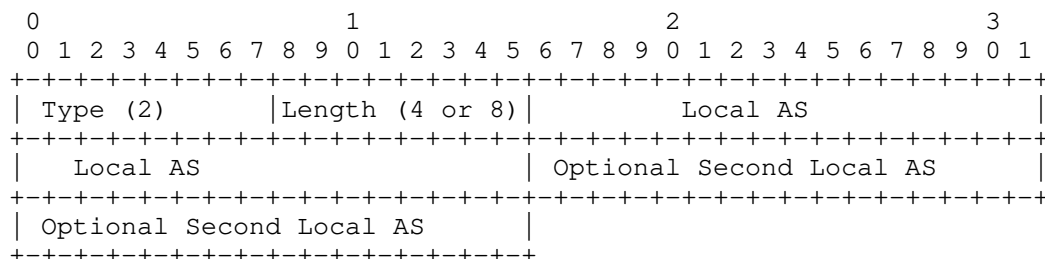
Address Family IANA Address family (1 for IPv4 or 2 for IPv6)

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.



Type The Sub-TLV Type value shall be 2.

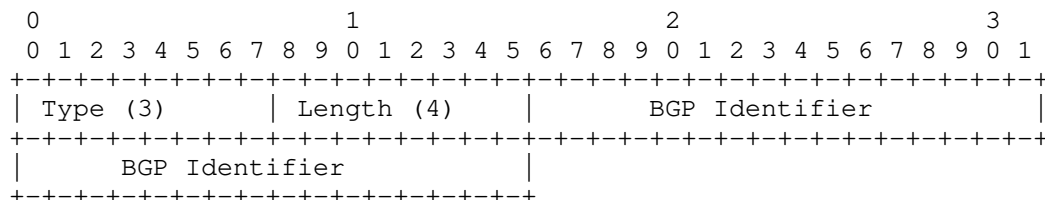
Length The Sub-TLV Length will be 4 or 8 octets.

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.



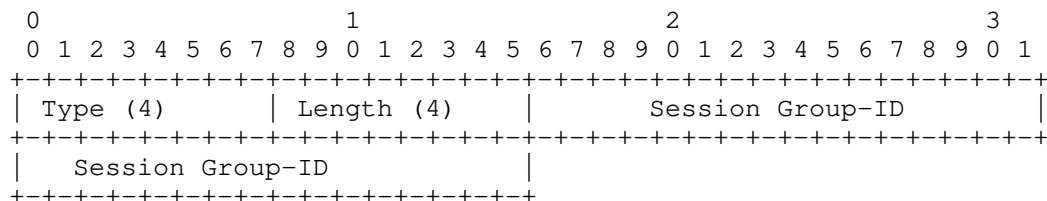
Type The Sub-TLV Type value shall be 3.

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.



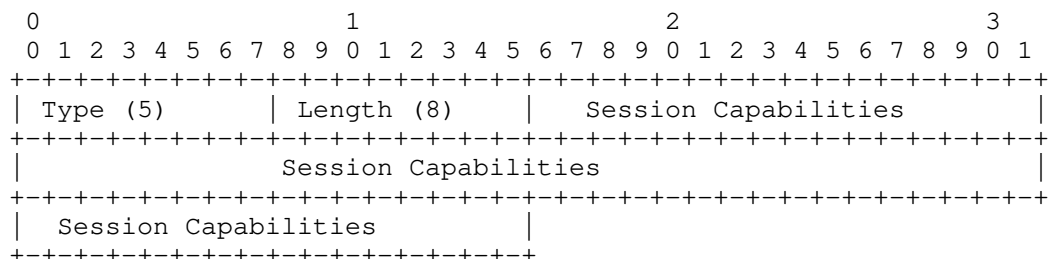
Type The Sub-TLV Type value shall be 4.

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.



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].

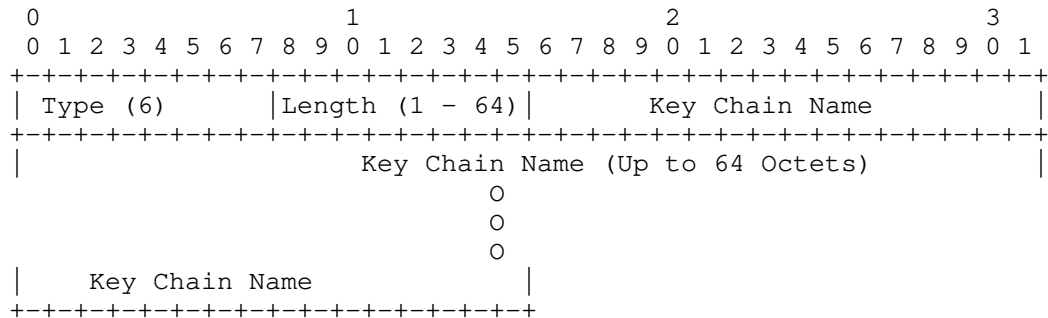
Bit 2: This bit indicates support for TCP-AO authentication [TCP-AO].

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 a 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.



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.

2.2.7. BGP Config OS-TLV - Local Address Sub-TLV

The BGP OS-TLV Local Address Sub-TLV will be used to advertise a local IP addresses used for BGP next-hops. Advertising a local interface address is useful when the address family is different from the advertised BGP peering address.

The format of the BGP Local Interface 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 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|  Type (7)   |   Length   | Address Family | IPv4/IPv6   |
+-----+-----+-----+-----+-----+-----+-----+-----+
~   IPv4/IPv6 Local Address ...                                     ~
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Type The Sub-TLV Type value shall be 7.

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.

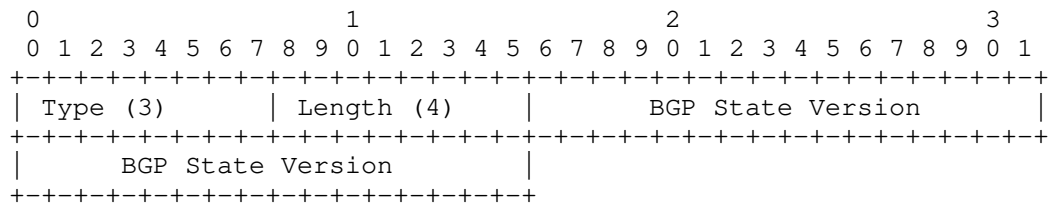
Address Family IANA Address family (1 for IPv4 or 2 for IPv6)

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

2.2.8. BGP Config OS-TLV - BGP State Version Sub-TLV

The BGP OS-TLV Version Sub-TLV will be used to advertise a monotonically increasing version. This version will indicate if any local BGP state that may impact BGP session establishment has changed. Changes can range from anything as obvious a change in local peering address to more indirect changes such as the modification of the key-chain being advertised.

The format of the BGP State Version Sub-TLV is shown below.



Type The Sub-TLV Type value shall be 8.

Length The Sub-TLV Length will be 4 octets.

BGP State Version BGP State Version - Monotonically increasing version number indicating if any local state that may effect BGP session establishment has changed.

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].

The BGP State Version associated with the LLDP peer SHOULD be retained to determine whether anything impacting BGP session establishment has changed. When session establishment fails, this can be used to avoid back-off on attempting to establish a BGP session when nothing has changed on the peer or locally.

3.3. Updating or Deleting Auto-Discovery Parameters

A BGP speaker MAY change or delete any BGP LLDP auto-discovery parameter by simply updating or removing the corresponding Sub-TLV previously advertised in the BGP-OS TLV. Additionally, the BGP State Version Sub-TLV should be advertised with the version incremented from the previous version. The BGP speaker(s) receiving the advertisement will update or delete the changed or deleted auto-discovery parameters. However, there will be no change to existing BGP sessions with the advertising BGP Speaker. Changes to existing BGP sessions are the purview of the BGP protocol and are beyond the scope of this document.

Since LLDP information is cumulative, reception of an LLDP PDU without the BGP-OS TLV indicates that BGP LLDP auto-discovery has been disabled for the BGP speaker and all parameters learnt during BGP LLDP auto-discovery SHOULD be deleted. As above, changes to existing BGP sessions are beyond the scope of this document.

4. LLDP Authentication/Encryption

The IEEE 802.1AE [MACsec] standard can be used for encryption and/or authentication to provide privacy and integrity. MACsec utilizes the Galois/Counter Mode Advanced Encryption Standard (AES-GCM) for authenticated encryption and Galois Message Authentication Code (GMAC) if only authentication, but not encryption is required.

The MACsec Key Agreement (MKA) is included as part of the IEEE 802.1X-20200 Port-Based Network Access Control Standard [MKA]. The purpose of MKA is to provide a method for discovering MACsec peers and negotiating the security keys needed to secure the link.

5. 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].

LLDP Authentication and/or encryption can provided as described in section Section 4.

6. IANA Considerations

6.1. IANA Assigned LLDP Subtype

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

Range	Assignment Policy
0	Reserved (not to be assigned)
1	BGP Configuration
2-127	Unassigned (IETF Review)
128-254	Reserved (Not to be assigned now)
255	Reserved (not to be assigned)

Figure 2: IANA LLDP IETF 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.

6.2. 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.

Range	Assignment Policy
0	Reserved (not to be assigned)
1	Peering Address
2	Local AS
3	BGP Identifier
4	Session Group-ID
5	Session Capabilities
6	Key Chain Name
7	Local Address
8	BGP State Version
9-127	Unassigned (IETF Review)
128-254	Reserved (Not to be assigned now)
255	Reserved (not to be assigned)

Figure 3: LLDP BGP Config OS-TLV Types

- * Types in the range 9-127 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.

7. Contributors

Contributors' Addresses

8. References

8.1. Normative References

- [LLDP] IEEE, "IEEE Standard for Local and metropolitan area networks-- Station and Media Access Control Connectivity Discovery Corrigendum 2: Technical and Editorial Corrections", IEEE 802.1AB-2009/Cor 2-2015, DOI 10.1109/ieeestd.2015.7056401, 9 March 2015, <<https://doi.org/10.1109/ieeestd.2015.7056401>>.
- [MACsec] IEEE, "IEEE Standard for Local and metropolitan area networks - Media Access Control (MAC) Security", IEEE Standard 802.1AE-2018, 27 September 2018.
- [MKA] IEEE, "IEEE Standard for Local and metropolitan area networks - Port Based Network Access Control", IEEE Standard 802.1X-2020, 30 January 2020.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

8.2. Informative References

- [RFC2385] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, DOI 10.17487/RFC2385, August 1998, <<https://www.rfc-editor.org/info/rfc2385>>.

- [RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <<https://www.rfc-editor.org/info/rfc4760>>.
- [RFC5082] Gill, V., Heasley, J., Meyer, D., Savola, P., Ed., and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, DOI 10.17487/RFC5082, October 2007, <<https://www.rfc-editor.org/info/rfc5082>>.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <<https://www.rfc-editor.org/info/rfc5226>>.
- [RFC5492] Scudder, J. and R. Chandra, "Capabilities Advertisement with BGP-4", RFC 5492, DOI 10.17487/RFC5492, February 2009, <<https://www.rfc-editor.org/info/rfc5492>>.
- [RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <<https://www.rfc-editor.org/info/rfc5925>>.
- [RFC7042] Eastlake 3rd, D. and J. Abley, "IANA Considerations and IETF Protocol and Documentation Usage for IEEE 802 Parameters", BCP 141, RFC 7042, DOI 10.17487/RFC7042, October 2013, <<https://www.rfc-editor.org/info/rfc7042>>.
- [RFC7938] Lapukhov, P., Premji, A., and J. Mitchell, Ed., "Use of BGP for Routing in Large-Scale Data Centers", RFC 7938, DOI 10.17487/RFC7938, August 2016, <<https://www.rfc-editor.org/info/rfc7938>>.
- [RFC8177] Lindem, A., Ed., Qu, Y., Yeung, D., Chen, I., and J. Zhang, "YANG Data Model for Key Chains", RFC 8177, DOI 10.17487/RFC8177, June 2017, <<https://www.rfc-editor.org/info/rfc8177>>.

Appendix A. Acknowledgments

Thanks to Sujay Gupta and Paul Congdon 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
United States of America

Email: acee@cisco.com

Keyur Patel
Arrcus, Inc

Email: keyur@arrcus.com

Shawn Zandi
LinkedIn
222 2nd Street
San Francisco, CA 94105
United States of America

Email: szandi@linkedin.com

Jeff Haas
Juniper Networks, Inc
1133 Innovation, Inc.
Sunnyvale, CA 94089
United States of America

Email: jhaas@juniper.net

Xiaohu Xu
Capitalonline

Email: xiaohu.xu@capitalonline.net

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 3 October 2022

A. Azimov
Qrator Labs & Yandex
E. Bogomazov
Qrator Labs
R. Bush
Internet Initiative Japan & Arrcus, Inc.
K. Patel
Arrcus
K. Sriram
USA NIST
1 April 2022

Route Leak Prevention and Detection using Roles in UPDATE and OPEN
Messages
draft-ietf-idr-bgp-open-policy-24

Abstract

Route leaks are the propagation of BGP prefixes that violate assumptions of BGP topology relationships, e.g., announcing a route learned from one transit provider to another transit provider or a lateral (i.e., non-transit) peer or announcing a route learned from one lateral peer to another lateral peer or a transit provider. These are usually the result of misconfigured or absent BGP route filtering or lack of coordination between autonomous systems (ASes). Existing approaches to leak prevention rely on marking routes by operator configuration, with no check that the configuration corresponds to that of the eBGP neighbor, or enforcement that the two eBGP speakers agree on the peering relationship. This document enhances the BGP OPEN message to establish an agreement of the peering relationship on each eBGP session between autonomous systems in order to enforce appropriate configuration on both sides. Propagated routes are then marked 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 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 3 October 2022.

Copyright Notice

Copyright (c) 2022 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. Terminology	3
2.1. Peering Relationships	4
3. BGP Role	5
3.1. BGP Role Capability	5
3.2. Role Correctness	6
4. BGP Only to Customer (OTC) Attribute	8
5. Additional Considerations	10
6. IANA Considerations	10
7. Security Considerations	11
8. References	12
8.1. Normative References	12
8.2. Informative References	13
Acknowledgments	14
Contributors	14
Authors' Addresses	14

1. Introduction

Route leaks are the propagation of BGP prefixes that violate assumptions of BGP topology relationships, e.g., announcing a route learned from one transit provider to another transit provider or a lateral (i.e., non-transit) peer or announcing a route learned from one lateral peer to another lateral peer or a transit provider [RFC7908]. These are usually the result of misconfigured or absent BGP route filtering or lack of coordination between autonomous systems (ASes).

Existing approaches to leak prevention rely on marking routes by operator configuration, with no check that the configuration corresponds to that of the eBGP neighbor, or enforcement that the two eBGP speakers agree on the relationship. This document enhances the BGP OPEN message to establish an agreement of the relationship on each eBGP session between autonomous systems in order to enforce appropriate configuration on both sides. Propagated routes are then marked according to the agreed relationship, allowing both prevention and detection of route leaks.

This document specifies a means of replacing the operator-driven configuration-based method of route leak prevention, described above, with an in-band method for route leak prevention and detection.

This method uses a new configuration parameter, BGP Role, which is negotiated using a BGP Role Capability in the OPEN message [RFC5492]. An eBGP speaker may require the use of this capability and confirmation of BGP Role with a neighbor for the BGP OPEN to succeed.

An optional, transitive BGP Path Attribute, called Only to Customer (OTC), is specified in Section 4. It prevents ASes from creating leaks and detects leaks created by the ASes in the middle of an AS path. The main focus/applicability is the Internet (IPv4 and IPv6 unicast route advertisements).

2. Terminology

The terms "local AS" and "remote AS" are used to refer to the two ends of an eBGP session. The "local AS" is the AS where the protocol action being described is to be performed, and "remote AS" is the AS at the other end of the eBGP session in consideration.

The use of the term "route is ineligible" in this document has the same meaning as in [RFC4271], i.e., "route is ineligible to be installed in Loc-RIB and will be excluded from the next phase of route selection."

2.1. Peering Relationships

The terms for peering relationships defined and used in this document (see below) do not necessarily represent business relationships based on payment agreements. These terms are used to represent restrictions on BGP route propagation, sometimes known as the Gao-Rexford model [Gao]. The terms Provider, Customer, and Peer used here are synonymous to the terms "transit provider", "customer", and "lateral (i.e., non-transit) peer", respectively, used in [RFC7908].

The following is a list of BGP Roles for eBGP peering and the corresponding rules for route propagation:

Provider: MAY propagate any available route to a Customer.

Customer: MAY propagate any route learned from a Customer, or locally originated, to a Provider. All other routes MUST NOT be propagated.

Route Server (RS): MAY propagate any available route to a Route Server Client (RS-Client).

Route Server Client (RS-Client): MAY propagate any route learned from a Customer, or locally originated, to an RS. All other routes MUST NOT be propagated.

Peer: MAY propagate any route learned from a Customer, or locally originated, to a Peer. All other routes MUST NOT be propagated.

If the local AS has one of the above Roles (in the order shown), then the corresponding peering relationship with the remote AS is Provider-to-Customer, Customer-to-Provider, RS-to-RS-Client, RS-Client-to-RS, or Peer-to-Peer (i.e., lateral peers), respectively. These are called normal peering relationships.

If the local AS has more than one peering role with the remote AS such peering relation is called Complex. An example is when the peering relationship is Provider-to-Customer for some prefixes while it is Peer-to-Peer for other prefixes [Gao].

A 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 route propagation rules listed above may result in route leaks [RFC7908]. Automatic enforcement of these rules should significantly reduce route leaks that may otherwise occur due to manual configuration mistakes.

As specified in Section 4, the Only to Customer (OTC) Attribute is used to identify all the routes in the AS that have been received from a Peer, Provider, or RS.

3. BGP Role

The BGP Role characterizes the relationship between the eBGP speakers forming a session. One of the Roles described below SHOULD be configured at the local AS for each eBGP session (see definitions in Section 2) based on the local AS's knowledge of its Role. The only exception is when the eBGP connection is Complex (see Section 5). BGP Roles are mutually confirmed using the BGP Role Capability (described in Section 3.1) on each eBGP session.

Allowed Roles for eBGP sessions are:

- * Provider - the local AS is a transit Provider of the remote AS;
- * Customer - the local AS is a transit Customer of the remote AS;
- * RS - the local AS is a Route Server (usually at an Internet exchange point) and the remote AS is its RS-Client;
- * RS-Client - the local AS is a client of an RS and the RS is the remote AS;
- * Peer - the local and remote ASes are Peers (i.e., have a lateral peering relationship).

3.1. BGP Role Capability

The BGP Role Capability is defined as follows:

- * Code - 9
- * Length - 1 (octet)
- * Value - integer corresponding to speaker's BGP Role (see Table 1).

Value	Role name (for the local AS)
0	Provider
1	RS
2	RS-Client
3	Customer
4	Peer (i.e., Lateral Peer)
5-255	Unassigned

Table 1: Predefined BGP Role Values

If BGP Role is locally configured, the eBGP speaker MUST advertise BGP Role Capability in the BGP OPEN message. An eBGP speaker MUST NOT advertise multiple versions of the BGP Role Capability. The error handling when multiple BGP Role Capabilities are received is described in Section 3.2.

3.2. Role Correctness

Section 3.1 described how BGP Role encodes the relationship on each eBGP session between autonomous systems (ASes).

The mere receipt of BGP Role Capability does not automatically guarantee the Role agreement between two eBGP neighbors. If the BGP Role Capability is advertised, and one is also received from the peer, the Roles MUST correspond to the relationships in Table 2. If the Roles do not correspond, the BGP speaker MUST reject the connection using the Role Mismatch Notification (code 2, subcode TBD).

Local AS Role	Remote AS Role
Provider	Customer
Customer	Provider
RS	RS-Client
RS-Client	RS
Peer	Peer

Table 2: Allowed Pairs of Role Capabilities

For backward compatibility, if the BGP Role Capability is sent but one is not received, the BGP Speaker SHOULD ignore the absence of the BGP Role Capability and proceed with session establishment. The locally configured BGP Role is used for the procedures described in Section 4.

An operator may choose to apply a "strict mode" in which the receipt of a BGP Role Capability from the remote AS is required. When operating in the "strict mode", if the BGP Role Capability is sent, but one is not received, then the connection is rejected using the Role Mismatch Notification (code 2, subcode TBD). See comments in Section 7.

If an eBGP speaker receives multiple but identical BGP Role Capabilities with the same value in each, then the speaker considers them to be a single BGP Role Capability and proceeds [RFC5492]. If multiple BGP Role Capabilities are received and not all of them have the same value, then the BGP speaker MUST reject the connection using the Role Mismatch Notification (code 2, subcode TBD).

The BGP Role value for the local AS (in conjunction with the OTC Attribute in the received UPDATE message) is used in the route leak prevention and detection procedures described in Section 4.

4. BGP Only to Customer (OTC) Attribute

The Only to Customer (OTC) Attribute is an optional transitive path attribute of the UPDATE message with Attribute Type Code 35 and a length of 4 octets. The purpose of this attribute is to enforce that once a route is sent to a Customer, Peer, or RS-Client (see definitions in Section 2.1), it will subsequently go only to Customers. The attribute value is an AS number (ASN) determined by the procedures described below.

The following ingress procedure applies to the processing of the OTC Attribute on route receipt:

1. If a route with the OTC Attribute is received from a Customer or RS-Client, then it is a route leak and MUST be considered ineligible (see Section 2).
2. If a route with the OTC Attribute is received from a Peer (i.e., remote AS with a Peer Role) and the Attribute has a value that is not equal to the remote (i.e., Peer's) AS number, then it is a route leak and MUST be considered ineligible.
3. If a route is received from a Provider, Peer, or RS, and the OTC Attribute is not present, then it MUST be added with a value equal to the AS number of the remote AS.

The following egress procedure applies to the processing of the OTC Attribute on route advertisement:

1. If a route is to be advertised to a Customer, Peer, or RS-Client (when the sender is an RS), and the OTC Attribute is not present, then when advertising the route, an OTC Attribute MUST be added with a value equal to the AS number of the local AS.
2. If a route already contains the OTC Attribute, it MUST NOT be propagated to Providers, Peers, or RS(s).

The above-described procedures provide both leak prevention for the local AS and leak detection and mitigation multiple hops away. In the case of prevention at the local AS, the presence of an OTC Attribute indicates to the egress router that the route was learned from a Peer, Provider, or RS, and it can be advertised only to the customers. The same OTC Attribute which is set locally also provides a way to detect route leaks by an AS multiple hops away if a route is received from a Customer, Peer, or RS-Client. For example, if an AS sets the OTC Attribute on a route sent to a Peer and the route is subsequently received by a compliant AS from a Customer, then the receiving AS detects (based on the presence of the OTC Attribute) that the route is a leak.

The OTC Attribute might be set at the egress of the remote AS or at the ingress of the local AS, i.e., if the remote AS is non-compliant with this specification, then the local AS will have to set the OTC Attribute if it is absent. In both scenarios, the OTC value will be the same. This makes the scheme more robust and benefits early adopters.

The OTC Attribute is considered malformed if the length value is not 4. An UPDATE message with a malformed OTC Attribute SHALL be handled using the approach of "treat-as-withdraw" [RFC7606].

The BGP Role negotiation and OTC Attribute based procedures specified in this document are NOT RECOMMENDED to be used between autonomous systems in an AS Confederation [RFC5065]. If an OTC Attribute is added on egress from the AS Confederation, its value MUST equal the AS Confederation Identifier. Also, on egress from the AS Confederation, an UPDATE MUST NOT contain an OTC Attribute with a value corresponding to any Member-AS Number other than the AS Confederation Identifier.

The procedures specified in this document in scenarios that use private AS numbers behind an Internet-facing ASN (e.g., a data center network [RFC7938] or stub customer) may be used, but any details are outside the scope of this document. On egress from the Internet-facing AS, the OTC Attribute MUST NOT contain a value other than the Internet-facing ASN.

Once the OTC Attribute has been set, it MUST be preserved unchanged (this also applies to an AS Confederation).

The described ingress and egress procedures are applicable only for the address families AFI 1 (IPv4) and AFI 2 (IPv6) with SAFI 1 (unicast) in both cases and MUST NOT be applied to other address families by default. The operator MUST NOT have the ability to modify the procedures defined in this section.

5. Additional Considerations

Roles MUST NOT be configured on an eBGP session with a Complex peering relationship. If multiple eBGP sessions can segregate the Complex peering relationship into eBGP sessions with normal peering relationships, BGP Roles SHOULD be used on each of the resulting eBGP sessions.

An operator may want to achieve an equivalent outcome by configuring policies on a per-prefix basis to follow the definitions of peering relations as described in Section 2.1. However, in this case, there are no in-band measures to check the correctness of the per-prefix peering configuration.

The incorrect setting of BGP Roles and/or OTC Attributes may affect prefix propagation. Further, this document does not specify any special handling of an incorrect AS number in the OTC Attribute.

In AS migration scenarios [RFC7705], a given router may represent itself as any one of several different ASes. This should not be a problem since the egress procedures in Section 4 specify that the OTC Attribute is to be attached as part of route transmission. Therefore, a router is expected to set the OTC value equal to the ASN it is currently representing itself as.

Section 6 of [RFC7606] documents possible negative impacts of "treat-as-withdraw" behavior. Such negative impacts may include forwarding loops or blackholes. It also discusses debugging considerations related to this behavior.

6. IANA Considerations

IANA has registered a new BGP Capability (Section 3.1) in the "Capability Codes" registry's "IETF Review" range [RFC5492]. The description for the new capability is "BGP Role". IANA has assigned the value 9 [to be removed upon publication: <https://www.iana.org/assignments/capability-codes/capability-codes.xhtml>]. This document is the reference for the new capability.

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" in the Capability Codes registry. Assignments consist of a Value and a corresponding Role name. Initially, this registry is to be populated with the data contained in Table 1 found in Section 3.1. Future assignments may be made by the "IETF Review" policy as defined in [RFC8126]. The registry is as shown in Table 3.

Value	Role name (for the local AS)	Reference
0	Provider	This document
1	RS	This document
2	RS-Client	This document
3	Customer	This document
4	Peer (i.e., Lateral Peer)	This document
5-255	To be assigned by IETF Review	

Table 3: IANA Registry for BGP Role

IANA has registered a new OPEN Message Error subcode named the "Role Mismatch" (see Section 3.2) in the OPEN Message Error subcodes registry. IANA has assigned the value 11 [to be removed upon publication: <https://www.iana.org/assignments/bgp-parameters/bgp-parameters.xhtml#bgp-parameters-6>]. This document is the reference for the new subcode.

Due to improper use of the values 8, 9, and 10 in the OPEN Message Error subcodes registry, this document requested IANA to mark these values as "Deprecated". IANA has marked values 8-10 as "Deprecated" in the OPEN Message Error subcodes registry. This document is listed as the reference.

IANA has also registered a new path attribute named "Only to Customer (OTC)" (see Section 4) in the "BGP Path Attributes" registry. IANA has assigned code value 35 [To be removed upon publication: <http://www.iana.org/assignments/bgp-parameters/bgp-parameters.xhtml#bgp-parameters-2>]. This document is the reference for the new attribute.

7. Security Considerations

The security considerations of BGP (as specified in [RFC4271] and [RFC4272]) apply.

This document proposes a mechanism using BGP Role for the prevention and detection of route leaks that are the result of BGP policy misconfiguration. A misconfiguration of the BGP Role may affect prefix propagation. For example, if a downstream (i.e., towards a Customer) peering link were misconfigured with a Provider or Peer

Role, this will limit the number of prefixes that can be advertised in this direction. On the other hand, if an upstream provider were misconfigured (by a local AS) with the Customer Role, this may result in propagating routes that are received from other Providers or Peers. But the BGP Role negotiation and the resulting confirmation of Roles make such misconfigurations unlikely.

Setting the strict mode of operation for BGP Role negotiation as the default may result in a situation where the eBGP session will not come up after a software update. Implementations with such default behavior are strongly discouraged.

Removing the OTC Attribute or changing its value can limit the opportunity for route leak detection. Such activity can be done on purpose as part of an on-path attack. For example, an AS can remove the OTC Attribute on a received route and then leak the route to its transit provider. This kind of threat is not new in BGP and it may affect any Attribute (Note: BGPsec [RFC8205] offers protection only for the AS_PATH Attribute).

Adding an OTC Attribute when the route is advertised from Customer to Provider will limit the propagation of the route. Such a route may be considered as ineligible by the immediate Provider or its Peers or upper layer Providers. This kind of OTC Attribute addition is unlikely to happen on the Provider side because it will limit the traffic volume towards its Customer. On the Customer side, adding an OTC Attribute for traffic engineering purposes is also discouraged because it will limit route propagation in an unpredictable way.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC5065] Traina, P., McPherson, D., and J. Scudder, "Autonomous System Confederations for BGP", RFC 5065, DOI 10.17487/RFC5065, August 2007, <<https://www.rfc-editor.org/info/rfc5065>>.

- [RFC5492] Scudder, J. and R. Chandra, "Capabilities Advertisement with BGP-4", RFC 5492, DOI 10.17487/RFC5492, February 2009, <<https://www.rfc-editor.org/info/rfc5492>>.
- [RFC7606] Chen, E., Ed., Scudder, J., Ed., Mohapatra, P., and K. Patel, "Revised Error Handling for BGP UPDATE Messages", RFC 7606, DOI 10.17487/RFC7606, August 2015, <<https://www.rfc-editor.org/info/rfc7606>>.
- [RFC7908] Sriram, K., Montgomery, D., McPherson, D., Osterweil, E., and B. Dickson, "Problem Definition and Classification of BGP Route Leaks", RFC 7908, DOI 10.17487/RFC7908, June 2016, <<https://www.rfc-editor.org/info/rfc7908>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

8.2. Informative References

- [Gao] Gao, L. and J. Rexford, "Stable Internet routing without global coordination", IEEE/ACM Transactions on Networking, Volume 9, Issue 6, pp 689-692, DOI 10.1109/90.974523, December 2001, <<https://ieeexplore.ieee.org/document/974523>>.
- [RFC4272] Murphy, S., "BGP Security Vulnerabilities Analysis", RFC 4272, DOI 10.17487/RFC4272, January 2006, <<https://www.rfc-editor.org/info/rfc4272>>.
- [RFC7705] George, W. and S. Amante, "Autonomous System Migration Mechanisms and Their Effects on the BGP AS_PATH Attribute", RFC 7705, DOI 10.17487/RFC7705, November 2015, <<https://www.rfc-editor.org/info/rfc7705>>.
- [RFC7938] Lapukhov, P., Premji, A., and J. Mitchell, Ed., "Use of BGP for Routing in Large-Scale Data Centers", RFC 7938, DOI 10.17487/RFC7938, August 2016, <<https://www.rfc-editor.org/info/rfc7938>>.
- [RFC8205] Lepinski, M., Ed. and K. Sriram, Ed., "BGPsec Protocol Specification", RFC 8205, DOI 10.17487/RFC8205, September 2017, <<https://www.rfc-editor.org/info/rfc8205>>.

Acknowledgments

The authors wish to thank Alvaro Retana, Bruno Decraene, Jeff Haas, John Scudder, Sue Hares, Ben Maddison, Andrei Robachevsky, Daniel Ginsburg, Ruediger Volk, Pavel Lunin, Gyan Mishra, and Ignas Bagdonas for review, comments, and suggestions during the course of this work. Thanks are also due to many IESG reviewers whose comments greatly helped improve the clarity, accuracy, and presentation in the document.

Contributors

Brian Dickson
Independent
Email: brian.peter.dickson@gmail.com

Doug Montgomery
USA National Institute of Standards and Technology
Email: dougm@nist.gov

Authors' Addresses

Alexander Azimov
Qrator Labs & Yandex
Ulitsa Iva Tolstogo 16
Moscow
119021
Russian Federation
Email: a.e.azimov@gmail.com

Eugene Bogomazov
Qrator Labs
1-y Magistralnyy tupik 5A
Moscow
123290
Russian Federation
Email: eb@qrator.net

Randy Bush
Internet Initiative Japan & Arrcus, Inc.
5147 Crystal Springs
Bainbridge Island, Washington 98110
United States of America
Email: randy@psg.com

Keyur Patel
Arrcus
2077 Gateway Place, Suite #400
San Jose, CA 95119
United States of America
Email: keyur@arrcus.com

Kotikalapudi Sriram
USA National Institute of Standards and Technology
100 Bureau Drive
Gaithersburg, MD 20899
United States of America
Email: ksriram@nist.gov

IDR Working Group
Internet-Draft
Obsoletes: 5575,7674 (if approved)
Intended status: Standards Track
Expires: April 18, 2021

C. Loibl
next layer Telekom GmbH
S. Hares
Huawei
R. Raszuk
Bloomberg LP
D. McPherson
Verisign
M. Bacher
T-Mobile Austria
October 15, 2020

Dissemination of Flow Specification Rules
draft-ietf-idr-rfc5575bis-27

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 also specifies BGP Extended Community encoding formats, that can be used to propagate Traffic Filtering Actions along with the Flow Specification NLRI. Those Traffic Filtering Actions encode actions a routing system can take if the packet matches the Flow Specification.

Additionally, it defines two applications of that encoding format: one that can be used to automate inter-domain coordination of traffic filtering, such as what is required in order to mitigate (distributed) denial-of-service attacks, and a second application to provide traffic filtering in the context of a BGP/MPLS VPN service. Other applications (e.g. centralized control of traffic in a SDN or NFV context) are also possible. Other documents may specify Flow Specification extensions.

The information is carried via BGP, thereby reusing protocol algorithms, operational experience, and administrative processes such as inter-provider peering agreements.

This document obsoletes both RFC5575 and RFC7674.

Status of This Memo

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

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

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

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

Copyright Notice

Copyright (c) 2020 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	5
4. Dissemination of IPv4 Flow Specification Information	6
4.1. Length Encoding	7
4.2. NLRI Value Encoding	7
4.2.1. Operators	7
4.2.2. Components	9
4.3. Examples of Encodings	14
5. Traffic Filtering	16
5.1. Ordering of Flow Specifications	17
6. Validation Procedure	18
7. Traffic Filtering Actions	19
7.1. Traffic Rate in Bytes (traffic-rate-bytes) sub-type 0x06	21

7.2.	Traffic Rate in Packets (traffic-rate-packets) sub-type	
	TBD	21
7.3.	Traffic-action (traffic-action) sub-type 0x07	21
7.4.	RT Redirect (rt-redirect) sub-type 0x08	22
7.5.	Traffic Marking (traffic-marking) sub-type 0x09	23
7.6.	Interaction with other Filtering Mechanisms in Routers .	23
7.7.	Considerations on Traffic Filtering Action Interference .	24
8.	Dissemination of Traffic Filtering in BGP/MPLS VPN Networks .	24
9.	Traffic Monitoring	25
10.	Error Handling	25
11.	IANA Considerations	25
11.1.	AFI/SAFI Definitions	25
11.2.	Flow Component Definitions	27
11.3.	Extended Community Flow Specification Actions	28
12.	Security Considerations	30
13.	Contributors	32
14.	Acknowledgements	32
15.	References	32
15.1.	Normative References	32
15.2.	Informative References	34
15.3.	URIs	35
Appendix A.	Example Python code: flow_rule_cmp	35
Appendix B.	Comparison with RFC 5575	38
Authors' Addresses	39

1. Introduction

This document obsoletes "Dissemination of Flow Specification Rules" [RFC5575] (see Appendix B for the differences). This document also obsoletes "Clarification of the Flowspec Redirect Extended Community" [RFC7674] since it incorporates the encoding of the BGP Flow Specification Redirect Extended Community in Section 7.4.

Modern IP routers have the capability to forward traffic and to classify, shape, rate limit, filter, or redirect packets based on administratively defined policies. These traffic policy mechanisms allow the operator 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.

Section 4 of this document defines a general procedure to encode Flow Specifications for aggregated traffic flows so that they can be distributed as a BGP [RFC4271] NLRI. Additionally, Section 7 of this

document defines the required Traffic Filtering Actions BGP Extended Communities and mechanisms to use BGP for 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 (Section 6). The Flow Specification received from an internal BGP peer within the same autonomous system [RFC4271] is assumed to have been validated prior to transmission within the internal BGP (iBGP) mesh of an autonomous system. 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 Specifications that may result in different forwarding behavior, as requested by this system.

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 this extension is originated both manually as well as automatically, the latter by systems that are able to detect malicious traffic flows. When automated systems are used, care should be taken to ensure the correctness of the automated system. The the limitations of the receiving systems that need to process these automated Flow Specifications need to be taken in consideration as well (see also Section 12).

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

AFI - Address Family Identifier.

AS - Autonomous System.

Loc-RIB - The Loc-RIB contains the routes that have been selected by the local BGP speaker's Decision Process [RFC4271].

NLRI - Network Layer Reachability Information.

PE - Provider Edge router.

RIB - Routing Information Base.

SAFI - Subsequent Address Family Identifier.

VRF - Virtual Routing and Forwarding instance.

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 Specification if it matches all the specified criteria. This n-tuple is encoded into a BGP NLRI defined below.

A given Flow Specification 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 a 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 must apply to the Flow specification defined NLRI-type. 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

This document defines 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]. When advertising Flow Specifications, the Length of Next Hop Network Address MUST be set to 0. The Network Address of Next Hop field MUST be ignored.

The NLRI field of the MP_REACH_NLRI and MP_UNREACH_NLRI is encoded as one or more 2-tuples of the form <length, NLRI value>. It consists of a 1- or 2-octet length field followed by a variable-length NLRI value. The length is expressed in octets.

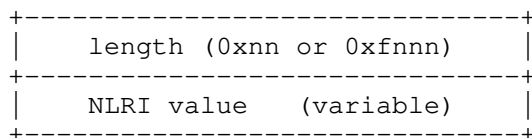


Figure 1: Flow Specification NLRI for IPv4

Implementations wishing to exchange Flow Specification 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

- o If the NLRI length is smaller than 240 (0xf0 hex) octets, the length field can be encoded as a single octet.
- o Otherwise, it is encoded as an extended-length 2-octet value in which the most significant nibble has the hex value 0xf.

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. For example the length value of 239 is encoded as 0xef (single octet) while 240 is encoded as 0xf0f0 (2-octet).

4.2. NLRI Value Encoding

The Flow Specification NLRI value consists of a list of optional components and is encoded as follows:

Encoding: <[component]+>

A specific packet is considered to match the Flow Specification when it matches the intersection (AND) of all the components present in the Flow Specification.

Components MUST follow strict type ordering by increasing numerical order. A given component type MAY (exactly once) be present in the Flow Specification. If present, it MUST precede any component of higher numeric type value.

All combinations of components within a single Flow Specification are allowed. However, some combinations cannot match any packets (e.g. "ICMP Type AND Port" will never match any packets), and thus SHOULD NOT be propagated by BGP.

A NLRI value not encoded as specified here, including a NLRI that contains an unknown component type, is considered malformed and error handling according to Section 10 is performed.

4.2.1. Operators

Most of the components described below make use of comparison operators. Which of the two operators is used is defined by the components in Section 4.2.2. The operators are encoded as a single octet.

4.2.1.1. Numeric Operator (numeric_op)

This operator is encoded as shown in Figure 2.

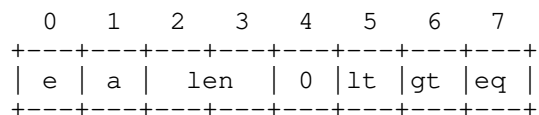


Figure 2: Numeric Operator (numeric_op)

e - end-of-list bit: Set in the last {op, value} pair in the list.

a - AND bit: If unset, the result of the previous {op, value} pair is logically ORed with the current one. If set, the operation is a logical AND. In the first operator octet of a sequence it MUST be encoded as unset 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: The length of the value field for this operator given as $(1 \ll \text{len})$. This encodes 1 (len=00), 2 (len=01), 4 (len=10), 8 (len=11) octets.

0 - MUST 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" as shown in Table 1.

lt	gt	eq	Resulting operation
0	0	0	false (independent of the value)
0	0	1	== (equal)
0	1	0	> (greater than)
0	1	1	>= (greater than or equal)
1	0	0	< (less than)
1	0	1	<= (less than or equal)
1	1	0	!= (not equal value)
1	1	1	true (independent of the value)

Table 1: Comparison operation combinations

4.2.1.2. Bitmask Operator (bitmask_op)

This operator is encoded as shown in Figure 3.

0	1	2	3	4	5	6	7
e	a	len	0	0	not	m	

Figure 3: Bitmask Operator (bitmask_op)

e, a, len - Most significant nibble: (end-of-list bit, AND bit, and length field), as defined in the Numeric Operator format in Section 4.2.1.1.

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

4.2.2. Components

The encoding of each of the components begins with a type field (1 octet) followed by a variable length parameter. The following sections 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].

4.2.2.1. Type 1 - Destination Prefix

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

Defines the destination prefix to match. The length and prefix fields are encoded as in BGP UPDATE messages [RFC4271]

4.2.2.2. Type 2 - Source Prefix

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

Defines the source prefix to match. The length and prefix fields are encoded as in BGP UPDATE messages [RFC4271]

4.2.2.3. Type 3 - IP Protocol

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

Contains a list of {numeric_op, value} pairs that are used to match the IP protocol value octet in IP packet header (see [RFC0791] Section 3.1).

This component uses the Numeric Operator (numeric_op) described in Section 4.2.1.1. Type 3 component values SHOULD be encoded as single octet (numeric_op len=00).

4.2.2.4. Type 4 - Port

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

Defines a list of {numeric_op, value} pairs that matches source OR destination TCP/UDP ports (see [RFC0793] Section 3.1 and [RFC0768] Section "Format"). This component matches if either the destination port OR the source port of a IP packet matches the value.

This component uses the Numeric Operator (numeric_op) described in Section 4.2.1.1. Type 4 component values SHOULD be encoded as 1- or 2-octet quantities (numeric_op len=00 or len=01).

In case of the presence of the port (destination-port Section 4.2.2.5, source-port Section 4.2.2.6) component only TCP or UDP packets can match the entire Flow Specification. The port component, if present, never matches when the packet's IP protocol value is not 6 (TCP) or 17 (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.

4.2.2.5. Type 5 - Destination Port

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

Defines a list of {numeric_op, value} pairs used to match the destination port of a TCP or UDP packet (see also [RFC0793] Section 3.1 and [RFC0768] Section "Format").

This component uses the Numeric Operator (numeric_op) described in Section 4.2.1.1. Type 5 component values SHOULD be encoded as 1- or 2-octet quantities (numeric_op len=00 or len=01).

The last paragraph of Section 4.2.2.4 also applies to this component.

4.2.2.6. Type 6 - Source Port

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

Defines a list of {numeric_op, value} pairs used to match the source port of a TCP or UDP packet (see also [RFC0793] Section 3.1 and [RFC0768] Section "Format").

This component uses the Numeric Operator (numeric_op) described in Section 4.2.1.1. Type 6 component values SHOULD be encoded as 1- or 2-octet quantities (numeric_op len=00 or len=01).

The last paragraph of Section 4.2.2.4 also applies to this component.

4.2.2.7. Type 7 - ICMP type

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

Defines a list of {numeric_op, value} pairs used to match the type field of an ICMP packet (see also [RFC0792] Section "Message Formats").

This component uses the Numeric Operator (numeric_op) described in Section 4.2.1.1. Type 7 component values SHOULD be encoded as single octet (numeric_op len=00).

In case of the presence of the ICMP type component only ICMP packets can match the entire Flow Specification. The ICMP type component, if present, never matches when the packet's IP protocol value is not 1 (ICMP), 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.

4.2.2.8. Type 8 - ICMP code

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

Defines a list of {numeric_op, value} pairs used to match the code field of an ICMP packet (see also [RFC0792] Section "Message Formats").

This component uses the Numeric Operator (numeric_op) described in Section 4.2.1.1. Type 8 component values SHOULD be encoded as single octet (numeric_op len=00).

In case of the presence of the ICMP code component only ICMP packets can match the entire Flow Specification. The ICMP code component, if present, never matches when the packet's IP protocol value is not 1 (ICMP), 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.

4.2.2.9. Type 9 - TCP flags

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

Defines a list of {bitmask_op, bitmask} pairs used to match TCP Control Bits (see also [RFC0793] Section 3.1).

This component uses the Bitmask Operator (bitmask_op) described in Section 4.2.1.2. Type 9 component bitmasks MUST be encoded as 1- or 2-octet bitmask (bitmask_op len=00 or len=01).

When a single octet (bitmask_op len=00) is specified, it matches octet 14 of the TCP header (see also [RFC0793] Section 3.1), which contains the TCP Control Bits. When a 2-octet (bitmask_op len=01) encoding is used, it matches octets 13 and 14 of the TCP header with the data offset (leftmost 4 bits) always treated as 0.

In case of the presence of the TCP flags component only TCP packets can match the entire Flow Specification. The TCP flags component, if present, never matches when the packet's IP protocol value is not 6 (TCP), 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.

4.2.2.10. Type 10 - Packet length

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

Defines a list of {numeric_op, value} pairs used to match on the total IP packet length (excluding Layer 2 but including IP header).

This component uses the Numeric Operator (numeric_op) described in Section 4.2.1.1. Type 10 component values SHOULD be encoded as 1- or 2-octet quantities (numeric_op len=00 or len=01).

4.2.2.11. Type 11 - DSCP (Diffserv Code Point)

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

Defines a list of {numeric_op, value} pairs used to match the 6-bit DSCP field (see also [RFC2474]).

This component uses the Numeric Operator (numeric_op) described in Section 4.2.1.1. Type 11 component values MUST be encoded as single octet (numeric_op len=00).

The six least significant bits contain the DSCP value. All other bits SHOULD be treated as 0.

4.2.2.12. Type 12 - Fragment

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

Defines a list of {bitmask_op, bitmask} pairs used to match specific IP fragments.

This component uses the Bitmask Operator (bitmask_op) described in Section 4.2.1.2. The Type 12 component bitmask MUST be encoded as single octet bitmask (bitmask_op len=00).

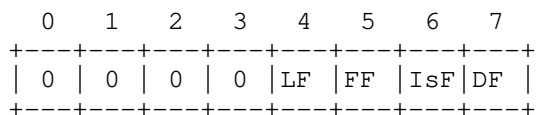


Figure 4: Fragment Bitmask Operand

Bitmask values:

DF - Don't fragment - match if [RFC0791] IP Header Flags Bit-1 (DF) is 1

IsF - Is a fragment other than the first - match if [RFC0791] IP Header Fragment Offset is not 0

FF - First fragment - match if [RFC0791] IP Header Fragment Offset is 0 AND Flags Bit-2 (MF) is 1

LF - Last fragment - match if [RFC0791] IP Header Fragment Offset is not 0 AND Flags Bit-2 (MF) is 0

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

4.3. Examples of Encodings

4.3.1. Example 1

An example of a Flow Specification NLRI encoding for: "all packets to 192.0.2.0/24 and TCP port 25".

length	destination	protocol	port
0x0b	01 18 c0 00 02	03 81 06	04 81 19

Decoded:

Value		
0x0b	length	11 octets (len<240 1-octet)
0x01	type	Type 1 - Destination Prefix
0x18	length	24 bit
0xc0	prefix	192
0x00	prefix	0
0x02	prefix	2
0x03	type	Type 3 - IP Protocol
0x81	numeric_op	end-of-list, value size=1, ==
0x06	value	6 (TCP)
0x04	type	Type 4 - Port
0x81	numeric_op	end-of-list, value size=1, ==
0x19	value	25

This constitutes a NLRI with a NLRI length of 11 octets.

4.3.2. Example 2

An example of a Flow Specification NLRI encoding for: "all packets to 192.0.2.0/24 from 203.0.113.0/24 and port {range [137, 139] or 8080}".

length	destination	source	port
0x12	01 18 c0 00 02	02 18 cb 00 71	04 03 89 45 8b 91 1f 90

Decoded:

Value		
0x12	length	18 octets (len<240 1-octet)
0x01	type	Type 1 - Destination Prefix
0x18	length	24 bit
0xc0	prefix	192
0x00	prefix	0
0x02	prefix	2
0x02	type	Type 2 - Source Prefix
0x18	length	24 bit
0xcb	prefix	203
0x00	prefix	0
0x71	prefix	113
0x04	type	Type 4 - Port
0x03	numeric_op	value size=1, >=
0x89	value	137
0x45	numeric_op	"AND", value size=1, <=
0x8b	value	139
0x91	numeric_op	end-of-list, value size=2, ==
0x1f90	value	8080

This constitutes a NLRI with a NLRI length of 18 octets.

4.3.3. Example 3

An example of a Flow Specification NLRI encoding for: "all packets to 192.0.2.1/32 and fragment { DF or FF } (matching packet with DF bit set or First Fragments)

length	destination	fragment
0x09	01 20 c0 00 02 01	0c 80 05

Decoded:

Value		
0x09	length	9 octets (len<240 1-octet)
0x01	type	Type 1 - Destination Prefix
0x20	length	32 bit
0xc0	prefix	192
0x00	prefix	0
0x02	prefix	2
0x01	prefix	1
0x0c	type	Type 12 - Fragment
0x80	bitmask_op	end-of-list, value size=1
0x05	bitmask	DF=1, FF=1

This constitutes a NLRI with a NLRI length of 9 octets.

5. Traffic Filtering

Traffic filtering policies have been traditionally considered to be relatively static. Limitations of these static mechanisms caused this new dynamic mechanism to be designed for the three new applications of traffic filtering:

- o Prevention of traffic-based, denial-of-service (DOS) attacks.
- o Traffic filtering in the context of BGP/MPLS VPN service.
- o Centralized traffic control for SDN/NFV networks.

These applications require coordination among service providers and/or coordination among the AS within a service provider.

The Flow Specification NLRI defined in Section 4 conveys information about traffic filtering rules for traffic that should be discarded or handled in a 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 document 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:

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

Encoding of the NLRI is described in Section 4 for IPv4 Flow Specification and in Section 8 for VPNv4 Flow Specification. The filtering actions are described in Section 7.

5.1. Ordering of Flow Specifications

More than one Flow Specification may match a particular traffic flow. Thus, it is necessary to define the order in which Flow Specifications get matched and actions being applied to a particular traffic flow. This ordering function is such that it does not depend on the arrival order of the Flow Specification via BGP and thus is consistent in the network.

The relative order of two Flow Specifications is determined by comparing their respective components. The algorithm starts by comparing the left-most components (lowest component type value) of the Flow Specifications. If the types differ, the Flow Specification with lowest numeric type value has higher precedence (and thus will match before) than the Flow Specification 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 one of the two prefixes to compare is a more specific prefix of the other, the more specific prefix has higher precedence. Otherwise 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 [ISO_IEC_9899]. 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 in Appendix A 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/rfc5575bis/tree/master/flowspec-cmp> [1].

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 the Flow Specification NLRI, to allow other validation procedures.

The validation process described below validates Flow Specifications against unicast routes received over the same AFI but the associated unicast routing information SAFI:

Flow Specification received over SAFI=133 will be validated against routes received over SAFI=1

Flow Specification received over SAFI=134 will be validated against routes received over SAFI=128

In the absence of explicit configuration a Flow Specification NLRI MUST be validated such that it is considered feasible if and only if all of the conditions below are true:

- a) A destination prefix component is embedded in the Flow Specification.
- b) The originator of the Flow Specification matches the originator of the best-match unicast route for the destination prefix embedded in the Flow Specification (this is the unicast route with the longest possible prefix length covering the destination prefix embedded in the Flow Specification).

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

However, rule a) MAY be relaxed by explicit 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 address of the originator in the ORIGINATOR_ID Attribute [RFC4456], or the source IP 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 here 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 rules a) to c) as described above.

Explanation:

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

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. The reasoning is that this is as if 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 document defines a minimum set of Traffic Filtering Actions that it standardizes as BGP extended communities [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.

The default action for a matching Flow Specification is to accept the packet (treat the packet according to the normal forwarding behaviour of the system).

This document defines the following extended communities values shown in Table 2 in the form 0xttss where tt indicates the type and ss indicates the sub-type of the extended community. Encodings for these extended communities are described below.

community 0xttss	action	encoding
0x8006	traffic-rate-bytes (Section 7.1)	2-octet AS, 4-octet float
TBD	traffic-rate-packets (Section 7.1)	2-octet AS, 4-octet float
0x8007	traffic-action (Section 7.3)	bitmask
0x8008	rt-redirect AS-2octet (Section 7.4)	2-octet AS, 4-octet value
0x8108	rt-redirect IPv4 (Section 7.4)	4-octet IPv4 address, 2-octet value
0x8208	rt-redirect AS-4octet (Section 7.4)	4-octet AS, 2-octet value
0x8009	traffic-marking (Section 7.5)	DSCP value

Table 2: Traffic Filtering Action Extended Communities

Multiple Traffic Filtering Actions defined in this document may be present for a single Flow Specification and SHOULD be applied to the traffic flow (for example traffic-rate-bytes and rt-redirect can be applied to packets at the same time). If not all of the Traffic Filtering Actions can be applied to a traffic flow they should be treated as interfering Traffic Filtering Actions (see below).

Some Traffic Filtering Actions may interfere with each other or even contradict. Section 7.7 of this document provides general considerations on such Traffic Filtering Action interference. Any additional definition of Traffic Filtering Actions SHOULD specify the action to take if those Traffic Filtering Actions interfere (also with existing Traffic Filtering Actions).

All Traffic Filtering 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-octet AS number. When a 4-octet AS number is locally present, the 2 least significant octets 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: May interfere with the traffic-rate-packets (see Section 7.2). A policy may allow both filtering by traffic-rate-packets and traffic-rate-bytes. If the policy does not allow this, these two actions will conflict.

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: May interfere with the traffic-rate-bytes (see Section 7.1). A policy may allow both filtering by traffic-rate-packets and traffic-rate-bytes. If the policy does not allow this, these two actions will conflict.

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

The traffic-action extended community consists of 6 octets of which only the 2 least significant bits of the 6th octet (from left to right) are defined by this document as shown in Figure 5.

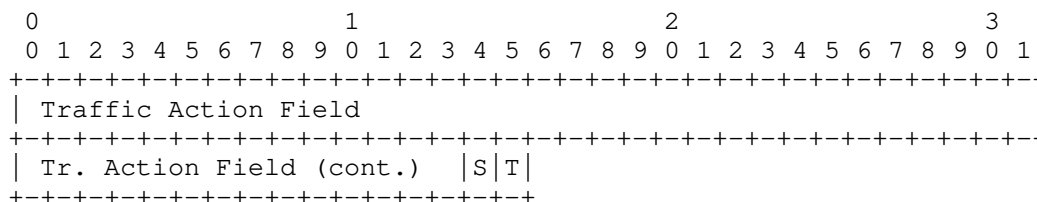


Figure 5: Traffic-action Extended Community Encoding

where S and T are defined as:

- o T: Terminal Action (bit 47): When this bit is set, the traffic filtering engine will evaluate any subsequent Flow Specifications (as defined by the ordering procedure Section 5.1). If not set, the evaluation of the traffic filters stops when this Flow Specification is evaluated.
- o S: Sample (bit 46): Enables traffic sampling and logging for this Flow Specification (only effective when set).
- o Traffic Action Field: Other Traffic Action Field (see Section 11) bits unused in this specification. These bits MUST be set to 0 on encoding, and MUST be ignored during decoding.

The use of the Terminal Action (bit 47) may result in more than one Flow Specification matching a particular traffic flow. All the Traffic Filtering Actions from these Flow Specifications shall be collected and applied. In case of interfering Traffic Filtering Actions it is an implementation decision which Traffic Filtering Actions are selected. See also Section 7.7.

Interferes with: No other BGP Flow Specification Traffic Filtering 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 in Sections 3.1 (type 0x80: 2-octet AS, 4-octet value), 3.2 (type 0x81: 4-octet IPv4 address, 2-octet value) and 4 of [RFC4360] and Section 2 (type 0x82: 4-octet

AS, 2-octet value) of [RFC5668] with the high-order octet of the Type field 0x80, 0x81, 0x82 respectively and the low-order of the Type field (Sub-Type) always 0x08.

Interferes with: No other BGP Flow Specification Traffic Filtering Action in this document.

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

The traffic marking extended community instructs a system to modify the DSCP bits in the IP header ([RFC2474] Section 3) of a transiting IP packet to the corresponding value encoded in the 6 least significant bits of the extended community value as shown in Figure 6.

The extended is encoded as follows:

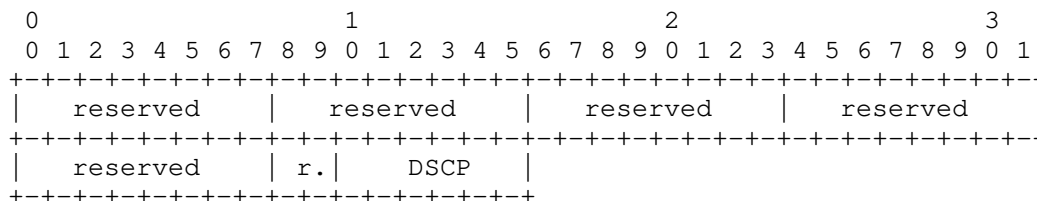


Figure 6: Traffic Marking Extended Community Encoding

- o DSCP: new DSCP value for the transiting IP packet.
- o reserved, r.: MUST be set to 0 on encoding, and MUST be ignored during decoding.

Interferes with: No other BGP Flow Specification Traffic Filtering Action in this document.

7.6. Interaction with other Filtering Mechanisms in Routers

Implementations should provide mechanisms that map an arbitrary BGP community value (normal or extended) to Traffic Filtering Actions that require different mappings on 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.

7.7. Considerations on Traffic Filtering Action Interference

Since Traffic Filtering Actions are represented as BGP extended community values, Traffic Filtering Actions may interfere with each other (e.g. there may be more than one conflicting traffic-rate-bytes Traffic Filtering Action associated with a single Flow Specification). Traffic Filtering Action interference has no impact on BGP propagation of Flow Specifications (all communities are propagated according to policies).

If a Flow Specification associated with interfering Traffic Filtering Actions is selected for packet forwarding, it is an implementation decision which of the interfering Traffic Filtering Actions are selected. Implementors of this specification SHOULD document the behaviour of their implementation in such cases.

Operators are encouraged to make use of the BGP policy framework supported by their implementation in order to achieve a predictable behaviour. See also Section 12.

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 defines an additional BGP NLRI type (AFI=1, SAFI=134) value, which can be used to propagate Flow Specification in a BGP/MPLS VPN environment.

The NLRI format for this address family consists of a fixed-length Route Distinguisher field (8 octets) followed by the Flow Specification NLRI value (Section 4.2). The NLRI length field shall include both the 8 octets of the Route Distinguisher as well as the subsequent Flow Specification NLRI value. The resulting encoding is shown in Figure 7.

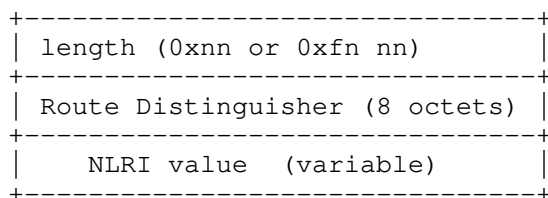


Figure 7: Flow Specification NLRI for MPLS

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 Specifications 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 Specifications are accepted by default, when received from remote PE routers.

The validation procedure (Section 6) and Traffic Filtering Actions (Section 7) are the same as for IPv4.

9. Traffic Monitoring

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

- o A mechanism to log the packet header of filtered traffic.
- o A mechanism to count the number of matches for a given Flow Specification.

10. Error Handling

Error handling according to [RFC7606] and [RFC4760] applies to this specification.

This document introduces Traffic Filtering Action Extended Communities. Malformed Traffic Filtering Action Extended Communities in the sense of [RFC7606] Section 7.14. are Extended Community values that cannot be decoded according to Section 7 of this document.

11. IANA Considerations

This section complies with [RFC7153].

11.1. AFI/SAFI Definitions

IANA maintains a registry entitled "SAFI Values". For the purpose of this work, IANA is requested to update the following SAFIs to read according to the table below (Note: This document obsoletes both

RFC7674 and RFC5575 and all references to those documents should be deleted from the registry below):

Value	Name	Reference
133	Dissemination of Flow Specification rules	[this document]
134	L3VPN Dissemination of Flow Specification rules	[this document]

Table 3: Registry: SAFI Values

The above textual changes generalise the definition of the SAFIs rather than change its underlying meaning. Therefore, based on "The YANG 1.1 Data Modeling Language" [RFC7950], the above text implies that the following YANG enums from "Common YANG Data Types for the Routing Area" [RFC8294] need to have their names and descriptions at <https://www.iana.org/assignments/iana-routing-types> [2] changed to:

```
<CODE BEGINS>
  enum flow-spec-safi {
    value 133;
    description
      "Dissemination of Flow Specification rules SAFI.";
  }
  enum l3vpn-flow-spec-safi {
    value 134;
    description
      "L3VPN Dissemination of Flow Specification rules SAFI.";
  }
<CODE ENDS>
```

A new revision statement should be added to the module as follows:

```
<CODE BEGINS>
  revision [revision date] {
    description "Non-backwards-compatible change of SAFI names
      (SAFI values 133, 134).";
    reference
      "[this document]: Dissemination of Flow Specification Rules.";
  }
<CODE ENDS>
```

11.2. 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". IANA is requested to update the reference for this registry to [this document]. Furthermore the references to the values should be updated according to the table below (Note: This document obsoletes both RFC7674 and RFC5575 and all references to those documents should be deleted from the registry below).

Value	Name	Reference
1	Destination Prefix	[this document]
2	Source Prefix	[this document]
3	IP Protocol	[this document]
4	Port	[this document]
5	Destination port	[this document]
6	Source port	[this document]
7	ICMP type	[this document]
8	ICMP code	[this document]
9	TCP flags	[this document]
10	Packet length	[this document]
11	DSCP	[this document]
12	Fragment	[this document]

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] are used:

Type Values	Policy
0	Reserved
[1 .. 127]	Specification Required
[128 .. 254]	Expert Review
255	Reserved

Table 5: Flow Spec Component Types Policies

Guidance for Experts:

128-254 requires Expert Review as the registration policy. The Experts are expected to check the clarity of purpose and use of the requested code points. The Experts must also verify that

any specification produced in the IETF that requests one of these code points has been made available for review by the IDR working group and that any specification produced outside the IETF does not conflict with work that is active or already published within the IETF. It must be pointed out that introducing new component types may break interoperability with existing implementations of this protocol.

11.3. 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 is requested to update the references to the following entries according to the table below (Note: This document obsoletes both RFC7674 and RFC5575 and all references to those documents should be deleted in the registry below):

Type Value	Name	Reference
0x81	Generic Transitive Experimental Use Extended Community Part 2 (Sub-Types are defined in the "Generic Transitive Experimental Use Extended Community Part 2 Sub-Types" Registry)	[this document]
0x82	Generic Transitive Experimental Use Extended Community Part 3 (Sub-Types are defined in the "Generic Transitive Experimental Use Extended Community Part 3 Sub-Types" Registry)	[this document]

Table 6: Registry: BGP Transitive Extended Community Types

For the sub-type part of the extended community Traffic Filtering Actions IANA maintains the following registries. IANA is requested to update all names and references according to the tables below and assign a new value for the "Flow spec traffic-rate-packets" Sub-Type (Note: This document obsoletes both RFC7674 and RFC5575 and all

references to those documents should be deleted from the registries below).

Sub-Type Value	Name	Reference
0x06	Flow spec traffic-rate-bytes	[this document]
TBD	Flow spec traffic-rate-packets	[this document]
0x07	Flow spec traffic-action (Use of the "Value" field is defined in the "Traffic Action Fields" registry)	[this document]
0x08	Flow spec rt-redirect AS-2octet format	[this document]
0x09	Flow spec traffic-remarking	[this document]

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

Sub-Type Value	Name	Reference
0x08	Flow spec rt-redirect IPv4 format	[this document]

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

Sub-Type Value	Name	Reference
0x08	Flow spec rt-redirect AS-4octet format	[this document]

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

Furthermore IANA is requested to update the reference for the registries "Generic Transitive Experimental Use Extended Community

Part 2 Sub-Types" and "Generic Transitive Experimental Use Extended Community Part 3 Sub-Types" to [this document].

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 registry entitled: "Traffic Action Fields". IANA is requested to update the reference for this registry to [this document]. Furthermore IANA is requested to update the references according to the table below. These values should be assigned via IETF Review rules only (Note: This document obsoletes both RFC7674 and RFC5575 and all references to those documents should be deleted from the registry below).

Bit	Name	Reference
47	Terminal Action	[this document]
46	Sample	[this document]

Table 10: Registry: Traffic Action Fields

12. Security Considerations

As long as Flow Specifications are restricted to match the corresponding unicast routing paths for the relevant prefixes (Section 6), the security characteristics of this proposal are equivalent to the existing security properties of BGP unicast routing. Any relaxation of the validation procedure described in Section 6 may allow unwanted Flow Specifications to be propagated and thus unwanted Traffic Filtering Actions may be applied to flows.

Where the above mechanisms are not in place, this could open the door to further denial-of-service attacks such as unwanted traffic filtering, remarking or redirection.

Deployment of specific relaxations of the validation within an administrative boundary of a network are useful in some networks for quickly distributing filters to prevent denial-of-service attacks. For a network to utilize this relaxation, the BGP policies must support additional filtering since the origin AS field is empty. Specifications relaxing the validation restrictions MUST contain security considerations that provide details on the required additional filtering. For example, the use of Origin validation can provide enhanced filtering within an AS confederation.

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 (unfiltered address space hijack). Since validation of the Flow Specification is tied to the announcement of the best unicast route, the failure in the validation of best path route may prevent the Flow Specification from being used by a local router. Possible mitigations are [RFC6811] and [RFC8205].

On IXPs routes are often exchanged via route servers which do not extend the AS_PATH. In such cases it is not possible to enforce the left-most AS in the AS_PATH to be the neighbor AS (the AS of the route server). Since the validation of Flow Specification (Section 6) depends on this, additional care must be taken. It is advised to use a strict inbound route policy in such scenarios.

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 octets 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.

Flow Specification BGP speakers (e.g. automated DDoS controllers) not properly programmed, algorithms that are not performing as expected, or simply rogue systems may announce unintended Flow Specifications, send updates at a high rate or generate a high number of Flow Specifications. This may stress the receiving systems, exceed their capacity, or lead to unwanted Traffic Filtering Actions being applied to flows.

While the general verification of the Flow Specification 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/screening of the Traffic Filtering Action communities at network borders and only exposing a predefined subset of Traffic Filtering Actions (see Section 7) to third parties. One way to achieve this is by mapping user-defined communities, that can be set by the third party, to Traffic Filtering Actions and not accepting Traffic Filtering Action extended communities from third parties.

This extension adds 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. Service providers need to consider the maximum capacity of their devices and may need to limit the number of Flow Specifications accepted and processed.

13. Contributors

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

14. 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 Traffic Filtering Action was also described in a Flow Specification extension draft and the authors like to thank Wesley Eddy, Justin Dailey and Gilbert Clark for their work.

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

15. References

15.1. Normative References

- [IEEE.754.1985]
IEEE, "Standard for Binary Floating-Point Arithmetic",
IEEE 754-1985, August 1985.
- [ISO_IEC_9899]
ISO, "Information technology -- Programming languages --
C", ISO/IEC 9899:2018, June 2018.
- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768,
DOI 10.17487/RFC0768, August 1980,
<<https://www.rfc-editor.org/info/rfc768>>.

- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/info/rfc791>>.
- [RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <<https://www.rfc-editor.org/info/rfc792>>.
- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <<https://www.rfc-editor.org/info/rfc793>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <<https://www.rfc-editor.org/info/rfc2474>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC4360] Sangli, S., Tappan, D., and Y. Rekhter, "BGP Extended Communities Attribute", RFC 4360, DOI 10.17487/RFC4360, February 2006, <<https://www.rfc-editor.org/info/rfc4360>>.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <<https://www.rfc-editor.org/info/rfc4364>>.
- [RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, DOI 10.17487/RFC4456, April 2006, <<https://www.rfc-editor.org/info/rfc4456>>.
- [RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <<https://www.rfc-editor.org/info/rfc4760>>.

- [RFC5668] Rekhter, Y., Sangli, S., and D. Tappan, "4-Octet AS Specific BGP Extended Community", RFC 5668, DOI 10.17487/RFC5668, October 2009, <<https://www.rfc-editor.org/info/rfc5668>>.
- [RFC7153] Rosen, E. and Y. Rekhter, "IANA Registries for BGP Extended Communities", RFC 7153, DOI 10.17487/RFC7153, March 2014, <<https://www.rfc-editor.org/info/rfc7153>>.
- [RFC7606] Chen, E., Ed., Scudder, J., Ed., Mohapatra, P., and K. Patel, "Revised Error Handling for BGP UPDATE Messages", RFC 7606, DOI 10.17487/RFC7606, August 2015, <<https://www.rfc-editor.org/info/rfc7606>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

15.2. Informative References

- [I-D.ietf-idr-flow-spec-v6] Loibl, C., Raszuk, R., and S. Hares, "Dissemination of Flow Specification Rules for IPv6", draft-ietf-idr-flow-spec-v6-15 (work in progress), September 2020.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<https://www.rfc-editor.org/info/rfc4303>>.
- [RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, DOI 10.17487/RFC5575, August 2009, <<https://www.rfc-editor.org/info/rfc5575>>.
- [RFC6811] Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", RFC 6811, DOI 10.17487/RFC6811, January 2013, <<https://www.rfc-editor.org/info/rfc6811>>.
- [RFC7674] Haas, J., Ed., "Clarification of the Flowspec Redirect Extended Community", RFC 7674, DOI 10.17487/RFC7674, October 2015, <<https://www.rfc-editor.org/info/rfc7674>>.

- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8205] Lepinski, M., Ed. and K. Sriram, Ed., "BGPsec Protocol Specification", RFC 8205, DOI 10.17487/RFC8205, September 2017, <<https://www.rfc-editor.org/info/rfc8205>>.
- [RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <<https://www.rfc-editor.org/info/rfc8294>>.

15.3. URIs

- [1] <https://github.com/stoffi92/rfc5575bis/tree/master/flowspec-cmp>
- [2] <https://www.iana.org/assignments/iana-routing-types>

Appendix A. Example Python code: flow_rule_cmp

<CODE BEGINS>

"""

Copyright (c) 2020 IETF Trust and the persons identified as authors of draft-ietf-idr-rfc5575bis. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).

"""

```
import itertools
import collections
import ipaddress
```

```
EQUAL = 0
```

```
A_HAS_PRECEDENCE = 1
```

```
B_HAS_PRECEDENCE = 2
```

```
IP_DESTINATION = 1
```

```
IP_SOURCE = 2
```

```
FS_component = collections.namedtuple('FS_component',
                                       'component_type op_value')
```

```

class FS_nlri(object):
    """
    FS_nlri class implementation that allows sorting.

    By calling .sort() on a array of FS_nlri objects these will be
    sorted according to the flow_rule_cmp algorithm.

    Example:
    nlri = [ FS_nlri(components=[
                FS_component(component_type=IP_DESTINATION,
                            op_value=ipaddress.ip_network('10.1.0.0/16') ),
                FS_component(component_type=4,
                            op_value=bytearray([0,1,2,3,4,5,6])),
            ]),
            FS_nlri(components=[
                FS_component(component_type=5,
                            op_value=bytearray([0,1,2,3,4,5,6])),
                FS_component(component_type=6,
                            op_value=bytearray([0,1,2,3,4,5,6])),
            ]),
        ]
    nlri.sort() # sorts the array accorindg to the algorithm
    """
    def __init__(self, components = None):
        """
        components: list of type FS_component
        """
        self.components = components

    def __lt__(self, other):
        # use the below algorithm for sorting
        result = flow_rule_cmp(self, other)
        if result == B_HAS_PRECEDENCE:
            return True
        else:
            return False

def flow_rule_cmp(a, b):
    """
    Example of the flowspec comparison algorithm.
    """
    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.op_value, comp_b.op_value of
    # type ipaddress.IPv4Network
    if comp_a.op_value.overlaps(comp_b.op_value):
        # longest prefixlen has precedence
        if comp_a.op_value.prefixlen > \
            comp_b.op_value.prefixlen:
            return A_HAS_PRECEDENCE
        if comp_a.op_value.prefixlen < \
            comp_b.op_value.prefixlen:
            return B_HAS_PRECEDENCE
        # components equal -> continue with next component
    elif comp_a.op_value > comp_b.op_value:
        return B_HAS_PRECEDENCE
    elif comp_a.op_value < comp_b.op_value:
        return A_HAS_PRECEDENCE
else:
    # assuming comp_a.op_value, comp_b.op_value of type
    # bytearray
    if len(comp_a.op_value) == len(comp_b.op_value):
        if comp_a.op_value > comp_b.op_value:
            return B_HAS_PRECEDENCE
        if comp_a.op_value < comp_b.op_value:
            return A_HAS_PRECEDENCE
        # components equal -> continue with next component
    else:
        common = min(len(comp_a.op_value), len(comp_b.op_value))
        if comp_a.op_value[:common] > comp_b.op_value[:common]:
            return B_HAS_PRECEDENCE
        elif comp_a.op_value[:common] < \
            comp_b.op_value[:common]:
            return A_HAS_PRECEDENCE
        # the first common bytes match
        elif len(comp_a.op_value) > len(comp_b.op_value):
            return A_HAS_PRECEDENCE
        else:
            return B_HAS_PRECEDENCE
return EQUAL
<CODE ENDS>
```

Appendix B. Comparison with RFC 5575

This document includes numerous editorial changes to [RFC5575]. It also completely incorporates the redirect action clarification document [RFC7674]. 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 BGPs database. This specification has removed all references to an opaque-key property. BGP implementations are able to understand the NLRI encoding.

Section 4.2.1.1 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 4.2.2.3 - Section 4.2.2.8, Section 4.2.2.10, Section 4.2.2.11 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 Filtering 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 Traffic Filtering 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. This section also completely incorporates the [RFC7674] clarifications of the Flowspec Redirect Extended Community.

Section 7.7 contains general considerations on interfering traffic actions. Section 7.3 also cross-references Section 7.7. [RFC5575] did not mention this.

Section 10 contains new error handling.

Authors' Addresses

Christoph Loibl
next layer Telekom GmbH
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@endzh.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

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 25, 2021

R. Bush
Internet Initiative Japan
J. Haas
J. Scudder
Juniper Networks, Inc.
A. Nipper
C. Dietzel
DE-CIX
September 21, 2020

Making Route Servers Aware of Data Link Failures at IXPs
draft-ietf-idr-rs-bfd-09

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 <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 March 25, 2021.

Copyright Notice

Copyright (c) 2020 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. Definitions	3
3. Overview	4
4. Next Hop Validation	5
4.1. ReachAsk	6
4.2. LocReach	6
4.3. ReachTell	7
4.4. NHIB	7
5. Advertising NH-Reach state in BGP	7
6. Client Procedures for NH-Reach Changes	9
7. Recommendations for Using BFD	9
8. Other Considerations	10
9. Acknowledgments	10
10. IANA Considerations	10
11. Security Considerations	10
12. References	11
12.1. Normative References	11
12.2. Informative References	12
Appendix A. Summary of Document Changes	12
Appendix B. Other Forms of Connectivity Checks	12
Authors' Addresses	13

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

- o 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.
- o 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

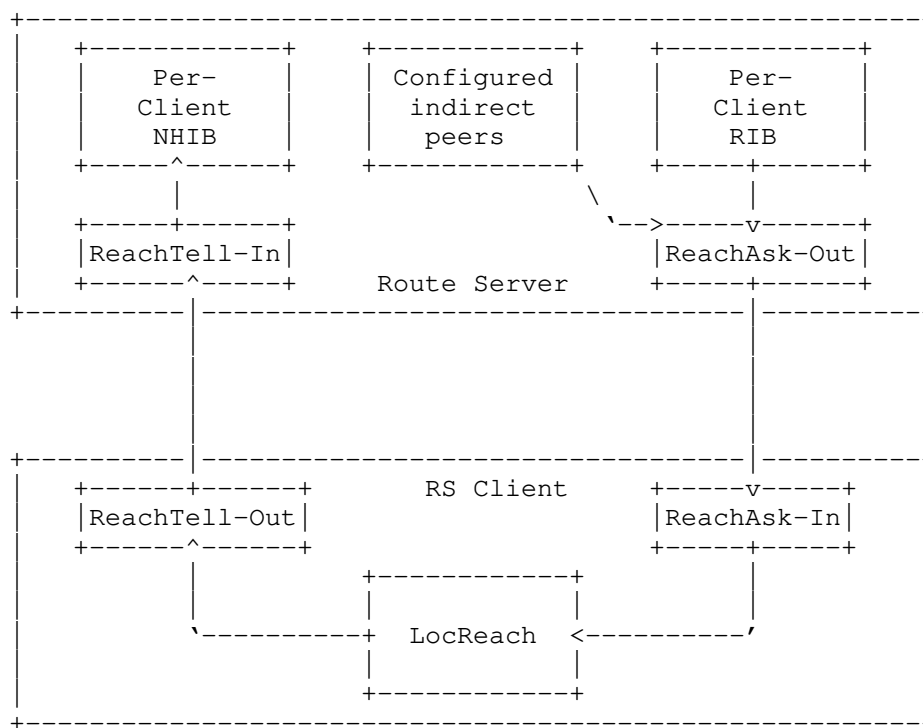
MP-BGP [RFC4760] Next Hop field; this is distinguished by its capitalization and should also be clear from context. Later in that section we define the Indirect Peer's Address field of the NLRI, also called "IPA". It will be clear to the reader that this refers to the "next hops" discussed elsewhere 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].

- o 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:

- o ReachAsk: The reachability request database. A database of next hops (host addresses) for which data plane reachability is being queried.
- o ReachAsk-Out: A set of queries sent to the client.
- o ReachAsk-In: A set of queries received from the route server.
- o ReachTell: The reachability response database. A database of responses to ReachAsk queries, indicating what is known about data plane reachability.
- o ReachTell-Out: The responses being sent to the route server.
- o ReachTell-In: The response received from the client.
- o LocReach: The local reachability database.
- o NHIB: Next Hop Information Base. Stores what is known about the client's reachability to its next hops.



Route Server, RS Client, and Reachability Ask and Tell databases with In/Out Queues

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:

- o The set of next hops found in the associated per-client Loc-RIB (see section 2.3.2.1 of [RFC7947]).
- o The set of addresses of this client's indirect peers (Section 2).
- o 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:

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

- o Up next hops are considered resolvable.
- o Unknown next hops MAY be considered resolvable. They MAY be less preferred for selection.
- o Down next hops MUST NOT be considered resolvable.
- o 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:

```

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

```

NH-Reach NLRI Format

- o 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.
- o Reserved: These five bits are reserved. They MUST be sent as zero and MUST be disregarded on receipt.
- o 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.

- o 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:

- o DesiredMinTxInterval: 1,000,000 (microseconds)
- o RequiredMinRxInterval: 1,000,000 (microseconds)
- o DetectMult: 3

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 subnet 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 thank 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

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <<https://www.rfc-editor.org/info/rfc4760>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<https://www.rfc-editor.org/info/rfc5880>>.
- [RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, DOI 10.17487/RFC5881, June 2010, <<https://www.rfc-editor.org/info/rfc5881>>.
- [RFC7947] Jasinska, E., Hilliard, N., Raszuk, R., and N. Bakker, "Internet Exchange BGP Route Server", RFC 7947, DOI 10.17487/RFC7947, September 2016, <<https://www.rfc-editor.org/info/rfc7947>>.

12.2. Informative References

- [I-D.chen-bfd-unsolicited]
Chen, E., Shen, N., and R. Raszuk, "Unsolicited BFD for Sessionless Applications", draft-chen-bfd-unsolicited-02 (work in progress), January 2018.
- [I-D.ietf-idr-bgp-bestpath-selection-criteria]
Asati, R., "BGP Bestpath Selection Criteria Enhancement", draft-ietf-idr-bgp-bestpath-selection-criteria-12 (work in progress), June 2019.
- [RFC4272] Murphy, S., "BGP Security Vulnerabilities Analysis", RFC 4272, DOI 10.17487/RFC4272, January 2006, <<https://www.rfc-editor.org/info/rfc4272>>.
- [RFC7880] Pignataro, C., Ward, D., Akiya, N., Bhatia, M., and S. Pallagatti, "Seamless Bidirectional Forwarding Detection (S-BFD)", RFC 7880, DOI 10.17487/RFC7880, July 2016, <<https://www.rfc-editor.org/info/rfc7880>>.

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:

- o Seamless BFD [RFC7880]
- o 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

IDR Working Group
Internet-Draft
Obsoletes: 5512, 5566 (if approved)
Updates: 5640 (if approved)
Intended status: Standards Track
Expires: July 11, 2021

K. Patel
Arrcus, Inc
G. Van de Velde
Nokia
S. Sangli
J. Scudder
Juniper Networks
January 7, 2021

The BGP Tunnel Encapsulation Attribute
draft-ietf-idr-tunnel-encaps-22

Abstract

This document defines a BGP Path Attribute known as the "Tunnel Encapsulation Attribute", which can be used with BGP UPDATES of various SAFIs to provide information needed to create tunnels and their corresponding encapsulation headers. It provides encodings for a number of Tunnel Types along with procedures for choosing between alternate tunnels and routing packets into tunnels.

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

Status of This Memo

This 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 July 11, 2021.

Copyright Notice

Copyright (c) 2021 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. Use Case for The Tunnel Encapsulation Attribute	5
1.4. Brief Summary of Changes from RFC 5512	6
1.5. Update to RFC 5640	7
1.6. Effects of Obsoleting RFC 5566	7
2. The Tunnel Encapsulation Attribute	8
3. Tunnel Encapsulation Attribute Sub-TLVs	9
3.1. The Tunnel Egress Endpoint Sub-TLV (type code 6)	9
3.1.1. Validating the Address Subfield	11
3.2. Encapsulation Sub-TLVs for Particular Tunnel Types (type code 1)	12
3.2.1. VXLAN (tunnel type 8)	12
3.2.2. NVGRE (tunnel type 9)	14
3.2.3. L2TPv3 (tunnel type 1)	16
3.2.4. GRE (tunnel type 2)	16
3.2.5. MPLS-in-GRE (tunnel type 11)	17
3.3. Outer Encapsulation Sub-TLVs	17
3.3.1. DS Field (type code 7)	18
3.3.2. UDP Destination Port (type code 8)	18
3.4. Sub-TLVs for Aiding Tunnel Selection	19
3.4.1. Protocol Type Sub-TLV (type code 2)	19
3.4.2. Color Sub-TLV (type code 4)	20
3.5. Embedded Label Handling Sub-TLV (type code 9)	20
3.6. MPLS Label Stack Sub-TLV (type code 10)	21
3.7. Prefix-SID Sub-TLV (type code 11)	23
4. Extended Communities Related to the Tunnel Encapsulation Attribute	24
4.1. Encapsulation Extended Community	24
4.2. Router's MAC Extended Community	25

4.3. Color Extended Community	26
5. Special Considerations for IP-in-IP Tunnels	26
6. Semantics and Usage of the Tunnel Encapsulation attribute . .	26
7. Routing Considerations	29
7.1. Impact on the BGP Decision Process	29
7.2. Looping, Mutual Recursion, Etc.	29
8. Recursive Next Hop Resolution	30
9. Use of Virtual Network Identifiers and Embedded Labels when Imposing a Tunnel Encapsulation	31
9.1. Tunnel Types without a Virtual Network Identifier Field .	31
9.2. Tunnel Types with a Virtual Network Identifier Field . .	31
9.2.1. Unlabeled Address Families	32
9.2.2. Labeled Address Families	32
10. Applicability Restrictions	33
11. Scoping	34
12. Operational Considerations	35
13. Validation and Error Handling	35
14. IANA Considerations	36
14.1. Obsoleting RFC 5512	36
14.2. Obsoleting Code Points Assigned by RFCs 5566	37
14.3. BGP Tunnel Encapsulation Parameters Grouping	37
14.4. BGP Tunnel Encapsulation Attribute Tunnel Types	37
14.5. Subsequent Address Family Identifiers	37
14.6. BGP Tunnel Encapsulation Attribute Sub-TLVs	37
14.7. Flags Field of VXLAN Encapsulation sub-TLV	38
14.8. Flags Field of NVGRE Encapsulation sub-TLV	39
14.9. Embedded Label Handling sub-TLV	39
14.10. Color Extended Community Flags	39
15. Security Considerations	40
16. Acknowledgments	41
17. Contributor Addresses	41
18. References	42
18.1. Normative References	42
18.2. Informative References	44
Appendix A. Impact on RFC 8365	46
Authors' Addresses	46

1. Introduction

This document obsoletes RFC 5512. The deficiencies of RFC 5512, and a summary of the changes made, are discussed in Sections 1.1-1.3. The material from RFC 5512 that is retained has been incorporated into this document. Since [RFC5566] relies on RFC 5512, it is likewise obsoleted.

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

14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.1. Brief Summary of RFC 5512

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

[RFC5512] only allows the Tunnel Encapsulation attribute to be attached to BGP UPDATE messages of the Encapsulation Address Family. These UPDATE messages have an 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:

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

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

1.2. Deficiencies in RFC 5512

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

- o 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 routes to two different address prefixes have the same next hop, [RFC5512] does not provide a straightforward method to associate each prefix with a different tunnel.
- 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, where a sub-TLV may require more than 255 octets for its encoding, a two-octet length field may be needed.

1.3. Use Case for The Tunnel Encapsulation Attribute

Consider the case of a router R1 forwarding an IP packet P. Let D be P's IP destination address. R1 must look up D in its forwarding table. Suppose that the "best match" route for D is route Q, where Q is a BGP-distributed route whose "BGP next hop" is router R2. And suppose further that the routers along the path from R1 to R2 have entries for R2 in their forwarding tables, but do NOT have entries for D in their forwarding tables. For example, the path from R1 to R2 may be part of a "BGP-free core", where there are no BGP-distributed routes at all in the core. Or, as in [RFC5565], D may be an IPv4 address while the intermediate routers along the path from R1 to R2 may support only IPv6.

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

In order for R1 to encapsulate P for transport to R2, R1 must know what encapsulation protocol to use for transporting different sorts of packets to R2. R1 must also know how to fill in the various fields of the encapsulation header. With certain encapsulation

types, this knowledge may be acquired by default or through manual configuration. Other encapsulation protocols have fields such as session id, key, or cookie that must be filled in. It would not be desirable to require every BGP speaker to be manually configured with the encapsulation information for every one of its BGP next hops.

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

1.4. Brief Summary of Changes from RFC 5512

This document addresses the deficiencies identified in Section 1.2 by:

- o Deprecating the Encapsulation SAFI.
- o Defining a new "Tunnel Egress Endpoint sub-TLV" (Section 3.1) that can be included in any of the TLVs contained in the Tunnel Encapsulation attribute. This sub-TLV can be used to specify the remote endpoint address of a particular tunnel.
- 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]),

NVGRE (Network Virtualization Using Generic Routing Encapsulation [RFC7637]), and MPLS-in-GRE (MPLS in Generic Routing Encapsulation [RFC4023]). MPLS-in-UDP [RFC7510] is also supported, but an Encapsulation sub-TLV for it is not needed since there are no additional parameters to be signaled.

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

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

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

1.5. Update to RFC 5640

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

1.6. Effects of Obsoleting RFC 5566

This specification obsoletes RFC 5566. This has the effect of, in turn, obsoleting a number of code points defined in that document. From the "BGP Tunnel Encapsulation Attribute Tunnel Types" registry, "Transmit tunnel endpoint" (type code 3), "IPsec in Tunnel-mode" (type code 4), "IP in IP tunnel with IPsec Transport Mode" (type code 5), and "MPLS-in-IP tunnel with IPsec Transport Mode" (type code 6) are obsoleted. From the "BGP Tunnel Encapsulation Attribute Sub-

TLVs" registry, "IPsec Tunnel Authenticator" (type code 3) is obsoleted. See Section 14.2.

2. The Tunnel Encapsulation Attribute

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

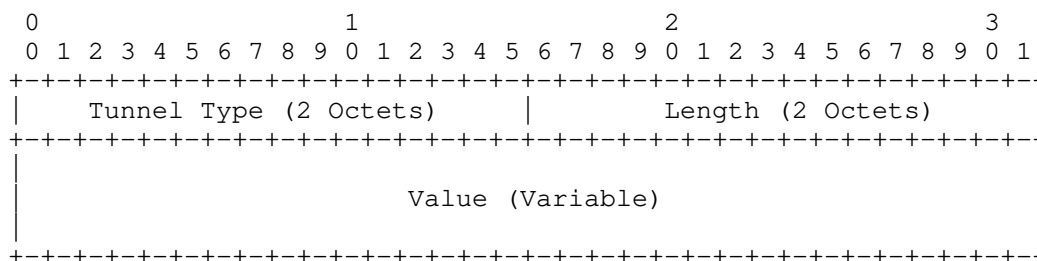


Figure 1: Tunnel Encapsulation TLV Value Field

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

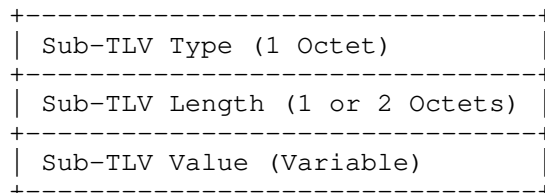


Figure 2: Encapsulation Sub-TLV Value Field

- o Sub-TLV Type (1 octet): each sub-TLV type defines a certain property about the Tunnel TLV that contains this sub-TLV. The field contains values from the IANA Registry "BGP Tunnel Encapsulation Attribute Sub-TLVs".
- o 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.
- o Sub-TLV Value (variable): encodings of the Value field depend on the sub-TLV type as enumerated above. The following sub-sections define the encoding in detail.

3. Tunnel Encapsulation Attribute Sub-TLVs

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

3.1. The Tunnel Egress Endpoint Sub-TLV (type code 6)

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

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

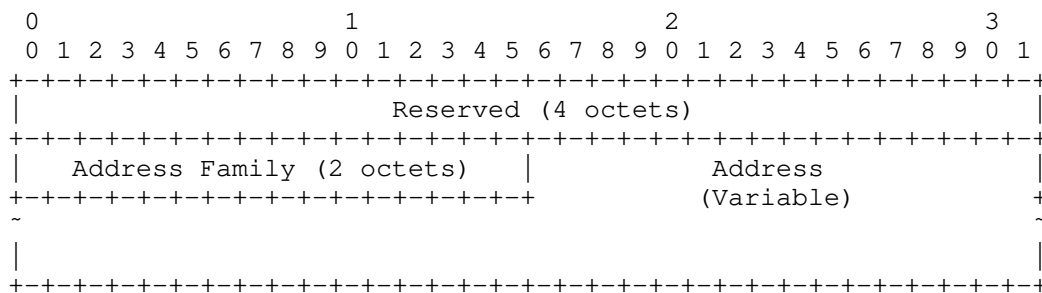


Figure 3: Tunnel Egress Endpoint Sub-TLV Value Field

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

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

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

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

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

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

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

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

- o The length of the sub-TLV's Value field is other than 6 added to the defined length for the address family given in its Address Family subfield. Therefore, for address family behaviors defined in this document, the permitted values are:
 - * 10, if the Address Family subfield contains the value for IPv4.

- * 22, if the Address Family subfield contains the value for IPv6.
- * 6, if the Address Family subfield contains the value zero.
- o The IP address in the sub-TLV's Address subfield lies within a block listed in the relevant Special-Purpose IP Address Registry [RFC6890] with either a "destination" attribute value or a "forwardable" attribute value of "false". (Such routes are sometimes colloquially known as "Martians".) This restriction MAY be relaxed by explicit configuration.
- o It can be determined that the IP address in the sub-TLV's Address subfield does not belong to the Autonomous System (AS) that originated the route that contains the attribute. Section 3.1.1 describes an optional procedure to make this determination.

Error Handling is specified in Section 13.

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

3.1.1. Validating the Address Subfield

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

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

In order to determine the Route Origin ASN of the address depicted in the Address subfield of the Tunnel Egress Endpoint sub-TLV, it is necessary to consider the forwarding route, that is, the route that will be used to forward traffic toward that address. This route is determined by a recursive route lookup operation for that address, as

discussed in [RFC4271] Section 5.1.3. The relevant AS Path to consider is the last one encountered while performing the recursive lookup; the procedures of RFC6811 Section 2 are applied to that AS Path to determine the Route Origin ASN. If no AS Path is encountered at all, for example if that route's source is a protocol other than BGP, the Route Origin ASN is the BGP speaker's own AS number. Call this value Egress_AS.

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

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

3.2. Encapsulation Sub-TLVs for Particular Tunnel Types (type code 1)

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

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

Recall that the tunnel type itself is identified by the Tunnel Type field in the attribute header (Section 2); the Encapsulation sub-TLV's structure is inferred from this. Regardless of the Tunnel Type, the sub-TLV type of the Encapsulation sub-TLV is 1. There are also tunnel types for which it is not necessary to define an Encapsulation sub-TLV, because there are no fields in the encapsulation header whose values need to be signaled from the tunnel egress endpoint.

3.2.1. VXLAN (tunnel type 8)

This document defines an Encapsulation sub-TLV for VXLAN [RFC7348] tunnels. When the Tunnel Type is VXLAN, the length of the sub-TLV is 12 octets. The following is the structure of the Value field in the Encapsulation sub-TLV:

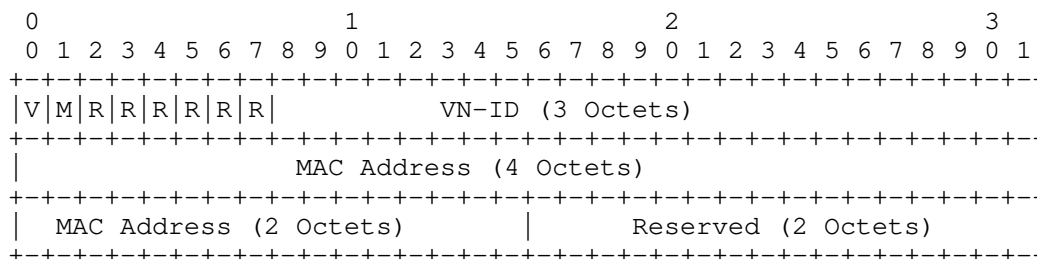


Figure 4: VXLAN Encapsulation Sub-TLV Value Field

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

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

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

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

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

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

When forming the VXLAN encapsulation header:

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

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

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

- o If the V bit is not set, and the BGP UPDATE message has AFI/SAFI other than Ethernet VPNs (SAFI 70, "BGP EVPNs") then the VXLAN tunnel cannot be used.
- o Section 9 describes how the VNI field of the VXLAN encapsulation header is set.

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

3.2.2. NVGRE (tunnel type 9)

This document defines an Encapsulation sub-TLV for NVGRE [RFC7637] tunnels. When the Tunnel Type is NVGRE, the length of the sub-TLV is 12 octets. The following is the structure of the Value field in the Encapsulation sub-TLV:

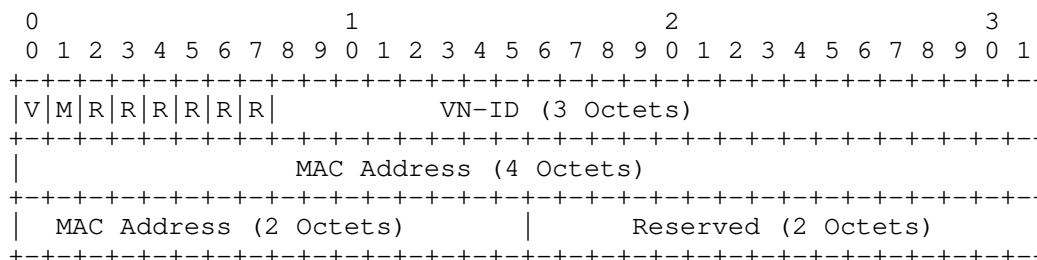


Figure 5: NVGRE Encapsulation Sub-TLV Value Field

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

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

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

VN-ID: If the V bit is set, the VN-ID field contains a 3 octet VN-ID value, used to set the NVGRE VSID (see Section 9). If the V bit is not set, the VN-ID field MUST be set to zero on transmission and disregarded on receipt.

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

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

When forming the NVGRE encapsulation header:

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

- o If the V bit is not set, and the BGP UPDATE message has AFI/SAFI other than Ethernet VPNs (EVPN) then the NVGRE tunnel cannot be used.
- o Section 9 describes how the VSID (Virtual Subnet Identifier) field of the NVGRE encapsulation header is set.

3.2.3. L2TPv3 (tunnel type 1)

When the Tunnel Type of the TLV is L2TPv3 over IP [RFC3931], the length of the sub-TLV is between 4 and 12 octets, depending on the length of the cookie. The following is the structure of the Value field of the Encapsulation sub-TLV:

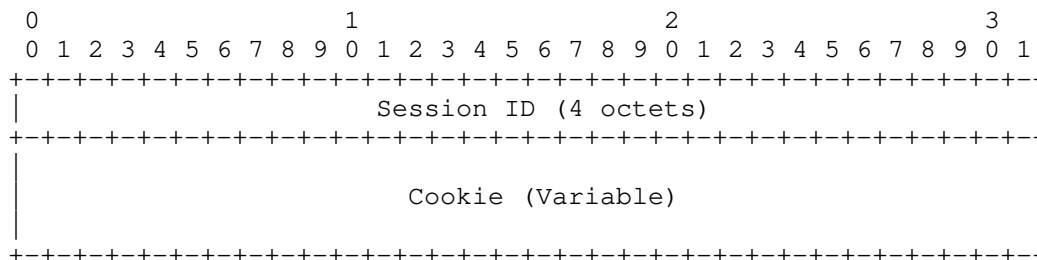


Figure 6: L2TPv3 Encapsulation Sub-TLV Value Field

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

Cookie: an optional, variable length (encoded in 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).

3.2.4. GRE (tunnel type 2)

When the Tunnel Type of the TLV is GRE [RFC2784], the length of the sub-TLV is 4 octets. The following is the structure of the Value field of the Encapsulation sub-TLV:

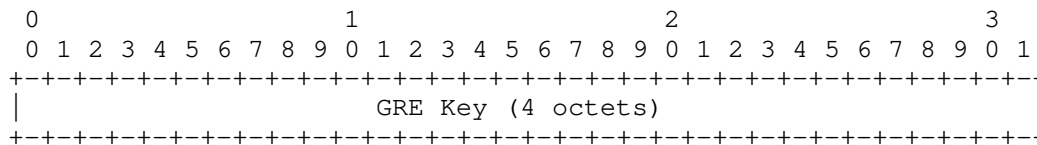


Figure 7: GRE Encapsulation Sub-TLV

GRE Key: 4-octet field [RFC2890] that is generated by the advertising router. Note that the key is optional. Unless a key

value is being advertised, the GRE Encapsulation sub-TLV MUST NOT be present.

3.2.5. MPLS-in-GRE (tunnel type 11)

When the Tunnel Type is MPLS-in-GRE [RFC4023], the length of the sub-TLV is 4 octets. The following is the structure of the Value field of the Encapsulation sub-TLV:

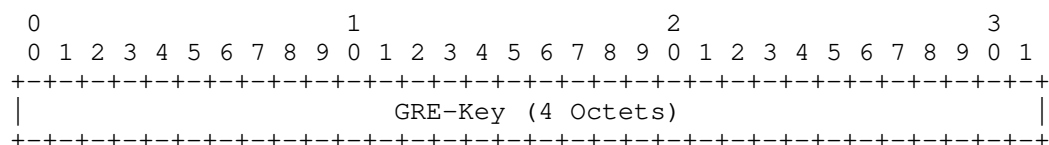


Figure 8: MPLS-in-GRE Encapsulation Sub-TLV Value Field

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

Note that the GRE Tunnel Type defined in Section 3.2.4 can be used instead of the MPLS-in-GRE Tunnel Type when it is necessary to encapsulate MPLS in GRE. Including a TLV of the MPLS-in-GRE tunnel type is equivalent to including a TLV of the GRE Tunnel Type that also includes a Protocol Type sub-TLV (Section 3.4.1) specifying MPLS as the protocol to be encapsulated.

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

3.3. Outer Encapsulation Sub-TLVs

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

If an outer Encapsulation sub-TLV occurs in a TLV for a Tunnel Type that does not use the corresponding outer encapsulation, the sub-TLV MUST be treated as if it were an unrecognized type of sub-TLV.

3.3.1. DS Field (type code 7)

Most of the tunnel types that can be specified in the Tunnel Encapsulation attribute require an outer IP encapsulation. The Differentiated Services (DS) Field sub-TLV can be carried in the TLV of any such Tunnel Type. It specifies the setting of the one-octet Differentiated Services field in the outer IPv4 or IPv6 encapsulation (see [RFC2474]). Any one-octet value can be transported; the semantics of the DSCP field is beyond the scope of this document. The Value field is always a single octet.

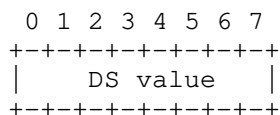


Figure 9: DS Field Sub-TLV Value Field

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

3.3.2. UDP Destination Port (type code 8)

Some of the tunnel types that can be specified in the Tunnel Encapsulation attribute require an outer UDP encapsulation. Generally there is a standard UDP Destination Port value for a particular Tunnel Type. However, sometimes it is useful to be able to use a 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. Any two-octet value other than zero can be transported. If the reserved value zero is received, the sub-TLV MUST be treated as malformed according to the rules of Section 13.

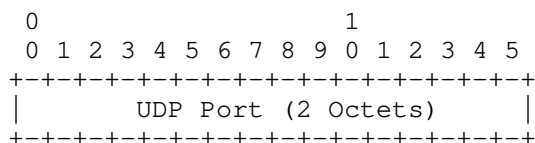


Figure 10: UDP Destination Port Sub-TLV Value Field

3.4. Sub-TLVs for Aiding Tunnel Selection

3.4.1. Protocol Type Sub-TLV (type code 2)

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

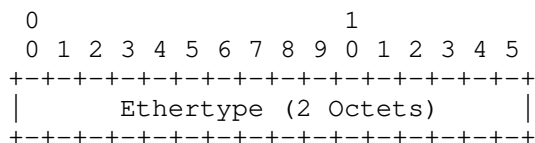


Figure 11: Protocol Type Sub-TLV Value Field

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

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

for "X-in-Y" type tunnels, a Protocol Type sub-TLV specifying anything other than "X" MUST be ignored; this is discussed further in Section 13.

3.4.2. Color Sub-TLV (type code 4)

The Color sub-TLV MAY be used as a way to "color" the corresponding Tunnel TLV. The Value field of the sub-TLV is eight octets long, and consists of a Color Extended Community, as defined in Section 4.3. For the use of this sub-TLV and Extended Community, please see Section 8.

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

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

3.5. Embedded Label Handling Sub-TLV (type code 9)

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

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

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

The Embedded Label Handling sub-TLV may be included in any TLV of the Tunnel Encapsulation attribute. If the Tunnel Encapsulation attribute is attached to an UPDATE of a non-labeled address family, then the sub-TLV MUST be disregarded. If the sub-TLV is contained in a TLV whose Tunnel Type does not have a virtual network identifier in

its encapsulation header, the sub-TLV MUST be disregarded. In those cases where the sub-TLV is ignored, it MUST NOT be stripped from the TLV before the route is propagated.

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

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

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

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

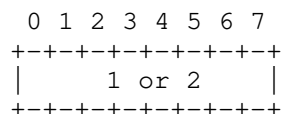


Figure 12: Embedded Label Handling Sub-TLV Value Field

3.6. MPLS Label Stack Sub-TLV (type code 10)

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

The length of the sub-TLV is a multiple of 4 octets and the Value field of this sub-TLV is a sequence of MPLS label stack entries. The first entry in the sequence is the "topmost" label, the final entry in the sequence is the "bottommost" label. When this label stack is pushed onto a packet, this ordering MUST be preserved.

Each label stack entry has the following format:

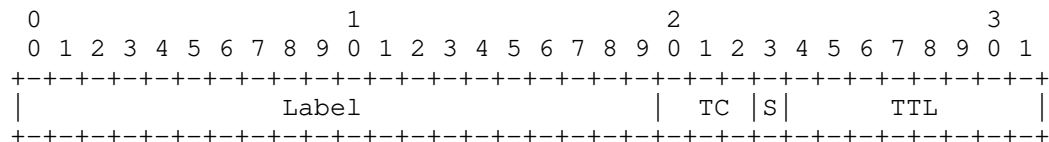


Figure 13: MPLS Label Stack Sub-TLV Value Field

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

If a packet is to be sent through the tunnel identified in a particular TLV, and if that TLV contains an MPLS Label Stack sub-TLV, then the label stack appearing in the sub-TLV MUST be pushed onto the packet before any other labels are pushed onto the packet. (See Section 6 for further discussion.)

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

If the MPLS Label Stack sub-TLV is included in a TLV identifying a Tunnel Type that uses virtual network identifiers (see Section 9), the contents of the MPLS Label Stack sub-TLV MUST be pushed onto the packet before the procedures of Section 9 are applied.

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

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

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

Note that this sub-TLV can appear within a TLV identifying any type of tunnel, not just within a TLV identifying an MPLS tunnel. However, if this sub-TLV appears within a TLV identifying an MPLS

tunnel (or an MPLS-in-X tunnel), this sub-TLV plays the same role that would be played by an MPLS Encapsulation sub-TLV. Therefore, an MPLS Encapsulation sub-TLV is not defined.

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

3.7. Prefix-SID Sub-TLV (type code 11)

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

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

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

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

If a Label-Index is present in the Prefix-SID sub-TLV, then when a packet is sent through the tunnel identified by the TLV, if that tunnel is from a labeled address family, the corresponding MPLS label MUST be pushed on the packet's label stack. The corresponding MPLS label is computed from the Label-Index value and the SRGB of the route's originator, as specified in section 4.1 of [RFC8669].

The corresponding MPLS label is pushed on after the processing of the MPLS Label Stack sub-TLV, if present, as specified in Section 3.6. It is pushed on before any other labels (for example, a label

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

The Prefix-SID sub-TLV has slightly different semantics than the Prefix-SID attribute. When the Prefix-SID attribute is attached to a given route, the BGP speaker that originally attached the attribute is expected to be in the same Segment Routing domain as the BGP speakers who receive the route with the attached attribute. The Label-Index tells the receiving BGP speakers what the prefix-SID is for the advertised prefix in that Segment Routing domain. When the Prefix-SID sub-TLV is used, there is no implication that the prefix-SID for the advertised prefix is the same in the Segment Routing domains of the BGP speaker that originated the sub-TLV and the BGP speaker that received it.

4. Extended Communities Related to the Tunnel Encapsulation Attribute

4.1. Encapsulation Extended Community

The Encapsulation Extended Community is a Transitive Opaque Extended Community.

The Encapsulation Extended Community encoding is as shown below

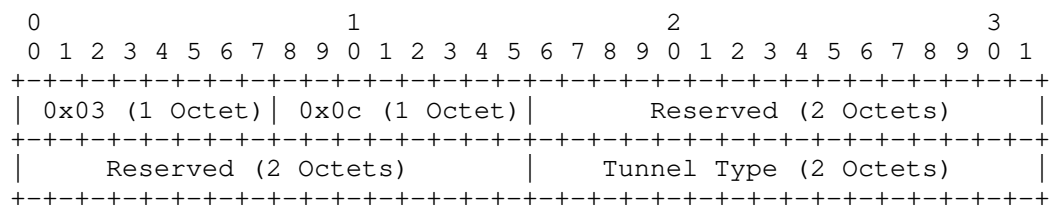


Figure 14: Encapsulation Extended Community

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

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

This Extended Community may be attached to a route of any AFI/SAFI to which the Tunnel Encapsulation attribute may be attached. Each such Extended Community identifies a particular Tunnel Type, its semantics are the same as semantics of a Tunnel Encapsulation attribute Tunnel TLV for which the following three conditions all hold:

1. it identifies the same Tunnel Type,

2. it has a Tunnel Egress Endpoint sub-TLV for which one of the following two conditions holds:
 - A. its "Address Family" subfield contains zero, or
 - B. its "Address" subfield contains the address of the next hop field of the route to which the Tunnel Encapsulation attribute is attached
3. it has no other sub-TLVs.

Such a Tunnel TLV is called a "barebones" Tunnel TLV.

The Encapsulation Extended Community was first defined in [RFC5512]. While it provides only a small subset of the functionality of the Tunnel Encapsulation attribute, it is used in a number of deployed applications, and is still needed for backwards compatibility. In situations where a tunnel could be encoded using a barebones TLV, it MUST be encoded using the corresponding Encapsulation Extended Community. Notwithstanding, an implementation MUST be prepared to process a tunnel received encoded as a barebones TLV.

Note that for tunnel types of the form "X-in-Y", for example, MPLS-in-GRE, the Encapsulation Extended Community implies that only packets of the specified payload type "X" are to be carried through the tunnel of type "Y". Packets with other payload types MUST NOT be carried through such tunnels. See also Section 2.

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

4.2. Router's MAC Extended Community

[I-D.ietf-bess-evpn-inter-subnet-forwarding] defines a Router's MAC Extended Community. This Extended Community, as its name implies, carries the MAC address of the advertising router. Since the VXLAN and NVGRE Encapsulation Sub-TLVs can also optionally carry a router's MAC, a conflict can arise if both the Router's MAC Extended Community and such an Encapsulation Sub-TLV are present at the same time but have different values. In case of such a conflict, the information in the Router's MAC Extended Community MUST be used.

4.3. Color Extended Community

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

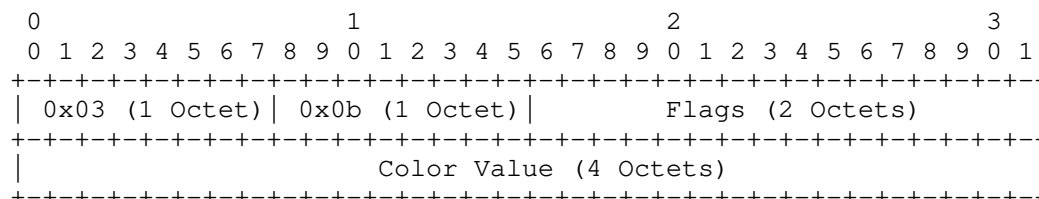


Figure 15: Color Extended Community

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

5. Special Considerations for IP-in-IP Tunnels

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

6. Semantics and Usage of the Tunnel Encapsulation attribute

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

There is no significance to the order in which the TLVs occur within the Tunnel Encapsulation attribute. Multiple TLVs may occur for a given Tunnel Type; each such TLV is regarded as describing a different tunnel. (This also applies if the Tunnel Encapsulation Extended Community encoding is used.)

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

Suppose that:

- o a given packet P must be forwarded by router R;
- o the path along which P is to be forwarded is determined by BGP UPDATE U;
- o UPDATE U has a Tunnel Encapsulation attribute, containing at least one TLV that identifies a "feasible tunnel" for packet P. A tunnel is considered feasible if it has the following four properties:
 - * The Tunnel Type is supported (that is, router R knows how to set up tunnels of that type, how to create the encapsulation header for tunnels of that type, etc.)
 - * The tunnel is of a type that can be used to carry packet P (for example, an MPLS-in-UDP tunnel would not be a feasible tunnel for carrying an IP packet, unless the IP packet can first be encapsulated in a MPLS packet).
 - * The tunnel is specified in a TLV whose Tunnel Egress Endpoint sub-TLV identifies an IP address that is reachable. The reachability condition is evaluated as per [RFC4271]. If the IP address is reachable via more than one forwarding table, local policy is used to determine which table to use.
 - * There is no local policy that prevents the use of the tunnel.

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

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

minimization of encapsulation header overhead, could also be used to influence selection.

The reachability to the address of the egress endpoint of the tunnel may change over time, directly impacting the feasibility of the tunnel. A tunnel that is not feasible at some moment, may become feasible at a later time when its egress endpoint address is reachable. The router may start using the newly feasible tunnel instead of an existing one. How this decision is made is outside the scope of this document.

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

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

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

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

Once router R has decided to send packet P through a particular tunnel, it encapsulates packet P appropriately and then forwards it according to the route that leads to the tunnel's egress endpoint. This route may itself be a BGP route with a Tunnel Encapsulation

attribute. If so, the encapsulated packet is treated as the payload and is encapsulated according to the Tunnel Encapsulation attribute of that route. That is, tunnels may be "stacked".

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

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

7. Routing Considerations

7.1. Impact on the BGP Decision Process

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

7.2. Looping, Mutual Recursion, Etc.

Consider a packet destined for address X. Suppose a BGP UPDATE for address prefix X carries a Tunnel Encapsulation attribute that specifies a tunnel egress endpoint of Y, and suppose that a BGP UPDATE for address prefix Y carries a Tunnel Encapsulation attribute that specifies a tunnel egress endpoint of X. It is easy to see that this can have no good outcome. [RFC4271] describes an analogous case as mutually recursive routes.

This could happen as a result of misconfiguration, either accidental or intentional. It could also happen if the Tunnel Encapsulation attribute were altered by a malicious agent. Implementations should be aware that such an attack will result in unresolvable BGP routes due to the mutually recursive relationship. This document does not specify a maximum number of recursions; that is an implementation-specific matter.

Improper setting (or malicious altering) of the Tunnel Encapsulation attribute could also cause data packets to loop. Suppose a BGP UPDATE for address prefix X carries a Tunnel Encapsulation attribute that specifies a tunnel egress endpoint of Y. Suppose router R receives and processes the advertisement. When router R receives a

packet destined for X, it will apply the encapsulation and send the encapsulated packet to Y. Y will decapsulate the packet and forward it further. If Y is further away from X than is router R, it is possible that the path from Y to X will traverse R. This would cause a long-lasting routing loop. The control plane itself cannot detect this situation, though a TTL field in the payload packets would prevent any given packet from looping infinitely.

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

8. Recursive Next Hop Resolution

Suppose that:

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

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

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

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

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

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

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

9.1. Tunnel Types without a Virtual Network Identifier Field

If a Tunnel Encapsulation attribute is attached to an UPDATE of a labeled address family, there will be one or more labels specified in the UPDATE's NLRI. When a packet is sent through a tunnel specified in one of the attribute's TLVs, and that tunnel type does not contain a virtual network identifier field, the label or labels from the NLRI are pushed on the packet's label stack. The resulting MPLS packet is then further encapsulated, as specified by the TLV.

9.2. Tunnel Types with a Virtual Network Identifier Field

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

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

This section specifies the procedures for choosing the value to set in the virtual network identifier field of the encapsulation header. These procedures apply only when the Tunnel Type is VXLAN or NVGRE.

9.2.1. Unlabeled Address Families

This sub-section applies when:

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

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

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

9.2.2. Labeled Address Families

This sub-section applies when:

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

9.2.2.1. When a Valid VNI has been Signaled

If the TLV identifying the tunnel contains an Encapsulation sub-TLV whose V bit is set, the virtual network identifier field of the encapsulation header is set to the value of the virtual network identifier field of the Encapsulation sub-TLV. However, the Embedded Label Handling sub-TLV will determine label processing as described below.

- o If the TLV contains an Embedded Label Handling sub-TLV whose value is 1, the embedded label (from the NLRI of the route that is carrying the Tunnel Encapsulation attribute) appears at the top of the MPLS label stack in the encapsulation payload.

- o If the TLV does not contain an Embedded Label Handling sub-TLV, or it contains an Embedded Label Handling sub-TLV whose value is 2, the embedded label is ignored entirely.

9.2.2.2. When a Valid VNI has not been Signaled

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

- o 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.

- o If the TLV does not contain an Embedded Label Handling sub-TLV, or if it contains an Embedded Label Handling sub-TLV whose value is 2, the embedded label is copied into the lower 3 octets of the virtual network identifier field of the encapsulation header.

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

10. Applicability Restrictions

In a given UPDATE of a labeled address family, the label embedded in the NLRI is generally a label that is meaningful only to the router represented by the address of the next hop. Certain of the procedures of Section 9.2.2.1 or Section 9.2.2.2 cause the embedded label to be carried by a data packet to the router whose address appears in the Tunnel Egress Endpoint sub-TLV. If the Tunnel Egress Endpoint sub-TLV does not identify the same router represented by the address of the next hop, sending the packet through the tunnel may cause the label to be misinterpreted at the tunnel's egress endpoint. This may cause misdelivery of the packet. Avoidance of this unfortunate outcome is a matter of network planning and design, and is outside the scope of this document.

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

contains an IP address that is not in same AS as the router receiving the route, it is very likely that the embedded label has been changed. Therefore use of the Tunnel Encapsulation attribute in an "Inter-AS option b" scenario is not recommended.

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

11. Scoping

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

To prevent the Tunnel Encapsulation attribute from being distributed beyond its intended scope, any BGP speaker that understands the attribute MUST be able to filter the attribute from incoming BGP UPDATE messages. When the attribute is filtered from an incoming UPDATE, the attribute is neither processed nor distributed. This filtering SHOULD be possible on a per-BGP-session basis; finer granularities (for example, per route and/or per attribute TLV) MAY be supported. For each external BGP (EBGP) session, filtering of the attribute on incoming UPDATES MUST be enabled by default.

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

Since the Tunnel Encapsulation Extended Community provides a subset of the functionality of the Tunnel Encapsulation attribute, these considerations apply equally in its case: any BGP speaker that understands it MUST be able to filter it from incoming BGP UPDATE messages, it MUST be possible to filter the Tunnel Encapsulation Extended Community from outgoing messages, and in both cases this filtering MUST be enabled by default for EBGP sessions.

12. Operational Considerations

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

13. Validation and Error Handling

The Tunnel Encapsulation attribute is a sequence of TLVs, each of which is a sequence of sub-TLVs. The final octet of a TLV is determined by its length field. Similarly, the final octet of a sub-TLV is determined by its length field. The final octet of a TLV MUST also be the final octet of its final sub-TLV. If this is not the case, the TLV MUST be considered to be malformed, and the "Treat-as-withdraw" procedure of [RFC7606] is applied.

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

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

The following sub-TLVs defined in this document MUST NOT occur more than once in a given Tunnel TLV: Tunnel Egress Endpoint (discussed below), Encapsulation, DS, UDP Destination Port, Embedded Label Handling, MPLS Label Stack, Prefix-SID. If a Tunnel TLV has more than one of any of these sub-TLVs, all but the first occurrence of each such sub-TLV type MUST be disregarded. However, the Tunnel TLV containing them MUST NOT be considered to be malformed, and all the sub-TLVs MUST be propagated if the route carrying the Tunnel Encapsulation attribute is propagated.

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

sub-TLV may appear multiple times to assign multiple colors to a tunnel.

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

In general, if a TLV contains a sub-TLV that is malformed, the sub-TLV MUST be treated as if it were an unrecognized sub-TLV. There is one exception to this rule -- if a TLV contains a malformed Tunnel Egress Endpoint sub-TLV (as defined in Section 3.1), the entire TLV MUST be ignored, and MUST be removed from the Tunnel Encapsulation attribute before the route carrying that attribute is distributed.

Within a Tunnel Encapsulation attribute that is carried by a BGP UPDATE whose AFI/SAFI is one of those explicitly listed in the second paragraph of Section 6, a TLV that does not contain exactly one Tunnel Egress Endpoint sub-TLV MUST be treated as if it contained a malformed Tunnel Egress Endpoint sub-TLV.

A TLV identifying a particular Tunnel Type may contain a sub-TLV that is meaningless for that Tunnel Type. For example, perhaps the TLV contains a UDP Destination Port sub-TLV, but the identified tunnel type does not use UDP encapsulation at all, or a tunnel of the form "X-in-Y" contains a Protocol Type sub-TLV that specifies something other than "X". Sub-TLVs of this sort MUST be disregarded. That is, they MUST NOT affect the creation of the encapsulation header. However, the sub-TLV MUST NOT be considered to be malformed, and MUST NOT be removed from the TLV before the route carrying the Tunnel Encapsulation attribute is distributed. An implementation MAY log a message when it encounters such a sub-TLV.

14. IANA Considerations

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

14.1. Obsoleting RFC 5512

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

14.2. Obsoleting Code Points Assigned by RFCs 5566

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

In the "BGP Tunnel Encapsulation Attribute Tunnel Types" registry:

Value	Name
3	Transmit tunnel endpoint
4	IPsec in Tunnel-mode
5	IP in IP tunnel with IPsec Transport Mode
6	MPLS-in-IP tunnel with IPsec Transport Mode

And in the "BGP Tunnel Encapsulation Attribute Sub-TLVs" registry:

Value	Name
3	IPsec Tunnel Authenticator

14.3. BGP Tunnel Encapsulation Parameters Grouping

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

14.4. BGP Tunnel Encapsulation Attribute Tunnel Types

Relocate the "BGP Tunnel Encapsulation Attribute Tunnel Types" registry to be under the "BGP Tunnel Encapsulation Parameters" grouping.

14.5. Subsequent Address Family Identifiers

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

14.6. BGP Tunnel Encapsulation Attribute Sub-TLVs

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

Add the following note to the registry:

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

Change the registration policy of the registry to the following:

Value(s)	Registration Procedure
0	Reserved
1-63	Standards Action
64-125	First Come First Served
126-127	Experimental Use
128-191	Standards Action
192-252	First Come First Served
253-254	Experimental Use
255	Reserved

Rename the following entries within the registry:

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

14.7. Flags Field of VXLAN Encapsulation sub-TLV

Create a registry named "Flags Field of VXLAN Encapsulation sub-TLV" under the "BGP Tunnel Encapsulation Parameters" grouping. The registration policy for this registry is "Standards Action". The minimum possible value is 0, the maximum is 7.

The initial values for this new registry are indicated below.

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

14.8. Flags Field of NVGRE Encapsulation sub-TLV

Create a registry named "Flags Field of NVGRE Encapsulation sub-TLV" under the "BGP Tunnel Encapsulation Parameters" grouping. The registration policy for this registry is "Standards Action". The minimum possible value is 0, the maximum is 7.

The initial values for this new registry are indicated below.

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

14.9. Embedded Label Handling sub-TLV

Create a registry named "Embedded Label Handling sub-TLV" under the "BGP Tunnel Encapsulation Parameters" grouping. The registration policy for this registry is "Standards Action". The minimum possible value is 0, the maximum is 255.

The initial values for this new registry are indicated below.

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

14.10. Color Extended Community Flags

Create a registry named "Color Extended Community Flags" under the "BGP Tunnel Encapsulation Parameters" grouping. The registration policy for this registry is "Standards Action". The minimum possible value is 0, the maximum is 15.

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

Bit Position	Description	Reference
--------------	-------------	-----------

15. Security Considerations

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

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

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

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

of this occurring. See the discussion of "option b" in Section 10 for further discussion of one such scenario.

RFC 8402 specifies that "SR domain boundary routers MUST filter any external traffic" ([RFC8402] Section 8.1). For these purposes, traffic introduced into a SR domain using the Prefix-SID sub-TLV lies within the SR domain, even though the prefix-SIDs used by the routers at the two ends of the tunnel may be different, as discussed in Section 3.7. This implies that the duty to filter external traffic extends to all routers participating in such tunnels.

16. Acknowledgments

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

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

17. Contributor Addresses

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

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

Email: randy@psg.com

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

Email: robert@raszuk.net

Eric C. Rosen

18. References

18.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <<https://www.rfc-editor.org/info/rfc2474>>.
- [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <<https://www.rfc-editor.org/info/rfc2784>>.
- [RFC2890] Dommety, G., "Key and Sequence Number Extensions to GRE", RFC 2890, DOI 10.17487/RFC2890, September 2000, <<https://www.rfc-editor.org/info/rfc2890>>.
- [RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001, <<https://www.rfc-editor.org/info/rfc3032>>.

- [RFC3270] Le Faucheur, F., Wu, L., Davie, B., Davari, S., Vaananen, P., Krishnan, R., Cheval, P., and J. Heinanen, "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services", RFC 3270, DOI 10.17487/RFC3270, May 2002, <<https://www.rfc-editor.org/info/rfc3270>>.
- [RFC3931] Lau, J., Ed., Townsley, M., Ed., and I. Goyret, Ed., "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, DOI 10.17487/RFC3931, March 2005, <<https://www.rfc-editor.org/info/rfc3931>>.
- [RFC4023] Worster, T., Rekhter, Y., and E. Rosen, Ed., "Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)", RFC 4023, DOI 10.17487/RFC4023, March 2005, <<https://www.rfc-editor.org/info/rfc4023>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <<https://www.rfc-editor.org/info/rfc4760>>.
- [RFC5129] Davie, B., Briscoe, B., and J. Tay, "Explicit Congestion Marking in MPLS", RFC 5129, DOI 10.17487/RFC5129, January 2008, <<https://www.rfc-editor.org/info/rfc5129>>.
- [RFC5462] Andersson, L. and R. Asati, "Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic Class" Field", RFC 5462, DOI 10.17487/RFC5462, February 2009, <<https://www.rfc-editor.org/info/rfc5462>>.
- [RFC6811] Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", RFC 6811, DOI 10.17487/RFC6811, January 2013, <<https://www.rfc-editor.org/info/rfc6811>>.
- [RFC6890] Cotton, M., Vegoda, L., Bonica, R., Ed., and B. Haberman, "Special-Purpose IP Address Registries", BCP 153, RFC 6890, DOI 10.17487/RFC6890, April 2013, <<https://www.rfc-editor.org/info/rfc6890>>.

- [RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <<https://www.rfc-editor.org/info/rfc7348>>.
- [RFC7606] Chen, E., Ed., Scudder, J., Ed., Mohapatra, P., and K. Patel, "Revised Error Handling for BGP UPDATE Messages", RFC 7606, DOI 10.17487/RFC7606, August 2015, <<https://www.rfc-editor.org/info/rfc7606>>.
- [RFC7637] Garg, P., Ed. and Y. Wang, Ed., "NVGRE: Network Virtualization Using Generic Routing Encapsulation", RFC 7637, DOI 10.17487/RFC7637, September 2015, <<https://www.rfc-editor.org/info/rfc7637>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8669] Previdi, S., Filsfils, C., Lindem, A., Ed., Sreekantiah, A., and H. Gredler, "Segment Routing Prefix Segment Identifier Extensions for BGP", RFC 8669, DOI 10.17487/RFC8669, December 2019, <<https://www.rfc-editor.org/info/rfc8669>>.

18.2. Informative References

- [Ethertypes] "IANA Ethertype Registry", <<http://www.iana.org/assignments/ieee-802-numbers/ieee-802-numbers.xhtml>>.
- [I-D.ietf-bess-evpn-inter-subnet-forwarding] Sajassi, A., Salam, S., Thoria, S., Drake, J., and J. Rabadan, "Integrated Routing and Bridging in EVPN", draft-ietf-bess-evpn-inter-subnet-forwarding-11 (work in progress), October 2020.
- [RFC4272] Murphy, S., "BGP Security Vulnerabilities Analysis", RFC 4272, DOI 10.17487/RFC4272, January 2006, <<https://www.rfc-editor.org/info/rfc4272>>.

- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <<https://www.rfc-editor.org/info/rfc4364>>.
- [RFC5512] Mohapatra, P. and E. Rosen, "The BGP Encapsulation Subsequent Address Family Identifier (SAFI) and the BGP Tunnel Encapsulation Attribute", RFC 5512, DOI 10.17487/RFC5512, April 2009, <<https://www.rfc-editor.org/info/rfc5512>>.
- [RFC5565] Wu, J., Cui, Y., Metz, C., and E. Rosen, "Softwire Mesh Framework", RFC 5565, DOI 10.17487/RFC5565, June 2009, <<https://www.rfc-editor.org/info/rfc5565>>.
- [RFC5566] Berger, L., White, R., and E. Rosen, "BGP IPsec Tunnel Encapsulation Attribute", RFC 5566, DOI 10.17487/RFC5566, June 2009, <<https://www.rfc-editor.org/info/rfc5566>>.
- [RFC5640] Filsfils, C., Mohapatra, P., and C. Pignataro, "Load-Balancing for Mesh Softwires", RFC 5640, DOI 10.17487/RFC5640, August 2009, <<https://www.rfc-editor.org/info/rfc5640>>.
- [RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, DOI 10.17487/RFC6514, February 2012, <<https://www.rfc-editor.org/info/rfc6514>>.
- [RFC7510] Xu, X., Sheth, N., Yong, L., Callon, R., and D. Black, "Encapsulating MPLS in UDP", RFC 7510, DOI 10.17487/RFC7510, April 2015, <<https://www.rfc-editor.org/info/rfc7510>>.
- [RFC8205] Lepinski, M., Ed. and K. Sriram, Ed., "BGPsec Protocol Specification", RFC 8205, DOI 10.17487/RFC8205, September 2017, <<https://www.rfc-editor.org/info/rfc8205>>.
- [RFC8277] Rosen, E., "Using BGP to Bind MPLS Labels to Address Prefixes", RFC 8277, DOI 10.17487/RFC8277, October 2017, <<https://www.rfc-editor.org/info/rfc8277>>.
- [RFC8365] Sajassi, A., Ed., Drake, J., Ed., Bitar, N., Shekhar, R., Uttaro, J., and W. Henderickx, "A Network Virtualization Overlay Solution Using Ethernet VPN (EVPN)", RFC 8365, DOI 10.17487/RFC8365, March 2018, <<https://www.rfc-editor.org/info/rfc8365>>.

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

Appendix A. Impact on RFC 8365

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

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

In particular, a review of Section 8.3.1 of RFC 8365 is called for, to consider whether the introduction of the Tunnel Encapsulation Attribute creates a need for any revisions to the split horizon procedures.

RFC 8365 also refers to a draft version of this specification in the final paragraph of section 5.1.3. That paragraph references Section 8.2.2.2 of the draft. In this version of the document the correct reference would be Section 9.2.2.2. There are no substantive differences between the section in the referenced draft, and that in this document.

Authors' Addresses

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

Email: keyur@arrcus.com

Gunter Van de Velde
Nokia
Copernicuslaan 50
Antwerpen 2018
Belgium

Email: gunter.van_de_velde@nokia.com

Srihari R. Sangli
Juniper Networks

Email: ssangli@juniper.net

John Scudder
Juniper Networks

Email: jgs@juniper.net

IDR
Internet-Draft
Updates: 4271, 4360, 7153 (if approved)
Intended status: Standards Track
Expires: September 4, 2018

Z. Li
China Mobile
J. Dong
Huawei Technologies
March 3, 2018

Carry congestion status in BGP community
draft-li-idr-congestion-status-extended-community-07

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

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.

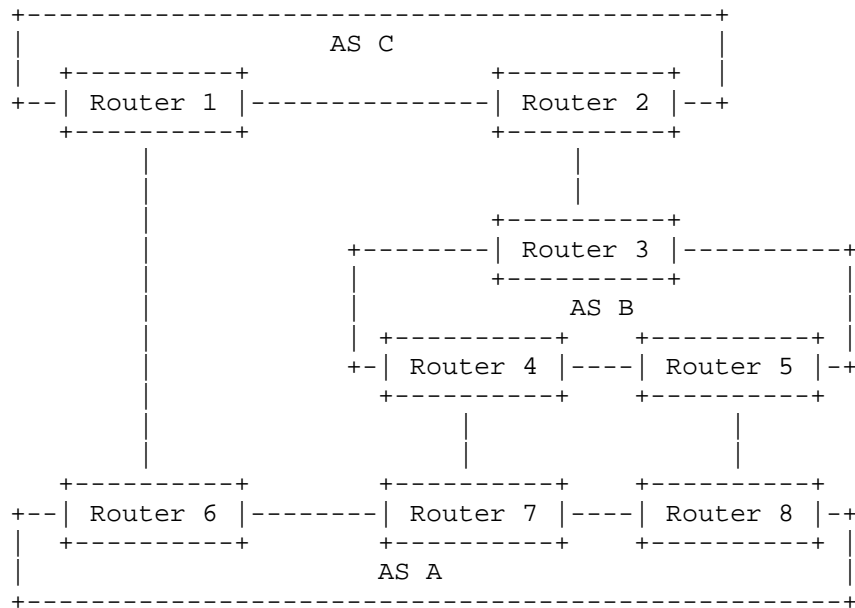
Table of Contents

1. Introduction	2
2. Requirements Language	4
3. Previous Work	4
4. Solution Alternative 1: Extended Community	4
5. Solution Alternative 2: Large Community	6
6. Solution Alternative 3: Community Container	6
7. Deployment Considerations	8
8. Security Considerations	9
9. IANA Considerations	9
10. Acknowledgments	9
11. References	9
11.1. Normative References	10
11.2. Informative References	10
Appendix A. Bandwidth Values	11
Authors' Addresses	12

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 mail 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 has 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.

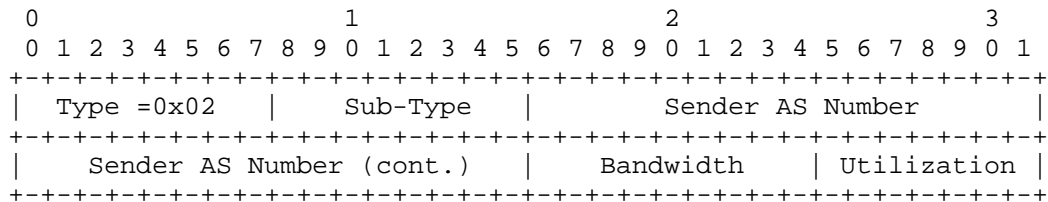


Figure 1: Congestion status extended community

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-octet 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.

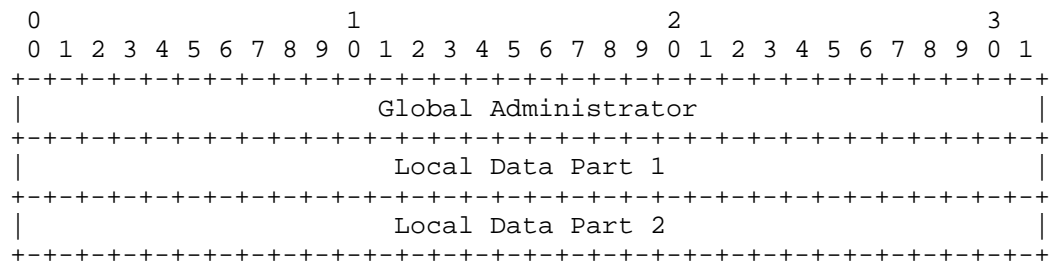


Figure 2

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.

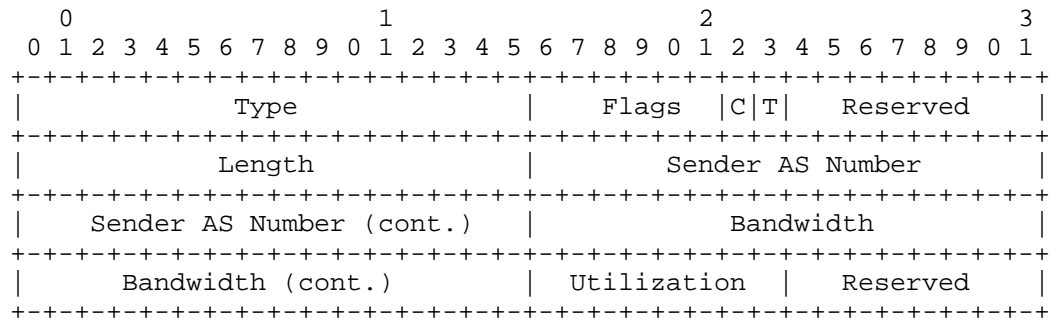


Figure 3

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-octet 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

o 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.

o 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.

o Other Consens

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,
<<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A
Border Gateway Protocol 4 (BGP-4)", RFC 4271,
DOI 10.17487/RFC4271, January 2006,
<<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC4360] Sangli, S., Tappan, D., and Y. Rekhter, "BGP Extended
Communities Attribute", RFC 4360, DOI 10.17487/RFC4360,
February 2006, <<https://www.rfc-editor.org/info/rfc4360>>.
- [RFC7153] Rosen, E. and Y. Rekhter, "IANA Registries for BGP
Extended Communities", RFC 7153, DOI 10.17487/RFC7153,
March 2014, <<https://www.rfc-editor.org/info/rfc7153>>.
- [RFC8092] Heitz, J., Ed., Snijders, J., Ed., Patel, K., Bagdonas,
I., and N. Hilliard, "BGP Large Communities Attribute",
RFC 8092, DOI 10.17487/RFC8092, February 2017,
<<https://www.rfc-editor.org/info/rfc8092>>.

11.2. Informative References

- [constrained-multiple-path]
Boucadair, M. and C. Jacquenet, "Constrained Multiple BGP
Paths", October 2010, <[https://www.ietf.org/archive/id/
draft-boucadair-idr-constrained-multiple-path-00.txt](https://www.ietf.org/archive/id/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.

[link-bandwidth-community]

Mohapatra, P. and R. Fernando, "BGP Link Bandwidth Extended Community", January 2013, <<https://www.ietf.org/archive/id/draft-ietf-idr-link-bandwidth-06.txt>>.

[RFC1997] Chandra, R., Traina, P., and T. Li, "BGP Communities Attribute", RFC 1997, DOI 10.17487/RFC1997, August 1996, <<https://www.rfc-editor.org/info/rfc1997>>.

[RFC2439] Villamizar, C., Chandra, R., and R. Govindan, "BGP Route Flap Damping", RFC 2439, DOI 10.17487/RFC2439, November 1998, <<https://www.rfc-editor.org/info/rfc2439>>.

[RFC3471] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description", RFC 3471, DOI 10.17487/RFC3471, January 2003, <<https://www.rfc-editor.org/info/rfc3471>>.

[RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, DOI 10.17487/RFC4456, April 2006, <<https://www.rfc-editor.org/info/rfc4456>>.

[RFC6793] Vohra, Q. and E. Chen, "BGP Support for Four-Octet Autonomous System (AS) Number Space", RFC 6793, DOI 10.17487/RFC6793, December 2012, <<https://www.rfc-editor.org/info/rfc6793>>.

[RFC7911] Walton, D., Retana, A., Chen, E., and J. Scudder, "Advertisement of Multiple Paths in BGP", RFC 7911, DOI 10.17487/RFC7911, July 2016, <<https://www.rfc-editor.org/info/rfc7911>>.

Appendix A. Bandwidth Values

Some typical bandwidth values encoded in 32-bit IEEE floating point format are enumerated below.

Link Type	Bit-rate (Mbps)	Bandwidth Value (Bytes/Sec) (32-bit IEEE Floating point)
-----	-----	-----
E1	2.048	0x487A0000
Ethernet	10.00	0x49989680
Fast Ethernet	100.00	0x4B3EBC20
OC-3/STM-1	155.52	0x4B9450C0
OC-12/STM-4	622.08	0x4C9450C0
GigE	1000.00	0x4CEE6B28
OC-48/STM-16	2488.32	0x4D9450C0
OC-192/STM-64	9953.28	0x4E9450C0
10GigE	10000.00	0x4E9502F9
OC-768/STM-256	39813.12	0x4F9450C0
100GigE	100000.00	0x503A43B7

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

IDR
Internet-Draft
Updates: 5575 (if approved)
Intended status: Standards Track
Expires: September 4, 2018

Z. Li
China Mobile
J. Dong
S. Zhuang
Huawei Technologies
March 3, 2018

Populate to FIB Action for FlowSpec
draft-li-idr-flowspec-populate-to-fib-02

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
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.

```

      40  41  42  43  44  45  46  47
+-----+-----+-----+-----+-----+
|           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:

- o F: Populate to FIB Action (bit 45, to be assigned 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

[RFC5575]) 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.

- o FlowSpec rules have higher priority than corresponding IGP and BGP routing entries.
- o When populating the FIB, the FlowSpec rules with F bit tagged are preferred than the corresponding IGP and BGP routing entries.
- o 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.
- o 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]

Hares, S., Loibl, C., Raszuk, R., McPherson, D., and M. Bacher, "Dissemination of Flow Specification Rules", draft-ietf-idr-rfc5575bis-06 (work in progress), October 2017.

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

[RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, DOI 10.17487/RFC5575, August 2009, <<https://www.rfc-editor.org/info/rfc5575>>.

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

INTERNET-DRAFT
Intended Status: <Standard Track>
Expires: December 30,2017

M. Sun
B.Pithawala
HUAWEI Technologies
F.Gao
Baidu Inc
June 28,2017

<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/lid-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).

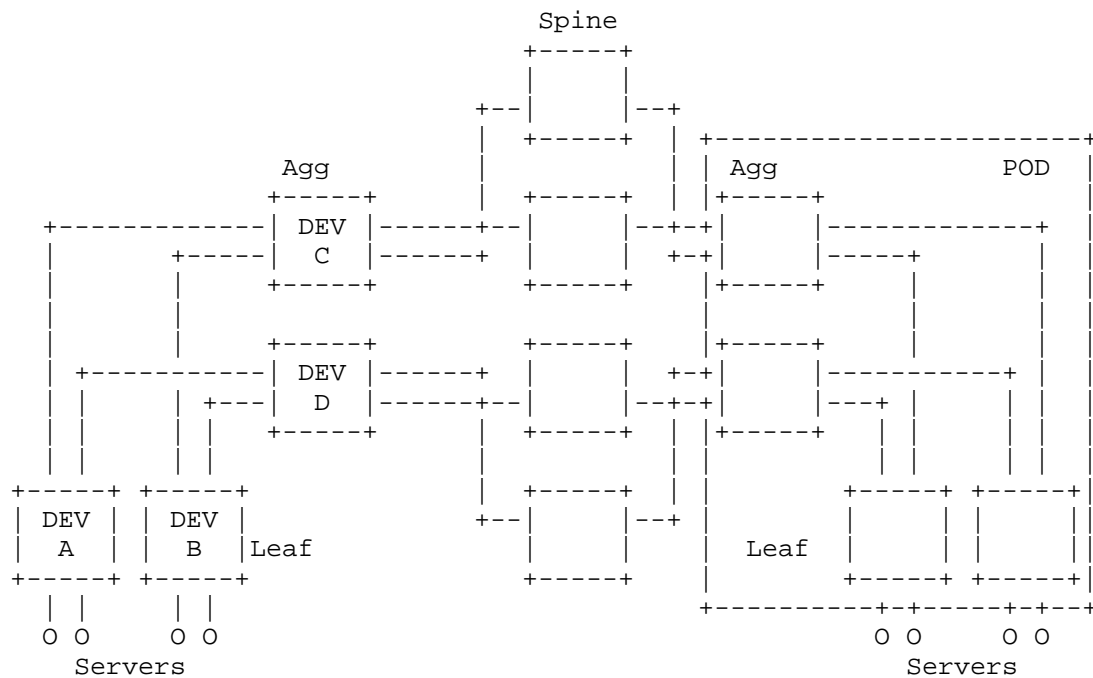


Figure 1 3-Layer Clos Topology

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.

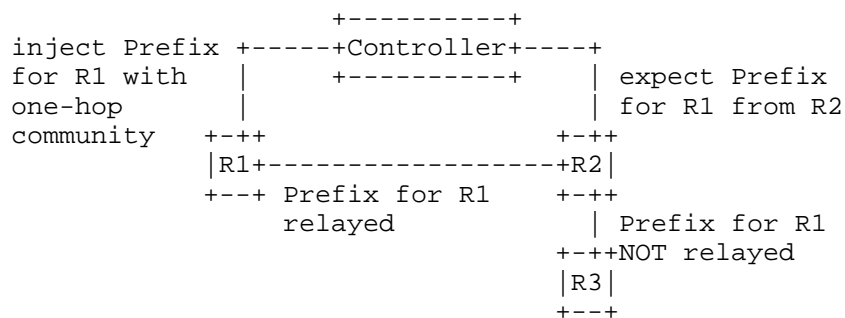


Figure 2 one kind of link detection method

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.

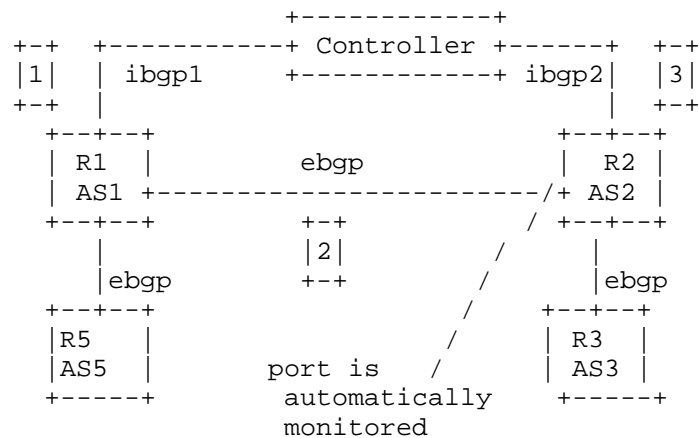


Figure 3 the principle of this solution

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 ebgp neighbor R2. R2 will publish this route message to controller through ibgp instead of its ebgp 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

- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010.
- [RFC7153] E. Rosen, Y. Rekhter, "IANA Registries for BGP Extended Communities", RFC 7153, March 2014.
- [RFC7938] P. Lapukhov, A. Premji, J. Mitchell, Ed., "Use of BGP for Routing in Large-Scale Data Centers", RFC 7938, August 2016.

4.2 Informative References

- [RFC3765] Huston, G., "NOPEER Community for Border Gateway Protocol (BGP) Route Scope Control", RFC 3765, April 2004.
- [RFC6286] E. Chen, J. Yuan, "Autonomous-System-Wide Unique BGP Identifier for BGP-4", RFC 6286, June 2011.
- [RFC6608] J. Dong, M. Chen, A. Suryanarayana, "Subcodes for BGP Finite State Machine Error", RFC 6608, May 2012.
- [RFC7606] E. Chen, Ed., J. Scudder, Ed., P. Mohapatra, K. Patel, "Revised Error Handling for BGP UPDATE Messages", RFC 7606, August 2015.
- [RFC7705] W. George, S. Amante, "Autonomous System Migration Mechanisms and Their Effects on the BGP AS_PATH Attribute", RFC 7705, November 2015.
- [RFC7752] H. Gredler, Ed., J. Medved, S. Previdi, A. Farrel, S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, March 2016.

Authors' Addresses

Marcus Sun
HUAWEI TECHNOLOGIES CO.,LTD
12 E. Mozhou Rd.Nanjing,Jiangsu
China

EEmail: marcus.sun@huawei.com

Burjiz Pithawala
HUAWEI TECHNOLOGIES CO.,LTD
2330 Central Expressway, Santa Clara, CA 95050
US

EEmail: burjiz.pithawalal@huawei.com

Feng Gao
BAIDU Inc.
10 shangdi shijie Haidian, Beijing

Email:gaofeng04@baidu.com

Marcus, et al.

Expires December 30,2017

[Page 8]

