Workgroup: Network Working Group
Internet-Draft: draft-ietf-idr-rpd-15
Published: 25 January 2022
Intended Status: Standards Track
Expires: 29 July 2022
Authors: Z. Li L. Ou
Huawei China Telcom Co., Ltd.
Y. Luo S. Lu G. Mishra
China Telcom Co., Ltd. Tencent Verizon Inc.
H. Chen S. Zhuang H. Wang
Futurewei Huawei Huawei

BGP Extensions for Routing Policy Distribution (RPD)

Abstract

It is hard to adjust traffic and optimize traffic paths in a traditional IP network from time to time through manual configurations. It is desirable to have a mechanism for setting up routing policies, which adjusts traffic and optimizes traffic paths automatically. This document describes BGP Extensions for Routing Policy Distribution (BGP RPD) to support this.

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

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction
2. Terminology
3. Problem Statement
  3.1. Inbound Traffic Control
  3.2. Outbound Traffic Control
4. Protocol Extensions
  4.1. Using a New AFI and SAFI
  4.2. BGP Wide Community and Atoms
    4.2.1. RouteAttr atom Sub-TLV
    4.2.2. Sub-TLVs of the Parameters TLV
  4.3. Capability Negotiation
5. Operations
  5.1. Application Scenario
  5.2. About Failure
6. Contributors
7. Security Considerations
8. Acknowledgements
9. IANA Considerations
  9.1. Existing Assignments
  9.2. Registered IANA Wide Communities
  9.3. RouteAttr Atom Type
  9.4. Route Attributes Sub-sub-TLV Registry
  9.5. Attribute Change Sub-TLV Registry
10. References
  10.1. Normative References
  10.2. Informative References
Authors' Addresses

1. Introduction

It is difficult to optimize traffic paths in a traditional IP network because of the following:

* Complex. Traffic can only be adjusted device by device. The configurations on all the routers that the traffic traverses need to be changed or added. There are already lots of policies configured on the routers in an operational network. There are different types of policies, which include security, management and control policies. These policies are relatively stable. However, the policies for adjusting traffic are dynamic. Whenever the traffic through a route is not expected, the policies to adjust the traffic for that route are configured on the related routers. It is complex to dynamically add or change the policies to the existing policies on the special routers to adjust the traffic. Some people would like to separate the stable route policies from the dynamic ones even though they have configuration automation systems (including YANG models).

* Difficult maintenance. The routing policies used to adjust network traffic are dynamic, posing difficulties to subsequent maintenance. High maintenance skills are required.

* Slow. Adding or changing some route policies on some routers through a configuration automation system for adjusting some traffic to avoid congestions may be slow.

It is desirable to have an automatic mechanism for setting up routing policies, which can simplify routing policy configuration and be fast. This document describes extensions to BGP for Routing Policy Distribution to resolve these issues.

2. Terminology

The following terminology is used in this document.

* ACL: Access Control List

* BGP: Border Gateway Protocol [RFC4271]

* FS: Flow Specification

* NLRI: Network Layer Reachability Information [RFC4271]

* PBR: Policy-Based Routing

* RPD: Routing Policy Distribution

* VPN: Virtual Private Network

3. Problem Statement

Providers have the requirement to adjust their business traffic routing policies from time to time because of the following:

* Business development or network failure introduces link congestion and overload.

* Business changes or network additions produce unused resources such as idle links.

* Network transmission quality is decreased as the result of delay, loss and they need to adjust traffic to other paths.

* To control OPEX and CPEX, they may prefer the transit provider with lower price.

3.1. Inbound Traffic Control

In Figure 1, for the reasons above, the provider P of AS100 may wish the inbound traffic from AS200 to enter AS100 through link L3 instead of the others. Since P doesn't have any administrative control over AS200, there is no way for P to directly modify the route selection criteria inside AS200.

```
               Traffic from PE1 to Prefix1
          ----------------------------------->

+-----------------+            +-------------------------+
|     +---------+ |        L1  | +----+      +----------+|
|     |Speaker1 | +------------+ |IGW1|      |policy    ||
|     +---------+ |**      L2**| +----+      |controller||
|                 |  **    **  |             +----------+|
| +---+           |    ****    |                         |
| |PE1|           |    ****    |                         |
| +---+           |  **    **  |                         |
|     +---------+ |**      L3**| +----+                  |
|     |Speaker2 | +------------+ |IGW2|      AS100       |
|     +---------+ |        L4  | +----+                  |
|                 |            |                         |
|    AS200        |            |                         |
|                 |            |  ...                    |
|                 |            |                         |
|     +---------+ |            | +----+      +-------+   |
|     |Speakern | |            | |IGWn|      |Prefix1|   |
|     +---------+ |            | +----+      +-------+   |
+-----------------+            +-------------------------+

            Prefix1 advertised from AS100 to AS200
          <----------------------------------------
```

Figure 1: Inbound Traffic Control case

3.2. Outbound Traffic Control

In Figure 2, the provider P of AS100 prefers link L3 for the traffic to the destination Prefix2 among multiple exits and links to AS200. This preference can be dynamic and might change frequently because of the reasons above. So, provider P expects an efficient and convenient solution.

```
               Traffic from PE2 to Prefix2
          ----------------------------------->
+-------------------------+            +-----------------+
|+----------+      +----+ |L1          | +---------+     |
||policy    |      |IGW1| +------------+ |Speaker1 |     |
||controller|      +----+ |**        **| +---------+     |
|+----------+             |L2**    **  |        +-------+|
|                         |    ****    |        |Prefix2||
|                         |    ****    |        +-------+|
|                         |L3**    **  |                 |
|      AS100       +----+ |**        **| +---------+     |
|                  |IGW2| +------------+ |Speaker2 |     |
|                  +----+ |L4          | +---------+     |
|                         |            |                 |
|+---+                    |            |    AS200        |
||PE2|              ...   |            |                 |
|+---+                    |            |                 |
|                  +----+ |            | +---------+     |
|                  |IGWn| |            | |Speakern |     |
|                  +----+ |            | +---------+     |
+-------------------------+            +-----------------+

            Prefix2 advertised from AS200 to AS100
          <----------------------------------------
```

Figure 2: Outbound Traffic Control case

4. Protocol Extensions

This document specifies a solution using a new AFI and SAFI with the BGP Wide Community for encoding a routing policy.

4.1. Using a New AFI and SAFI

A new AFI and SAFI are defined: the Routing Policy AFI whose codepoint 16398 has been assigned by IANA, and SAFI whose codepoint 75 has been assigned by IANA.

The AFI and SAFI pair uses a new NLRI, which is defined as follows:

```
 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
+-+-+-+-+-+-+-+-+
| NLRI Length   |
+-+-+-+-+-+-+-+-+
| Policy Type   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Distinguisher (4 octets)                                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Peer IP (4/16 octets)                                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: AFI and SAFI with new NLRI

Where:

NLRI Length: 1 octet represents the length of NLRI. If the Length is anything other than 9 or 21, the NLRI is corrupt and the enclosing UPDATE message MUST be ignored.

Policy Type: 1 octet indicates the type of a policy. 1 is for Export policy. 2 is for Import policy. If the Policy Type is any other value, the NLRI is corrupt and the enclosing UPDATE message MUST be ignored.

Distinguisher: 4 octet unsigned integer that uniquely identifies the content/policy. It is used to sort/order the polices from the lower to higher distinguisher. They are applied in ascending order. A policy with a lower/smaller distinguisher is applied before the policies with a higher/larger distinguisher.

Peer IP: 4/16 octet value indicates IPv4/IPv6 peers. Its default value is 0, which indicates that when receiving a BGP UPDATE message with the NLRI, a BGP speaker will apply the policy in the message to all its IPv4/IPv6 peers.

Under RPD AFI/SAFI, the RPD routes are stored and ordered according to the keys (Policy type, Distinguisher, Peer IP). Under IPv4/IPv6 Unicast AFI/SAFI, there are IPv4/IPv6 unicast routes learned and various static policies configured. In addition, there are dynamic RPD policies from the RPD AFI/SAFI when RPD is enabled.

Before advertising an IPv4/IPv6 Unicast AFI/SAFI route, the configured policies are applied to it first, and then the RPD Export policies are applied.

The NLRI containing the Routing Policy is carried in MP_REACH_NLRI and MP_UNREACH_NLRI path attributes in a BGP UPDATE message, which MUST also contain the BGP mandatory attributes and MAY contain some BGP optional attributes.

When receiving a BGP UPDATE message with routing policy, a BGP speaker processes it as follows:

* If the peer IP in the NLRI is 0, then apply the routing policy to all the remote peers of this BGP speaker.

* If the peer IP in the NLRI is non-zero, then the IP address indicates a remote peer of this BGP speaker and the routing policy will be applied to it.

The content of the Routing Policy is encoded in a BGP Wide Community.

4.2. BGP Wide Community and Atoms

The BGP wide community attribute is defined in [I-D.ietf-idr-wide-bgp-communities]. This document specifies how two wide communities associate the routing policy NLRI to Routing Policy NLRI (section 4.1) to distribute routing policy to BGP peers. The wide communities which define routing policy are:

* MATCH AND SET ATTR (TBD1)

* MATCH and NOT ADVERTISE (TBD2)

These wide communities are passed in the BGP wide community container in the wide community attribute. These communities support three of the optional TLVs: Target TLV, Exclude Target TLV, and Parameter TLV. The value of each of these TLVs comprises a series of Atoms, each of which is a TLV (or sub-TLV).

A new wide community Atom is defined for BGP Wide Community Target(s) TLV (RouteAttr), and two new Atoms are defined for BGP Wide Community Parameter(s) TLV. For your reference, the format of the TLV is illustrated 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      |              Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Value (variable)                      ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Format of Wide Community Atom TLV

4.2.1. RouteAttr atom Sub-TLV

A RouteAttr Atom sub-TLV (or RouteAttr sub-TLV for short) is defined and may be included in a Target TLV. It has the following format.

```
 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 (TBD3)  |        Length (variable)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          sub-sub-TLVs                          ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: Format of RouteAttr Atom sub-TLV

The Type for RouteAttr atom is TBD3. In RouteAttr sub-TLV, four sub-sub-TLVs are defined: IPv4 Prefix, IPv6 Prefix, AS-Path, and Community sub-sub-TLV.

An IP prefix sub-sub-TLV gives matching criteria on IPv4 prefixes. Its format is illustrated 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  1      |          Length (N x 8)        |M-Type | Flags |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          IPv4 Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Mask      |     GeMask    |     LeMask    |M-Type | Flags |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~       . . .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          IPv4 Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Mask      |     GeMask    |     LeMask    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Format of IPv4 Prefix sub-sub-TLV

Type: 1 for IPv4 Prefix.

Length: N x 8, where N is the number of tuples <M-Type, Flags, IPv4 Address, Mask, GeMask, LeMask>. If Length is not a multiple of 8, the Atom is corrupt and the enclosing UPDATE message MUST be ignored.

M-Type: 4-bit field specifying match type. The following four values are defined. IPaddress is the IP address in the sub-sub-TLV while IProute is the IP route being matched.

M-Type = 0: Exact match with the Mask length IP address prefix. GeMask and LeMask MUST be sent as zero and ignored on receipt.

M-Type = 1: Matches if the Mask number of prefix bits exactly match between IPaddress and IProute and the actual prefix length of IProute is greater than or equal to GeMask. LeMask MUST be sent as zero and ignored on receipt.

M-Type = 2: Matches if the Mask number of prefix bits exactly match between IPaddress and IProute and the actual prefix length of IProute is less than or equal to LeMask. GeMask MUST be sent as zero and ignored on receipt.

M-Type = 3: Matches if the Mask number of prefix bits exactly match between IPaddress and IProute and the actual prefix length of IProute is less than or equal to LeMask and greater than or equal to GeMask.

Flags: 4 bits. No flags are currently defined. They MUST be sent as zero and ignored on receipt.

IPv4 Address: 4 octets for an IPv4 address.

Mask: 1 octet for the IP address prefix length that needs to exactly match between the IP address in the sub-sub-TLV and the route.

GeMask: 1 octet for route prefix length match range's lower bound, MUST not be less than Mask or be 0.

LeMask: 1 octet for route prefix length match range's upper bound, MUST be greater than Mask or be 0.

For example, tuple <M-Type=0, Flags=0, IPv4 Address = 1.1.0.0, Mask = 22, GeMask = 0, LeMask = 0> represents an exact IP prefix match for 1.1.0.0/22.

<M-Type=1, Flags=0, IPv4 Address = 16.1.0.0, Mask = 24, GeMask = 24, LeMask = 0> represents match IP prefix 16.1.0.0/24 greater-equal 24 (i.e., route matches if route's first Mask=24 bits match 16.1.0 and 24 =< route's prefix length =< 32).

<M-Type=2, Flags=0, IPv4 Address = 17.1.0.0, Mask = 24, GeMask = 0, LeMask = 26> represents match IP prefix 17.1.0.0/24 less-equal 26 (i.e., route matches if route's first Mask=24 bits match 17.1.0 and 24 =< route's prefix length <= 26).

<M-Type=3, Flags=0, IPv4 Address = 18.1.0.0, Mask = 24, GeMask = 24, LeMask = 30> represents match IP prefix 18.1.0.0/24 greater-equal 24 and less-equal 30 (i.e., route matches if route's first Mask=24 bits match 18.1.0 and 24 =< route's prefix length <= 30).

Similarly, an IPv6 Prefix sub-sub-TLV represents match criteria on IPv6 prefixes. Its format is illustrated 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  4     |         Length (N x 20)       |M-Type | Flags |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     IPv6 Address (16 octets)                  ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Mask      |     GeMask    |     LeMask    |M-Type | Flags |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~       . . .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      IPv6 Address (16 octets                  ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Mask      |     GeMask    |     LeMask    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: Format of IPv6 Prefix sub-sub-TLV

An AS-Path sub-sub-TLV represents a match criteria in a regular expression string. Its format is illustrated 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  2      |      Length (Variable)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    AS-Path Regex String                       |
:                                                               :
|                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: Format of AS Path sub-sub-TLV

Type: 2 for AS-Path.

Length:
Variable, maximum is 1024.

AS-Path Regex String: AS-Path regular expression string.

A community sub-sub-TLV represents a list of communities to be matched all. Its format is illustrated 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  3      |        Length (N x 4 + 1)       |    Flags    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Community 1 Value                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                              . . .                            ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Community N Value                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: Format of Community sub-sub-TLV

Type: 3 for Community.

Length: N x 4 + 1, where N is the number of communities. If Length is not a multiple of 4 plus 1, the Atom is corrupt and the enclosing UPDATE MUST be ignored.

Flags: 1 octet. No flags are currently defined. These bits MUST be sent as zero and ignored on receipt.

4.2.2. Sub-TLVs of the Parameters TLV

This document introduces 2 community values:

MATCH AND SET ATTR (TBD1): If the IPv4/IPv6 unicast routes to a remote peer match the specific conditions defined in the routing policy extracted from the RPD route, then the attributes of the IPv4/IPv6 unicast routes will be modified when sending to the remote peer per the actions defined in the RPD route.

MATCH AND NOT ADVERTISE (TBD2): If the IPv4/IPv6 unicast routes to a remote peer match the specific conditions defined in the routing policy extracted from the RPD route, then the IPv4/IPv6 unicast routes will not be advertised to the remote peer.

For the Parameter(s) TLV, two action sub-TLVs are defined: MED change sub-TLV and AS-Path change sub-TLV. When the community in the container is MATCH AND SET ATTR, the Parameter(s) TLV can include these sub-TLVs. When the community is MATCH AND NOT ADVERTISE, the Parameter(s) TLV's value is empty.

A MED change sub-TLV indicates an action to change the MED. Its format is illustrated 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  1      |          Length (5)           |      OP       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Value                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: Format of MED Change sub-TLV

Type: 1 for MED Change.

Length: 5. If Length is any other value, the sub-TLV is corrupt and the enclosing UPDATE MUST be ignored.

OP: 1 octet. Three are defined:

OP = 0: assign the Value to the existing MED.

OP = 1: add the Value to the existing MED. If the sum is greater than the maximum value for MED, assign the maximum value to MED.

OP = 2: subtract the Value from the existing MED. If the existing MED minus the Value is less than 0, assign 0 to MED.

If OP is any other value, the sub-TLV is ignored. 4 octets.

Value:

An AS-Path change sub-TLV indicates an action to change the AS-Path. Its format is illustrated 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  2      |        Length (n x 5)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             AS1                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Count1     |
+-+-+-+-+-+-+-+-+
~       . . .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             ASn                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Countn     |
+-+-+-+-+-+-+-+-+
```

Figure 11: Format of AS-Path Change sub-TLV

Type: 2 for AS-Path Change.

Length: n x 5. If Length is not a multiple of 5, the sub-TLV is corrupt and the enclosing UPDATE MUST be ignored.

ASi: 4 octet. An AS number.

Counti: 1 octet. ASi repeats Counti times.

The sequence of AS numbers are added to the existing AS Path.

4.3. Capability Negotiation

It is necessary to negotiate the capability to support BGP Extensions for Routing Policy Distribution (RPD). The BGP RPD Capability is a new BGP capability [RFC5492]. The Capability Code for this capability is 72 assigned by the IANA. The Capability Length field of this capability is variable. The Capability Value field consists of one or more of the following tuples:

```
+--------------------------------------------------+
|  Address Family Identifier (2 octets)            |
+--------------------------------------------------+
|  Subsequent Address Family Identifier (1 octet)  |
+--------------------------------------------------+
|  Send/Receive (1 octet)                          |
+--------------------------------------------------+
```

Figure 12: BGP RPD Capability

The meaning and use of the fields are as follows:

Address Family Identifier (AFI): This field is the same as the one used in [RFC4760].

Subsequent Address Family Identifier (SAFI): This field is the same as the one used in [RFC4760].

Send/Receive: This field indicates whether the sender is (a) willing to receive Routing Policies from its peer (value 1), (b) would like to send Routing Policies to its peer (value 2), or (c) both (value 3) for the <AFI, SAFI>. If Send/Receive is any other value, that tuple is ignored but any other tuples present are still used.

5. Operations

This section presents a typical application scenario and some details about handling a related failure.

5.1. Application Scenario

Figure 13 illustrates a typical scenario, where RPD is used by a controller with a Route Reflector (RR) to adjust traffic dynamically.

```
+--------------+
|  Controller  |
+-------+------+
         \
          \ RPD
        .--\._.+--+                       ___...__
    __(     \       '.---...             (         )
   /      RR o -------- A o) ---------- (o X   AS2  )
  (o E       |\             )     _____//(___   ___)
   (         | \_______ B o) ____/     /     '''
    (o F      \           )       ____/
     (         \_____ C o) ______/         ___...__
      '    AS1        _)  \_____          (         )
       '---._.-.     )          \_______ (o Y   AS3  )
                '---'                     (___   ___)
                                              '''
```

Figure 13: Controller with RR Adjusts Traffic

The controller connects the RR through a BGP session. There is a BGP session between the RR and each of routers A, B and C in AS1, which is shown in the figure. Other sessions in AS1 are not shown in the figure.

There is router X in AS2. There is a BGP session between X and each of routers A, B and C in AS1.

There is router Y in AS3. There is a BGP session between Y and router C in AS1.

The controller sends a RPD route to the RR. After receiving the RPD route from the controller, the RR reflects the RPD route to routers A, B and C. After receiving the RPD route from the RR, routers A, B and C extract the routing policy from the RPD route. If the peer IP in the NLRI of the RPD route is 0, then apply the routing policy to all the remote peers of routers A, B and C. If the peer IP in the NLRI of the RPD route is non-zero, then the IP address indicates a remote peer of routers A, B and C and such routing policy is applied to the specific remote peer. The IPv4/IPv6 unicast routes towards router X in AS2 and router Y in AS3 will be adjusted based on the routing policy sent by the controller via a RPD route.

The controller uses the RT extend community to notify a router whether to receive a RPD policy. For example, if there is not any adjustment on router B, the controller sends RPD routes with the RTs for A and C. B will not receive the routes.

The process of adjusting traffic in a network is a close loop. The loop starts from the controller with some traffic expectations on a set of routes. The controller obtains the information about traffic flows for the related routes. It analyzes the traffic and checks whether the current traffic flows meet the expectations. If the expectations are not met, the controller adjusts the traffic. And then the loop goes to the starter of the loop (The controller obtains the information about traffic ...).

5.2. About Failure

This section describes some details about handling a failure related to a RPD route being applied.

A RPD route is not a configuration. When it is sent to a router from a controller, no ack is needed from the router. The existing BGP mechanisms are re-used for delivering a RPD route. After the route is delivered to a router, it will be successful. This is guaranteed by the BGP protocols.

If there is a failure for the router to install the route locally, this failure is a bug of the router. The bug needs to be fixed.

For the errors mentioned in [RFC7606], they are handled according to [RFC7606]. These errors are bugs, which need to be resolved.

When the controller fails while a RPD route is being applied such as on the way to the router, some existing mechanisms such BGP Graceful Restart (GR) [RFC4724] and BGP Long-lived Graceful Restart (LLGR) can be used to let the router keep the routes from the controller for some time.

With support of "Long-lived Graceful Restart Capability" [I-D.ietf-idr-long-lived-gr], the routes can be retained for a longer time after the controller fails.

After the controller recovers from its failure, the router will have all the routes (including the RPD route being applied) from the controller.

In the worst case, the controller fails and the RPD routes for adjusting the traffic are withdrawn. The traffic adjusted/redirected may take its old path. This should be acceptable.

6. Contributors

The following people have substantially contributed to the definition of the BGP-FS RPD and to the editing of this document:

Peng Zhou
Huawei
Email: Jewpon.zhou@huawei.com

7. Security Considerations

Protocol extensions defined in this document do not affect BGP security other than as discussed in the Security Considerations section of [RFC8955].

8. Acknowledgements

The authors would like to thank Acee Lindem, Jeff Haas, Jie Dong, Lucy Yong, Qiandeng Liang, Zhenqiang Li, Robert Raszuk, Donald Eastlake, Ketan Talaulikar, and Jakob Heitz for their comments to this work.

9. IANA Considerations

9.1. Existing Assignments

IANA has assigned an AFI of value 16398 from the registry "Address Family Numbers" for Routing Policy.

IANA has assigned a SAFI of value 75 from the registry "Subsequent Address Family Identifiers (SAFI) Parameters" for Routing Policy.

IANA has assigned a Code Point of value 72 from the registry "Capability Codes" for Routing Policy Distribution.

9.2. Registered IANA Wide Communities

IANA Should assign from the Registered Wide Community Values" the following values:

Community Value	Description	Reference
TBD1	MATCH AND SET ATTR	This document
TBD2	MATCH AND NOT ADVISE	This document

9.3. RouteAttr Atom Type

IANA is requested to assign a code-point from the registry "BGP Community Container Atom Types" as follows:

Atom Code Point	Description	Reference
TBD3 (48 suggested)	RouteAttr Atom	This document

9.4. Route Attributes Sub-sub-TLV Registry

IANA is requested to create a registry called "Route Attributes Sub-sub-TLV" under RouteAttr Atom Sub-TLV. The allocation policy of this registry is "First Come First Served (FCFS)".

The initial code points are as follows:

Code Point	Description	Reference
0	Reserved	
1	IPv4 Prefix Sub-sub-TLV	This document
2	AS-Path Sub-sub-TLV	This document
3	Community Sub-sub-TLV	This document
4	IPv6 Prefix Sub-sub-TLV	This document
5 - 255	Available	

9.5. Attribute Change Sub-TLV Registry

IANA is requested to create a registry called "Attribute Change Sub-TLV" under Parameter(s) TLV. The allocation policy of this registry is "First Come First Served (FCFS)".

Initial code points are as follows:

Code Point	Description	Reference
0	Reserved	
1	MED Change Sub-TLV	This document
2	AS-Path Change Sub-TLV	This document
3 - 255	Available	

10. References

10.1. 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", Work in Progress, Internet-Draft, draft-ietf-idr-wide-bgp-communities-06, 10 January 2022, <https://www.ietf.org/archive/id/draft-ietf-idr-wide-bgp-communities-06.txt>.

[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>.

[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>.

[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>.

[RFC8955] Loibl, C., Hares, S., Raszuk, R., McPherson, D., and M. Bacher, "Dissemination of Flow Specification Rules", RFC 8955, DOI 10.17487/RFC8955, December 2020, <https://www.rfc-editor.org/info/rfc8955>.

10.2. Informative References

[I-D.ietf-idr-long-lived-gr] Uttaro, J., Chen, E., Decraene, B., and J. G. Scudder, "Support for Long-lived BGP Graceful Restart", Work in Progress, Internet-Draft, draft-ietf-idr-long-lived-gr-00, 5 September 2019, <https://www.ietf.org/archive/id/draft-ietf-idr-long-lived-gr-00.txt>.

[I-D.ietf-idr-registered-wide-bgp-communities] Raszuk, R. and J. Haas, "Registered Wide BGP Community Values", Work in Progress, Internet-Draft, draft-ietf-idr-registered-wide-bgp-communities-02, 31 May 2016, <https://www.ietf.org/archive/id/draft-ietf-idr-registered-wide-bgp-communities-02.txt>.

[RFC4724] Sangli, S., Chen, E., Fernando, R., Scudder, J., and Y. Rekhter, "Graceful Restart Mechanism for BGP", RFC 4724, DOI 10.17487/RFC4724, January 2007, <https://www.rfc-editor.org/info/rfc4724>.

[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>.

Authors' Addresses

Zhenbin Li
Huawei
Huawei Bld., No.156 Beiqing Rd.
Beijing
100095
China

Email: lizhenbin@huawei.com

Liang Ou
China Telcom Co., Ltd.
109 West Zhongshan Ave,Tianhe District

Guangzhou
510630
China

Email: ouliang@chinatelecom.cn

Yujia Luo
China Telcom Co., Ltd.
109 West Zhongshan Ave,Tianhe District
Guangzhou
510630
China

Email: luoyuj@sdu.edu.cn

Sujian Lu
Tencent
Tengyun Building,Tower A ,No. 397 Tianlin Road
Shanghai
Xuhui District, 200233
China

Email: jasonlu@tencent.com

Gyan S. Mishra
Verizon Inc.
13101 Columbia Pike
Silver Spring, MD 20904
United States of America

Phone: 301 502-1347
Email: gyan.s.mishra@verizon.com

Huaimo Chen
Futurewei
Boston, MA,
United States of America

Email: Huaimo.chen@futurewei.com

Shunwan Zhuang
Huawei
Huawei Bld., No.156 Beiqing Rd.
Beijing
100095
China

Email: zhuangshunwan@huawei.com

Haibo Wang

Huawei
Huawei Bld., No.156 Beiqing Rd.
Beijing
100095
China

Email: rainsword.wang@huawei.com