Group Co-operative Route Filtering capability for BGP-4
draft-muley-hares-idr-orf-order-01

Status of this Memo

Abstract

Currently BGP-4 is capable of carrying multiple (Outbound Route Filters) ORFs entries for a given "AFI/SAFI". Each ORF provides a filter that a route whose NLRI matches AFI/SAFI, must pass through to be transmitted in the BGP Update message. Efficient processing of ORF filters may require ordering of individual ORFs in certain sequence and grouping of ORFs that should be applied together. The grouping functionality also provides the support for logical OR operation between the grouped ORFs.

This group ORF provides an ORF type that specifies that ordering and grouping. The route set that passes set of ORFs running in a "Group ORF" will pass the same ORFs sent in normal ORFs.

Table of Contents

---

1. Definitions

1.1. Conventions used in this document

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 (Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," March 1997.) [RFC2119].

---

2. Introduction

Currently it is not uncommon for a BGP speaker to receive set of ORFs from an ORF capable BGP peer. Each ORF provides a filter that a route must pass through to be transmitted in the BGP update message. Today's operational procedure for cooperative route filtering provides the AFI/

SAFI as the only context. ORF by definition in its current form have logical OR within ORF entries and logical AND within the ORF types. Efficient processing and expression of filters for ORF may require ordering of ORFs filters and ORF entries in certain sequence. Efficient processing entails both BGP processing and quick processing of the ORF generated from User policies.

This document defines GROUP ORF, a new BGP-4 ORF type that allows BGP to send to its peer a group of set of ordered ORF filters. Group ORF provides a context for filtering rules that need to be interpreted together further within that context AFI/SAFI. The grouping functionality also provides the support for logical OR operation between grouped ORFs.

Today ORF entries of same ORF type have logical OR functionality only, which limits the flexibility of defining an efficient ORF with less grammar description. Group ORF capability will help in defining the relational meaning within the ORF entries as well.

One example of this is the use of ORFs to efficiently handle complex ORF policies applied per VPN per peer, in BGP/MPLS VPN deployment as defined in: RFC4364 (Cisco and Juniper, "BGP/MPLS VPN," February 2006.) [RFC4364].

---

3. Past Contriburos

The authors want to acknowledge the contributions of Luyuan Fang and Nabil Bitar.

---

4. Group ORF Type

The group ORF type allows a set of ORF entries of different ORF types and ORF entries within the type, to be grouped, and ordered in the BGP ROUTE-REFRESH message RFC2918 (Cisco, "Route Refresh Capability for BGP-4," Septebmer 2000.) [RFC2918]. The Group ORF provides group cooperative route filtering.

Conceptually a Group ORF consist of multiple different types of ORF entries as defined in [BGP-ORF] (Sangli, S. and Cisco, ""Cooperative Router Filtering for BGP-4"," July 2005.) [bgp-orf], [ASPATH-ORF] (Hares, S. and K. Patel, ""Aspath Based Outbound Route Filter for BGP-4"," August 2007.) [aspath-orf] and [prefix-ORF] (Sangli, S. and Cisco, ""Address Prefix Based Outbound Route Filter for BGP-4"," March 2005.) [prefix-orf]. Each such ORF type, will have ordered ORF entries, with an operation of logical OR or logical AND defined with each other.

5. Group ORF Encoding

The value of the ORF-Type for Group ORF is [TBD].
Conceptually the Route Refresh message carrying Group ORF can be viewed
as below.

```
+--------------------------------------------------+
| Address Family Identifier (2 octets)             |
+--------------------------------------------------+
| Reserved (1 octet)                               |
+--------------------------------------------------+
| Subsequent Address Family Identifier (1 octet)   |
+--------------------------------------------------+
| When-to-refresh (1 octet)                        |
+--------------------------------------------------+
|  ORF Type (1 octet) (Group)                      |
+--------------------------------------------------+
|  ORF Group Flags                                 |
+--------------------------------------------------+
| Length of ORFs (2 octets)                        |
+--------------------------------------------------+
| First ORF entry (variable) (Group)               |
+--------------------------------------------------+
| Second ORF entry (variable)  (Group)             |
+--------------------------------------------------+
    .........
+--------------------------------------------------+
| N-th ORF entry (variable)                        |
+--------------------------------------------------+
```

ORF byte layout

Each Group ORF entry will be encoded as follows.

```
+---------------------------------------------------+
| ORF Group id (1 octet)                            |
+---------------------------------------------------+
| ORF Type (1 octet)     [Group]                    |
+---------------------------------------------------+
| ORF Action (1 octet)                              |
+---------------------------------------------------+
| ORF FLags (1 octet)                               |
| Length of ORFs (2 octets)                         |
+---------------------------------------------------+
| First ORF entry (variable)                        |
+---------------------------------------------------+
| Second ORF entry (variable)                       |
+---------------------------------------------------+
    .........
+---------------------------------------------------+
| N-th ORF entry (variable)                         |
+---------------------------------------------------+
| ORF Type (1 octet)                                |
+---------------------------------------------------+
| Length of ORFs (2 octets)                         |
+---------------------------------------------------+
| First ORF entry (variable)                        |
+---------------------------------------------------+
| Second ORF entry (variable)                       |
+---------------------------------------------------+
    .........
```

Group ORF byte layout

ORF Action:
Byte that indicates operation of "AND and OR" function within the full
Group ORF.
Value 000 Use AND or OR specified in Group ORF. OR between GROUP IDs.
Value 001 Always AND between attributes in Flag Byte
Always ORed between same Route-map's different sequences.
ORF FLAG:
Value 0000 0001 Route-Map
Value 0000 0002 Access Control List
Value 1000 0000 Ignore ORF Action BIT

---

6. ORF Entry Encoding

As specified in the [BGP-ORF] (Sangli, S. and Cisco, ""Cooperative Router Filtering for BGP-4"," July 2005.) [bgp-orf] the ORF entry definition has common part and type specific part. The encoding of common part of the ORF entries on Group ORF capability negotiation as defined below.

```
+-------------------------------+
|   Action (2 bit)              |
+-------------------------------+
|  Match (1 bit)                |
+-------------------------------+
|  AND/OR (1 bit)               |
+-------------------------------+
|   Reserved (4 bits)           |
+-------------------------------+
|   Type specific part (variable) |
+-------------------------------+
```

Group ORF Entry Byte

The new AND/OR is a one bit field additional to the definition specified in [BGP-ORF] (Sangli, S. and Cisco, ""Cooperative Router Filtering for BGP-4"," July 2005.) [bgp-orf]. The value of this field is 0 for expression of logical OR and 1 for logical AND with the next ORF entry. The ORF entries grammar expression for logical OR followed by logical AND ORF entries will be for the over all the group of such and-ed ORF entries. This field in the last entry of the ORF type will be ignored.

7. Group Cooperative route filtering capability

A BGP speaker that is willing to receive Group ORF entries from its peer, or a BGP speaker that would like to send ORF entries to its peer advertises this to the peer by using the Cooperative Route Filtering Capability, as described below.
The Group ORF Capability is a new BGP capability [BGP-CAP] (Cisco and Cisco, "Capabilities Advertisement with BGP-4," November 2002.) [bgp-cap] defined as follows

Capability code: X [IANA consideration]

        Capability length: variable

        Capability value: one or more of the entries as defined for ORF
        entries in the [BGP-ORF] (Sangli, S. and Cisco, ""Cooperative Router
        Filtering for BGP-4"," July 2005.) [bgp-orf].

    The use of ORF entry in Group ORF will depend upon the send/receive
    value of the ORF type in capability negotiation

---

8. Operation

In addition to operational procedures defined in [BGP-ORF] (Sangli, S.
and Cisco, ""Cooperative Router Filtering for BGP-4"," July 2005.)
[bgp-orf] several additional operational rules needs to be followed.
The Group ORF Group-ID allows a tag for a group of ORFs. If multiple
instances of a Group ORF with the same Group-ID exist within a Route
Refresh Message, the additional group information is appended to the
set of previously received ORFs with the same Group-ID. The complete
list of ORFs within the Group will be included in the "and-ing" process
for that Group.
When processing multiple Group ORFs into a filter, the Group ORFs will
be applied in ascending order of the Group ID.
When the BGP speaker receives multiple ORFs within a Group ORF entry,
the order of the ORFs is preserved and applied as per first ORF entry
match rule.
For each Group ORF, a BGP speaker will pass the set of candidate NLRIs
(routes) through each ORF that is a member of the Group, and-ing the
results (and-ing of PERMIT and DENY results in DENY). If a Group ORF
would result in a PERMIT for a given NLRI, it is advertised to the peer
and it is removed from the list of candidate NLRIs. If a Group ORF
would result in a DENY for a given NLRI, it is not advertised to the
peer and it is removed from the list of candidate NLRIs. The remaining
list of candidate NLRIs are then filtered through the next Group ORF.
This process is repeated until the candidate NRLIs have been filtered
through all Group ORFs. The remaining candidate NLRIs are then filtered
through any non-Group ORFs that might exist. While processing the ORF
entries in the ORF type the result of ORF entry match with AND/OR bit
set will be and-ed with the subsequent ORF entry match. The ORF entry
with AND/OR bit is set to 0 will OR the match result with subsequent
entry match result. Rest of the procedure for treatment of NLRI remains
same as described in the [BGP-ORF] (Sangli, S. and Cisco, ""Cooperative
Router Filtering for BGP-4"," July 2005.) [bgp-orf].
To remove a group ORF then all the ORF entries in the Group ORF will
have the action component as REMOVE.

The Group ORF MUST always have higher preference than non-Group ORFs, and will be processed first.
Note: Group ORF are independent of ORF preference and configuration to occur without concern for order of transmission. Individual ORF type preference within the Group ORF will occur based on configuration policy.

---

9. Deployment Scenarios

Following are few deployment examples where Group ORF helps in filtering and offering flexibility in ORF expression.
Scenario 1
An example of group ORF could be in the IPVPN case is where multiple BGP communities [BGP-COM} (Cisco, Cisco, and Cisco, "BGP/MPLS VPN," August 1996.) [bgp-com] and/or extended communities need to be considered together for a given VPN in the filtering rules although all IPVPN routes belong to the same AFI/SAFI.
Consider the case where there are two VPNs, red and yellow, have routes belonging to sites tagged by community attributes that express city locations. The Red VPN wants or needs to get routes from certain site locations but not others. Similarly, the yellow VPN wants or needs to get sites from certain location not others. For the purpose of ease in manageability, it would be desired to affiliate the community attribute value with the city location only. That is the red VPN and yellow VPN routes will carry the same city community value when these routes reside in same city.
If cooperative route filtering is used without group ORF, it will not be possible to express the ORFs when the the two VPNs have different rules whereby different city routes need to be imported to another city for the two VPNs. This is because of the ORing operation within the same ORF type and ANDing operation across ORF types.
To illustrate further consider the following example where routers in the red VPN in city 4 needs to get the routes from city 1 only and not from city 2 and 3. On the other hand, the routers in the yellow VPN in city 4 needs to get routes from city 2 but not city 3 and city 1. In current procedures for cooperative route filtering without group ORF, the only way to express that for ORF from the routers hosting VPN sites in city 4 is:

    AFI/SAFI = IPVPN

    Extended ORF Type = Target Extended Community

        Permit Red

        Permit Yellow

```
ORF Type = Community

    Permit City1

    Permit City2
```

Based on this ORF routes tagged with Red and City2 will be permitted
and routes tagged with Yellow and City1 will be permitted although
these are not deired routes and will need to be filtered out by a local
policy on the routers in City4.
If group ORF was available, the ORF policy can be more flexibly
expressed in the following manner:

```
AFI/SAFI = IPVPN

Group 1 (implicitly Red VPN)

    Extended ORF Type = Target Extended Community

        Permit Red

    ORF Type = Community

        Permit City1

Group 2 (implicitly Yellow VPN)

    Extended ORF Type = Target Extended Community

        Permit Yellow

    ORF Type = Community

        Permit City2
```

Thus Group 1 provides a context for the red VPN and Group 2 provides a
context for the yellow VPN. Only routes that are tagged with Red target
extended community attributes and City1 community or routes tagged with
Yellow target community attributes and City2 community will be
permitted to be advertised to the BGP client that sent this ORF,
achieving the desired result.
It may be argued that in this simple case, the problem could be
addressed with existing ORF capabilities by simply tagging the Red and
City 1 routes with a new extended community tag (Red_City1) and the
routes with the Yellow and City2 tags with a new target extended
community tag (Yellow_City2) and including these in the ORF. However,
this can get complicated as the number of combinations of BGP extended
communities [RFC4360 (Cisco, Cisco, and Juniper, "BGP Extended
Communities Attribute," July 2006.)](#) [RFC4360] and communities
increases. Having group ORF capability offers the needed flexibility.

It is also true that the same result can be achieved by defining local
policies on the routers but the tradeoff as in any ORF case is between
the work done at a client and that done at the route reflector when
used in topology.

Scenario 2

An example Group ORF in the global routing context can be where a BGP
speaker wants to learn certain more specific NLRIs only if they
traverse certain AS path. Otherwise routes which are less specific than
/25. A Group ORF will look like as defined below where X,Y,Z are the
more specific NLRIs to be learnt only if it learn it from AS1.

```
AFI/SAFI = IPV4

Group 1

   ORF Type = Prefix

      Permit X

      Permit Y

      Permit Z

   ORF Type = ASpath

      Permit AS1

Group 2

   ORF Type = Prefix

      Deny prefix( */25) or longer

Group 3

   ORF Type = Prefix

      Permit prefix(*)
```

Although its possible to achieve the above mentioned application by
making use of route policies, use of Group ORF helps in simplifying the
configuration.

---

10. Security Considerations

This extension to BGP does not change the underlying security issues.

---

11. IANA Considerations

IANA will need to assign the value of the ORF-Type for Group ORF.

12. References

[RFC2119]	Bradner, S., "[Key words for use in RFCs to Indicate Requirement Levels](#)," March 1997.
[RFC4364]	Cisco and Juniper, "[BGP/MPLS VPN](#)," February 2006.
[bgp-cap]	Cisco and Cisco, "[Capabilities Advertisement with BGP-4](#)," November 2002.
[bgp-com]	Cisco, Cisco, and Cisco, "[BGP/MPLS VPN](#)," August 1996.
[RFC4360]	Cisco, Cisco, and Juniper, "[BGP Extended Communities Attribute](#)," July 2006.
[RFC2918]	Cisco, "[Route Refresh Capability for BGP-4](#)," Septebmer 2000.
[bgp-orf]	Sangli, S. and Cisco, "["Cooperative Router Filtering for BGP-4"](#)," July 2005.
[prefix-orf]	Sangli, S. and Cisco, "["Address Prefix Based Outbound Route Filter for BGP-4"](#)," March 2005.
[aspath-orf]	Hares, S. and K. Patel, "["Aspath Based Outbound Route Filter for BGP-4"](#)," August 2007.

Authors' Addresses

	Susan Hares
	Green Hills Software
	825 Victors Way
	Ann Arbor, MI 48108
Phone:	+1-734-222-1610
Email:	shares@ghs.com
	Praveen Muley
	Alcatel
	701, E. Middle Field Rd
	Mountain View, CA 94043
Phone:	+1-650-623-2622
Email:	Praveen.Muley@alcatel.com
	Keyur Patel
	Cisco Systems
Phone:	
Email:	keyupate@cisco.com

	Benson Schliesser
	Saavis Communications
	1 Savvis Parkway
	Town and Country, MO 63017
Phone:	+1-314-628-7036
Email:	bensons@savvis.net

Full Copyright Statement

Intellectual Property