IDR Working Group W. Wang
Internet-Draft A. Wang
Intended status: Experimental China Telecom
Expires: 3 December 2026 H. Wang
Huawei Technologies
G. Mishra
Verizon Inc.
J. Dong
Huawei Technologies
1 June 2026

VPN Prefix Outbound Route Filter (VPN Prefix ORF) for BGP-4
draft-ietf-idr-vpn-prefix-orf-44

Abstract

This document defines an experimental specification for a new type of Outbound Route Filter (ORF), known as the Virtual Private Network (VPN) Prefix ORF. The VPN Prefix ORF mechanism is applicable when VPN routes from different Virtual Routing and Forwarding (VRF) instances are exchanged through a single shared Border Gateway Protocol (BGP) session. The purpose of the VPN Prefix ORF mechanism is to control the overload of VPN routes based on Route Distinguisher (RD), Route Target (RT) and other necessary routing information. This mechanism is applicable to intra-domain scenarios.

Table of Contents

1. Introduction

The BGP Maximum Prefix feature [RFC4486] is often used at the network boundary to control the number of prefixes injected into the network. However, in scenarios where VPN routes from multiple VRFs are advertised over a shared BGP session, there is a lack of appropriate methods to control route flooding within one VRF. This flooding can overwhelm the processing of VPN routes in other VRFs, consequently degrading their performance (e.g., causing route drops, processing delays, and abnormal customer services). Therefore, it is desirable to control excessive VPN route advertisements individually for each VRF within such a shared BGP session.

Several solutions can be used to alleviate this problem:

* Route Target Constraint (RTC) as defined in [RFC4684]

* Address Prefix ORF as defined in [RFC5292]

* Covering Prefixes Outbound Route Filter (CP-ORF) mechanism as defined in [RFC7543]

* Provider Edge (PE) - Customer Edge (CE) edge peer Maximum Prefix as mentioned in Section 7.6.3.2 in [RFC9182]

* Configuring the Maximum Prefix for each VRF on edge nodes

However, each existing solution has its own limitation as described in Section 3.

This document defines an experimental specification for a new type of Outbound Route Filter (ORF), called the VPN Prefix ORF; refer to Appendix A for more details on the experimental aspects. When the number of VPN routes in a VRF exceeds the prefix limit, the router identifies the VPN prefix (Route Distinguisher (RD), Route Target (RT), source PE, etc.) of the overload VPN routes (VPN routes that exceed the maximum number of storable VPN routes of the corresponding VRF on the receiver and thus cannot be imported.) and sends a VPN Prefix ORF message to the BGP peer that announced these overload VPN routes. Upon receiving a VPN Prefix ORF entry, the BGP speaker filters and withdraws any overload VPN routes that were previously announced to its peer.

The purpose of this mechanism is to control the overload VPN routes, avoid route churn effects and resource exhaustion when a VRF overloads. The VPN Prefix ORF mechanism is applicable when VPN routes from different VRFs are exchanged via a shared BGP session.

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. Terminology

The following terms are used in This document:

* AFI: Address Family Identifier, defined in [RFC4760]

* ASBR: Autonomous System Border Router

* BGP: Border Gateway Protocol, defined in [RFC4271]

* EVPN: BGP/MPLS Ethernet VPN, defined in [RFC7432]

* MPLS: Multi-Protocol Label Switching

* ORF: Outbound Route Filter, defined in [RFC5291]

* Quota: A threshold to limit the number of VPN routes under specific granularities (such as <PE>)

* RD: Route Distinguisher, defined in [RFC4364]

* RIB: Routing Information Base

* RR: Route Reflector, provides a simple solution to the problem of Internal Border Gateway Protocol (iBGP) full mesh connection in large-scale iBGP implementation [RFC4456]

* RT: Route Target, defined in [RFC4364]

* SAFI: Subsequent Address Family Identifier, defined in [RFC4760]

* VPN: Virtual Private Network, defined in [RFC4364]

* VRF: VPN Routing and Forwarding, defined in [RFC4364]

3. Existing Solutions

3.1. Route Target Constraint (RTC)

RTC [RFC4684] can only filter the VPN routes from any uninterested VRFs, if the route overload comes from an interested VRF, the RTC mechanism can’t filter them.

3.2. Address Prefix ORF

Using Address Prefix ORF [RFC5292] to filter VPN routes requires a pre-configuration, but it is impossible to know in advance which prefix may exceed the predefined threshold.

3.3. CP-ORF Mechanism

[RFC7543] defines the Covering Prefixes ORF (CP-ORF). A BGP speaker sends a CP-ORF to a peer in order to pull routes that cover a specified host address. A prefix covers a host address if it can be used to forward traffic towards that host address.

CP-ORF is applicable in Virtual Hub-and-Spoke [RFC7024] VPN and also BGP/MPLS Ethernet VPN (EVPN) [RFC7432] networks, but its primary function is to retrieve interested VPN prefixes and it cannot be used to filter overload of VPN prefixes dynamically.

3.4. PE-CE edge peer Maximum Prefix

The BGP Maximum-Prefix feature controls the number of prefixes received from a neighbor. Configuring it on every PE-CE link can prevent VPN route overload. However, relying solely on sender-side protection is insufficient. If the sender has not configured Maximum Prefix, the VPN Prefix ORF mechanism can still prevent VPN route overload.

3.5. Configuring the Maximum Prefix for each VRF on edge nodes

When a VRF overloads, some implementations may stop importing routes. Any additional VPN routes are held in the Routing Information Base (RIB). However, PEs still need to parse the incoming BGP messages, which consumes CPU cycles and further burdens the overloaded PE.

The VPN Prefix ORF mechanism intends to improve upon this by enabling the overloaded PE to signal the VPN routes matching the overload criteria back to the sender. The sender can then suppress these routes at the source, reducing wasted processing and preserving resources for non-overloaded VRFs.

If the sender does not support or does not honor the VPN Prefix ORF signal, the overloaded PE still needs to parse the incoming BGP messages, and the behavior falls back to existing local protection mechanisms, such as local VRF prefix-limit handling. This limitation is part of the experimental assessment described in Section 7.5.

4. VPN Prefix ORF Encoding

This section describes the encoding of VPN Prefix ORF entries. The VPN Prefix ORF entries are carried in the BGP ROUTE-REFRESH message as defined in [RFC5291]. A BGP ROUTE-REFRESH message can carry one or more ORF entries. VPN Prefix ORF entries consist exclusively of Match fields with the value DENY. VPN Prefix ORF entries are evaluated in sequential order based on the Sequence field, defined below. When no VPN Prefix ORF entries in a non-empty VPN Prefix ORF match the route that is passed through the ORF, the Match criterion for the route is considered PERMIT. The format of a ROUTE-REFRESH message carrying VPN Prefix ORF entries is as follow:

* AFI (2 octets). The AFI MUST be set to IPv4, IPv6, or Layer 2 VPN (L2VPN).

* SAFI (1 octet). If the AFI is set to IPv4 or IPv6, the SAFI can be set to MCAST-VPN, or MPLS-Labeled VPN. If the AFI is set to L2VPN, the SAFI can be set to BGP EVPN, VPLS, or MCAST-VPLS. The combination relationships between SAFI and AFI are presented in Table 1:

Table 1 Allowed SAFI and AFI combinations

AFI	SAFI	Document
IPv4(1)/ IPv6(2)	MCAST-VPN(5)	[RFC6514]
	MPLS-labeled VPN address (128)	[RFC4364][RFC8277][RFC9252]
L2VPN(25)	BGP EVPNs(70)	[RFC7432]
	VPLS(65)	[RFC4761][RFC6074]
	MCAST-VPLS(8)	[RFC7117]

* When-to-refresh (1 octet): the value MUST be IMMEDIATE or DEFER.

* ORF Type (1 octet): The type of VPN Prefix ORF is 66.

* Length of ORF entries (2 octets)

A VPN Prefix ORF entry contains a common part and type-specific part. The encoding of the common part is shown in Figure 1.

```
+------------------------------------------+
|                                          |
|              Action (2 bits)             |
|                                          |
+------------------------------------------+
|                                          |
|               Match (1 bit)              |
|                                          |
+------------------------------------------+
|                                          |
|        Overload VPN routes process       |
|               method (1 bit)             |
|                                          |
+------------------------------------------+
|                                          |
|             Reserved (4 bits)            |
|                                          |
+------------------------------------------+
```

Figure 1: VPN Prefix ORF type-common part encoding

* Action (2 bits): the value is ADD, REMOVE or REMOVE-ALL as described in [RFC5291]. If the received ORF entry contains an unrecognized value(0x11), such an ORF entry MUST be removed.

* Match (1 bit): the value is PERMIT(0) or DENY(1) as described in [RFC5291]. For the purpose of this document, only the DENY value is permitted. This bit MUST be set to DENY(1). If Match = 0, the entry MUST be discarded and a warning MUST be sent to the operator.

* Overload VPN routes process method (1 bit): if the value is set to 0, it means the receiver of such message MUST withdraw all previously advertised overload VPN routes that match the ORF's type-specific part. If the value is set to 1, it means the sender of the VPN Prefix ORF message will refuse to accept VPN routes matching the overload criteria and that the receiver of the VPN Prefix ORF message MUST NOT announce VPN routes matching the overload criteria. The default value is 0. Note well, this bit is specific to the ORF Type introduced by this document.

* Reserved (4 bits): These bits are set to zero and ignored by the receiver.

VPN Prefix ORF also contains a type-specific part. The encoding of the type-specific part is shown in Figure 2.

```
+-----------------------------------------+
|                                         |
|           Sequence (4 octets)           |
|                                         |
+-----------------------------------------+
|                                         |
|           Length (2 octets)             |
|                                         |
+-----------------------------------------+
|                                         |
|     Route Distinguisher (8 octets)      |
|                                         |
+-----------------------------------------+
|                                         |
|        Optional TLVs (variable)         |
|                                         |
+-----------------------------------------+
```

Figure 2: VPN Prefix ORF type-specific encoding

* Sequence: Identifies the order in which the VPN Prefix ORF is generated and evaluated. It uniquely identifies a VPN Prefix ORF entry, along with the AFI/SAFI, ORF-Type, and Route Distinguisher. The Sequence SHOULD be non-contiguous to facilitate the insertion of new rules at a later stage, which means the receiver should not assume the sequence numbers are contiguous; a gap in sequence numbers (e.g., receiving sequence N then N+k, k>1) is valid.

* Length: Specifies the length of this VPN Prefix ORF entry.

* Route Distinguisher: Distinguishes different user routes. The VPN Prefix ORF filters the VPN routes it intends to send based on Route Distinguisher. If the RD is set to 0, it indicates all VPN prefixes.

* Optional Type-Length-Values (TLVs): Carries potential additional information to provide extensibility for the VPN Prefix ORF mechanism. Its format is shown in Figure 3. If one or more TLV(s) are unrecognized, the entire VPN Prefix ORF entry MUST be discarded.

```
 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 3 The format of optional TLV(s)

Note that if the Action component of an ORF entry specifies REMOVE-ALL, the ORF entry does not include the type-specific part.

When the BGP ROUTE-REFRESH message carries VPN Prefix ORF entries, it MUST be set as follows:

* The ORF-Type MUST be set to 66 (VPN Prefix ORF).

* The purpose of VPN Prefix ORF is to block unwanted VPN prefixes; therefore, the "Action" of a valid entry MUST be set to "DENY". VPN routes that do not match any corresponding VPN Prefix ORF entries MUST be imported into the corresponding VRF.

According to [RFC5291], if any field in a VPN Prefix ORF entry in the message contains an unrecognized value, the entire specified ORF previously received is removed.

A BGP speaker that is willing to receive ORF entries from its peer, or a BGP speaker that would like to send ORF entries to its peer, advertises this capability by using the Outbound Route Filtering Capability defined in [RFC5291].

4.1. Source PE TLV (including 3 types)

The Source PE TLV is defined to identify the originator of the VPN routes. The sender of the VPN Prefix ORF MUST check for the existence of the Source PE Extended Community (SPE EC; see section 6) on the VPN route being matched. If the SPE EC exists, the sender MUST include its value in the Source PE TLV. Otherwise, the value of Source PE TLV MUST be set to the Next Hop address.

The Source PE TLV MUST appear at most once within an individual ORF entry. If an ORF entry contains multiple Source PE TLVs, the entire ORF entry MUST be ignored.

The source PE TLV supports the following types:

* IPv4 Source PE TLV: Type = 1, Length = 4 octets, value = Next Hop address in IPv4 format.

* IPv6 Source PE TLV: Type = 2, Length = 16 octets, value = Next Hop address in IPv6 format (global IPv6 address only).

* Source PE identifier TLV: Type = 3, Length = 4 octets, value = the value of ORIGINATOR_ID in the Source PE Extended Community.

4.2. Route Target TLV

The Route Target TLV is defined to identify the RT of the overloaded VPN routes. Route Target TLV MUST be exist in VPN Prefix ORF entry. The RT and RD can be used together to filter VPN routes if the source VRF contains multiple RTs, and the VPN routes with different RTs MAY be assigned to different VRFs on the receiver.

If this TLV contains only one RT but multiple RTs are configured on the VPN route, the VPN prefix ORF receiver MUST check whether the RT included in this TLV exists among the configured RTs. If so, it MUST filter out the VPN route.

The Route Target TLV is defined as:

Type = 5, Length = 8*n octets (where n is the number of RTs associated with the overloaded VPN routes), value = the RT value(s) of the overload VPN routes. If multiple RTs are included, an exact match is required.

4.3. Route Type TLV

This TLV applies to all VPN routes containing a route type field, to distinguish between different types of VPN routes and enable more granular control.

The encoding of the Route Type TLV is as follow:

Type = 6, Length = 1 octet, value = the value of Route Type field of the overload routes.

5. The general procedures of the VPN Prefix ORF mechanism

5.1. Process of VPN Prefix ORF mechanism on sender

The operation of the VPN Prefix ORF mechanism on each sender is independent. Each sender makes a local judgment to determine whether it needs to send a VPN Prefix ORF message to its upstream peer. Operators can configure the algorithms according to their specific circumstances.

This section specifies procedures for a receiving BGP speaker to process VPN route information received from a BGP speaker that sends VPN routes. The VPN route information includes VPN routes and associated Route Distinguishers (RDs). The receiving BGP speaker determines the newly received VPN routes and evaluates whether installation of those routes would cause the configured VPN route limit for the associated VRF to be exceeded.

If the configured VPN route limit for a VRF is exceeded, the receiving BGP speaker SHOULD send a VPN Prefix ORF message to the BGP speaker that sends VPN routes to request that transmission of the identified VPN routes cease.

Before originating a VPN Prefix ORF message, the receiving BGP speaker MUST compare the Route Targets (RTs) associated with the affected VPN routes against the import RTs configured on other VRFs of the device. If any RT associated with a VPN route is also imported by another VRF, it MUST NOT originate a VPN Prefix ORF message for that route. The procedures described in this section apply to iBGP peers within the same Autonomous System (AS). VPN route identification is based on the RD. VPN Prefix ORF information is carried using ORF entries within a ROUTE-REFRESH message.

The receiving BGP speaker includes the following information in the VPN Prefix ORF message:

* ORF entries carried within a ROUTE-REFRESH message.

* An Action field in each ORF entry indicating that the BGP speaker requests installation of the specified outbound route filter.

* A Match field in each ORF entry indicating that VPN route updates matching the specified ORF entry are to be denied.

* An RD value identifying the affected VPN routes, encoded in the type-specific portion of the ORF entry. If multiple VRFs on a PE import VPN routes associated with the same RD, and only a subset of those VRFs exceed their configured route limits, the PE MUST NOT originate a VPN Prefix ORF entry for that RD.

When the VPN Prefix ORF mechanism is triggered, the VPN Prefix ORF sender MUST send alarm information to network operators.

The procedures for senders of VPN Prefix ORF entries are described below:

```
S01. When VPN routes with a specific RD are installed in
     multiple VRFs, for each VRF v {
S02.     If (the total number of received prefixes + the number
         of prefixes already in VRF v exceeds its configured
         prefix limit) {
S03.         RT_set = the set of Route Targets that were to be
             imported into VRF v.
S04.         overload_RD_source_pairs = all <RD, Source PE>
             tuples from the newly received routes that were
             sent for incorporation into VRF v.

             // Check if any RT in RT_set is also imported by
                another VRF that has NOT exceeded its limit
S05.         conflict_exists = FALSE;
S06.         For each RT r in RT_set {
S07.             For each other VRF u on the VPN Prefix ORF sender {
S08.                 If (r is in the import RT list of VRF u) {
S09.                     conflict_exists = TRUE;
S09a                     break;
S10.                 }
S11.             }
S12.         }

S13.         If (conflict_exists == TRUE) {
S14.             // Cannot send ORF: would block routes needed
                    by non-overloaded VRFs
S15.             Send warning message to the operator.
                 break;
S16.         }

S17.         // Safe to send ORF entries
S18.         For each <RD_x, PE_y> in overload_RD_source_pairs {
S19.             Collect all RTs carried by routes with RD=RD_x
                 from source PE_y that are imported into VRF v.

S20.             Construct a VPN Prefix ORF entry with:
S21.                 Action = ADD,
S22.                 Match = DENY,
S23.                 Overload VPN routes process method = 0,
S24.                 Sequence = Generate unique Sequence number,
S25.                 Route Distinguisher = RD_x,
S26.                 Optional TLVs include:
S27.                     Source PE TLV = PE_y,
S28.                     Route Target TLV = RT_list.

S29.             Send a BGP ROUTE-REFRESH message containing this
                 ORF entry to the upstream BGP peer (e.g., RR).
S30.             Send an alarm message to the operator indicating
```

```
                 VRF v overload and ORF transmission.
S31.         }
S32.     } Else {
S33.         // No overload in this VRF; no ORF triggered
S34.         Continue normal route processing.
S35.     }
S36. }
```

5.1.1. Intra-domain Scenarios and Solutions

For intra-AS VPN deployment, there are two scenarios:

* unique RD (per VPN, per PE).

* the same RD (per VPN, same on all PEs)

Detailed descriptions about the above solutions are provided in Appendix B.

5.2. Protocol process of VPN Prefix ORF mechanism on receiver

The VPN Prefix ORF is primarily used to block unwanted BGP updates. When the receiver receives a VPN Prefix ORF entry, it MUST check first whether the "Match" bit is "DENY" or not.

If the "Match" bit is "PERMIT", the entry MUST be discarded and a warning MUST be sent to the operator.

The default entry for the VPN Prefix ORF type is "Permit All", which means that all routes that do not match the existing entries in the VPN Prefix ORF table shall be advertised. The following procedures will only be evaluated when the "Match" bit is "DENY".

The receiver of VPN Prefix ORF entries (which may be an RR, ASBR, or PE) performs the following actions upon receiving a VPN Prefix ORF entry from its BGP peer:

```
S01. The receiver checks the combination of <AFI/SAFI, ORF-Type,
     Sequence, Route Distinguisher> in the received VPN Prefix
     ORF entry.
S03. If (Action == ADD) {
S04     If (entry exists) {
            Replace existing entry.
        }
S05     else {
              Add new entry.
        }
     }
 S06 else if (Action == REMOVE) {
 S07      If (entry exists) {
                             Remove entry.
          }
     }
 S08 else if (Action == REMOVE-ALL) {
         Remove all matching VPN Prefix ORF entries.
     }
 S09 else {
          Handle as malformed or unsupported Action.
     }
```

The filtering conditions for stored VPN Prefix ORF entries include the RD and RT of the overloaded VPN Prefix Route.

If the SPE EC is not attached to the BGP Update message for the VPN prefixes, the receiver MUST use the NEXT_HOP or ORIGINATOR_ID attribute as the originator of the VPN prefix to match against the VPN Prefix ORF entry.

After installing the filter entries for outbound VPN prefixes, the receiver performs the following actions before sending VPN routes:

```
S01. The receiver checks if there are matching filtering conditions
     in the ORF-Policy table for the VPN routes.
S02. If (no matching filtering conditions exist) {
S03.     The receiver sends the VPN routes.
S04. } else {
S05.     If (the "Overload VPN routes process method" bit is set
         to 0) {
S06.         The receiver withdraws all the VPN routes identified
             by RD, RT, and any relevant information in the optional
             TLVs within the entry, and stops sending the
             corresponding VPN routes to the sender of the VPN
             Prefix ORF entry.
S07.     } else {
S08.         The receiver stops sending the newly learnt matched VPN
             routes according to the value of RD, RT, and any relevant
             information in optional TLVs within the entry to the
             sender of the VPN Prefix ORF entry.

S09. }
}
```

The procedure above can be used for route refresh processing after receiving an ORF update and the usual VPN route propagation. A change to the ORF prefixes triggers a rescan of the relevant routing information, followed by a route refresh. In contrast, regular individual VPN route updates are only subject to matching against the existing ORF rules.

The route-refresh procedure as specified in [RFC5291] is modified in the presence of VPN Prefix ORF Type entry with O-bit set to 1. The receiver is required to keep track of routes matching the ORF entries with O-bit set to 1 that have been already sent to the peer before those ORF entries were received and continue to advertise them even if denied by those ORF entries during both route-refresh processing and subsequent updates received for those routes.

6. Source PE Extended Community

Next Hop does not always identify the source as seen in the following scenarios:

* a PE MAY have multiple addresses, so that its BGP peer MAY receive several different next hop addresses from the same source.

* In an Option B inter-domain scenario, the ASBR will change the Next Hop.

ORIGINATOR_ID is a non-transitive attribute generated by an RR to identify the source, but ORIGINATOR_ID cannot be advertised outside the local AS. To address these scenarios, this section defines a new Extended Community: Source PE Extended Community (SPE EC), which is designed to transmit the identifier of the source PE. The value of the SPE EC can be set by the source PE, RR, or Autonomous System Boundary Router (ASBR). Once set and attached to a BGP UPDATE message, its value MUST NOT be altered along the advertisement path.

The format of SPE EC is shown as Figure 4.

```
 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 (TBD)        |          ORIGINATOR_ID        :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:     ORIGINATOR_ID (cont.)     |            Reserved           :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4 The format of SPE EC

Where:

* Type(16 bit): Specifies the type value assigned by IANA, now it is TBD.

* ORIGINATOR_ID(32 bit): Specifies the identifier of the source PE.

* Reserved(16 bit): The sender MUST set the Reserved field to 0 and a received MUST ignore the Reserved field.

An RR/ASBR SHOULD perform the following actions:

* Check for the existence of the SPE EC. If it exists, the RR/ASBR MUST NOT change it.

* If the SPE EC does not exist, check for the existence of the ORIGINATOR_ID. If it exists, put it into the SPE EC.

* If the ORIGINATOR_ID does not exist, put the router-id of the source PE into the SPE EC.

This section extends route reflection behaviors, meaning that if support for this feature extension is required, the RR MUST perform the additional actions specified above.

7. Operational and Experimental Considerations

7.1. Experimental Scope and Topology

This document specifies an Experimental mechanism. The intended scope of the experiment is intra-domain deployment where VPN routes from multiple VRFs are exchanged over a shared BGP session. The experiment focuses on whether VPN Prefix ORF can be used to filter VPN route overload for a specific VRF or a specific set of RD, RT, Source PE, and Route Type information without negatively affecting non-overloaded VRFs.

Figure 5 illustrates a general intra-domain topology that can be used to evaluate the VPN Prefix ORF mechanism.

```
    +----------------------------------------------------------------+
    |    +-------+                                       +-------+   |
    |    | PE1   +----------------+    +-----------------+ PE4   |   |
    |    +-------+                |    |                 +-------+   |
    |                             |    |                             |
    |                             |    |                             |
    |                           +-+----+-+                           |
    |                           |  RR    |                           |
    |                           +-+----+-+                           |
    |                             |    |                             |
    |                             |    |                             |
    |    +-------+                |    |                 +-------+   |
    |    | PE2   +----------------+    +-----------------+ PE3   |   |
    |    +-------+                                       +-------+   |
    |                                                                |
    |                            AS 100                              |
    +----------------------------------------------------------------+
```

Figure 5 Intra-domain network topology

This topology can be used to verify the following:

* whether the VPN Prefix ORF mechanism can block overload routes in intra-domain scenarios.

* whether the VPN Prefix ORF mechanism conflicts with an existing mechanism and causes failure.

* whether the quota value leads to route flapping.

To achieve the finer control of VPN prefixes, the VPN Prefix ORF mechanism needs both the sender and receiver of such message support the procedures described in this document. It influences only the advertisement and withdrawn of overloaded routes, and has no impact to other non-matched VPN routes. It can lessen the receiving and parsing stress of the overloaded routes on the VPN prefixes routes receiver timely. If the sender does not support or does not honor the VPN Prefix ORF signal, the overloaded PE will still need to parse the incoming BGP messages, and the behavior falls back to the existing local protection mechanisms. Therefore, the experiment needs to assess both cases: peers that honor VPN Prefix ORF entries and peers that do not.

In order to reduce the complex configuration, it is RECOMMENDED that PEs within the network use the same formula to calculate the quota value, start from the basic mode, to the granular mode that described in the following section. Once such mechanism is triggered, it is RECOMMENDED that the operator to identify the source of overloaded VPN routes, based on the VPN Prefix ORF message, to find the root cause of overloaded VPN routes (malicious inject, or error manual configuration etc.), eliminate the advertisements of the overloaded VPN routes at the source and recover the advertisement of VPN routes into normal states.

The algorithm that triggers the VPN Prefix ORF may lead some CPU consumption on the BGP speaker, but it depends on the implementation of such algorithm. Actually, it requires only the judgement whether the overloaded VPN routes are required also by other VRFs on the same device. The implementation of this mechanism should give the flexibility on controlling of advertisement of VPN Prefix ORF message, and also the withdrawn methods of the overloaded VPN routes, based on the description of this document.

7.2. Quota value calculation

The VPN Prefix ORF mechanism is designed for intra-domain BGP/MPLS IP VPN [RFC4364] and BGP/MPLS Ethernet VPN (EVPN) [RFC7432] deployments where multiple VRFs on a Provider Edge (PE) router exchange VPN routes via a single shared iBGP session (typically with a Route Reflector).

This mechanism operates in two modes:

* Basic mode: Triggered solely by local VRF-level prefix limits. No per-source quota configuration is required. In this mode, the PE sends a VPN Prefix ORF only if all VRFs that import the same Route Target(s) have exceeded their respective prefix limits.

* Granular mode (optional): Enabled when operators configure per-<Route Distinguisher, Source PE> quotas via their Network Management System (NMS) or CLI. This enables finer-grained

control, allowing ORF triggering even if only one VRF exceeds its limit while others sharing the same RT remain non-overloaded, provided that the overload routes originate from a specific source.

Quota is a threshold to limit the number of VPN routes under specific granularities (such as <PE>). In deployment, quota values SHOULD be set and delivered by the Network Management System (NMS).

When the granular mode is enabled, an operator may configure a quota for each <RD, Source PE> tuple imported into a VRF. This quota represents the maximum number of prefixes allowed from that specific source for the given RD.

The quota value can be derived based on historical traffic patterns, service level agreements (SLAs), or static provisioning via NMS/CLI. It is not a prerequisite for the VPN Prefix ORF mechanism to operate; the mechanism defaults to VRF-level prefix limit enforcement if no per-source quotas are configured.

If the quota value is set to (VRF prefix limit/the number of PEs), whenever new PE access is added to the network, the quota value SHOULD be re-evaluated or adjusted accordingly.

To avoid frequent changes to the quota value, the value SHOULD be set based on the following formula:

Quota=MIN[(Adjust Coefficient)*<PE,CE Prefix Limit>*<Number of PEs within the VPN, includes the possibility of expansion in futures>, Local VRF Prefixes Limit]

It should be noted that the above formula is only an example; operators can use different formulas based on actual needs in the management plane.

7.3. Withdraw of VPN Prefix ORF entries

When the VPN Prefix ORF mechanism is triggered, the device SHOULD notify the network operator.

Withdrawal of VPN Prefix ORF entries is manually initiated and requires the following conditions:

1. The network operator has confirmed that the overload condition causing the VRF route limit to be exceeded has been resolved.

2. The network operator has identified the VPN Prefix ORF entry to be withdrawn. Devices SHOULD maintain records of originated VPN Prefix ORF entries for this purpose.

To withdraw VPN Prefix ORF entries, the operator configures the originating device to send a VPN Prefix ORF entry with the Action field set to REMOVE or REMOVE-ALL. The withdrawal request MUST include sufficient information to identify the target ORF entry. Automatic withdrawal of VPN Prefix ORF entries is not defined.

7.4. Potential Operational Challenges

Since this document specifies an Experimental mechanism, deployment experience is needed to determine whether the benefits of VPN Prefix ORF outweigh the potential operational risks. Operators have different views on whether dynamically suppressing VPN routes based on VPN Prefix ORF entries is beneficial or potentially dangerous. Therefore, the experiment is expected to assess the following operational challenges:

* the experiment needs to assess the risk of over-filtering. Incorrect selection of the RD, RT, Source PE, or Route Type may cause VPN Prefix ORF entries to filter routes that are still required by one or more non-overloaded VRFs.

* the experiment needs to assess the potential service impact of withdrawing or filtering matched VPN routes.

* the experiment needs to assess whether the quota value and the trigger conditions may cause route oscillation or frequent generation and removal of VPN Prefix ORF entries.

* the experiment needs to assess the interaction with existing mechanisms, including RTC, Address Prefix ORF, CP-ORF, BGP Maximum Prefix, route refresh, and local VRF import policy. The mechanism should not cause unexpected suppression of routes that are controlled by these existing mechanisms.

* the experiment needs to assess whether local independent triggering causes excessive ORF updates, duplicate or inconsistent filtering of VPN routes.

7.5. Experimental Assessment Criteria

The purpose of the experiment is not only to verify that VPN Prefix ORF can filter VPN routes that match the overload criteria, but also to determine whether the mechanism can be operated safely in intra-domain deployments.

The experiment SHOULD collect quantitative and qualitative information for at least the following aspects:

* Routing stability: the experiment SHOULD measure whether the generation, installation, and withdrawal of VPN Prefix ORF entries causes route oscillation, route churn, or unexpected loss and recovery of reachability.

* CPU and memory impact: the experiment SHOULD compare CPU and memory utilization on the overloaded PE and on the BGP speaker that applies the VPN Prefix ORF entries, with and without VPN Prefix ORF enabled, under the same route load.

* Overall efficiency: the experiment SHOULD measure the number of VPN routes received before and after VPN Prefix ORF is applied.

* Quota tuning and optimization: the experiment SHOULD evaluate whether the configured quota formula and trigger thresholds are appropriate for the deployment.

* Operational complications: the experiment SHOULD evaluate whether operators have sufficient information to understand why VPN Prefix ORF entries were generated, which routes were affected, how to troubleshoot the overload condition, and how to safely remove the entries after the root cause has been resolved.

The experiment can be considered successful only if the mechanism filters the intended overload VPN routes without causing unacceptable route instability, excessive control-plane overhead, repeated incorrect suppression of non-overloaded VPN routes, or operational complexity that outweighs the benefit of the mechanism.

8. Security Considerations

Security considerations for this work are the same as what is documented in [RFC4271].

A compromised or misconfigured BGP peer could potentially send an excessive number of VPN Prefix ORF entries. Since these entries are processed on the VPN Prefix ORF receiver, an unbounded number of VPN Prefix ORF entries could consume excessive system resources (CPU cycles for route filtering and memory for storing the entries). On devices that support the VPN Prefix ORF mechanism, it is necessary to enforce a per-peer limit on the number of VPN Prefix ORF entries. Once this limit is exceeded, this device will ignore all newly received VPN Prefix ORF entries to prioritize its own stability, rather than continuously processing new filter rules advertised by the peer.

9. IANA Considerations

9.1. VPN Prefix Outbound Route Filter

This document defines a new Outbound Route Filter type, named "VPN Prefix Outbound Route Filter (VPN Prefix ORF)", and assigns a value of 66 from the BGP Outbound Route Filtering (ORF) Types space which is under the "Border Gateway Protocol (BGP) Parameters" registry group.

Value	Description	Reference
66	VPN Prefix ORF	This document

9.2. VPN Prefix ORF TLV types

This document defines a new "VPN Prefix ORF TLV Type" registry in the "Border Gateway Protocol (BGP) Parameters" registry group. The registration policies, per [RFC8126], for this registry are as follows:

Under "Border Gateway Protocol (BGP) Parameters"
Registry: "VPN Prefix ORF TLV Type"

Range	Registration Procedures
0-127	IETF Review
128-255	First Come First Served

IANA should make initial assignments as follows:

Value	Description	Reference
0	Reserved	This document
1	IPv4 Source PE TLV	This document
2	IPv6 Source PE TLV	This document
3	Source PE Identifier TLV	This document
4	Route Target TLV	This document
5	Route Type TLV	This document
6-127	Unassigned	
128-255	Unassigned	

9.3. Source PE Extended Community

This document defines a new BGP Transitive Extended Community Type called "Source PE Extended Community" under "BGP Transitive Extended Community Types"

Under "BGP Transitive Extended Community Types"

Type Value	Name	Reference
TBD	Source PE Extended Community	This document

9.4. Common part of ORF entry

IANA is requested to make a new "ORF Entry Bits" registry in the "Border Gateway Protocol (BGP) Parameters" registry group. The registration policy for this registry is IETF Review.

IANA should make initial assignments as follows:

Bit Position	Name	Description	Reference
0-1	Action	The value of this field is 0 for ADD,1 for REMOVE, and 2 for REMOVE-ALL.	RFC5291
2	Match	The value of this field is 0 for PERMIT and 1 for DENY.	RFC5291
3	Overload VPN routes process method	The value of this field is 0 for withdrawn all overload VPN routes, and 1 for refusing to receive VPN routes matching the overload criteria.	This document
4-7	Unassigned		RFC5291

10. Contributor

Shunwan Zhuang

Huawei Technologies

Huawei Building, No.156 Beiqing Rd.

Beijing

Beijing, 100095 China

11. Acknowledgement

Jeffrey Haas, Robert Raszuk, Jim Uttaro, Jakob Heitz, Jeff Tantsura, Rajiv Asati, John E Drake, Gert Doering, Shuanglong Chen, Enke Chen, Srihari Sangli and Igor Malyushkin are thanked for their valuable comments on This document.

Thanks Qian Wang and Penglun Zhang for their development work on the FRR-based implementation of the technical solution described in this document.

12. 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>.

[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>.

[RFC4486] Chen, E. and V. Gillet, "Subcodes for BGP Cease Notification Message", RFC 4486, DOI 10.17487/RFC4486, April 2006, <https://www.rfc-editor.org/info/rfc4486>.

[RFC4684] Marques, P., Bonica, R., Fang, L., Martini, L., Raszuk, R., Patel, K., and J. Guichard, "Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)", RFC 4684, DOI 10.17487/RFC4684, November 2006, <https://www.rfc-editor.org/info/rfc4684>.

[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>.

[RFC4761] Kompella, K., Ed. and Y. Rekhter, Ed., "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, DOI 10.17487/RFC4761, January 2007, <https://www.rfc-editor.org/info/rfc4761>.

[RFC5291] Chen, E. and Y. Rekhter, "Outbound Route Filtering Capability for BGP-4", RFC 5291, DOI 10.17487/RFC5291, August 2008, <https://www.rfc-editor.org/info/rfc5291>.

[RFC5292] Chen, E. and S. Sangli, "Address-Prefix-Based Outbound Route Filter for BGP-4", RFC 5292, DOI 10.17487/RFC5292, August 2008, <https://www.rfc-editor.org/info/rfc5292>.

[RFC6074] Rosen, E., Davie, B., Radoaca, V., and W. Luo, "Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs)", RFC 6074, DOI 10.17487/RFC6074, January 2011, <https://www.rfc-editor.org/info/rfc6074>.

[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>.

[RFC7024] Jeng, H., Uttaro, J., Jalil, L., Decraene, B., Rekhter, Y., and R. Aggarwal, "Virtual Hub-and-Spoke in BGP/MPLS VPNs", RFC 7024, DOI 10.17487/RFC7024, October 2013, <https://www.rfc-editor.org/info/rfc7024>.

[RFC7117] Aggarwal, R., Ed., Kamite, Y., Fang, L., Rekhter, Y., and C. Kodeboniya, "Multicast in Virtual Private LAN Service (VPLS)", RFC 7117, DOI 10.17487/RFC7117, February 2014, <https://www.rfc-editor.org/info/rfc7117>.

[RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <https://www.rfc-editor.org/info/rfc7432>.

[RFC7543] Jeng, H., Jalil, L., Bonica, R., Patel, K., and L. Yong, "Covering Prefixes Outbound Route Filter for BGP-4", RFC 7543, DOI 10.17487/RFC7543, May 2015, <https://www.rfc-editor.org/info/rfc7543>.

[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>.

[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>.

[RFC9182] Barguil, S., Gonzalez de Dios, O., Ed., Boucadair, M., Ed., Munoz, L., and A. Aguado, "A YANG Network Data Model for Layer 3 VPNs", RFC 9182, DOI 10.17487/RFC9182, February 2022, <https://www.rfc-editor.org/info/rfc9182>.

[RFC9252] Dawra, G., Ed., Talaulikar, K., Ed., Raszuk, R., Decraene, B., Zhuang, S., and J. Rabadan, "BGP Overlay Services Based on Segment Routing over IPv6 (SRv6)", RFC 9252, DOI 10.17487/RFC9252, July 2022, <https://www.rfc-editor.org/info/rfc9252>.

Appendix A. Intra-domain Scenarios and Solutions

This section describes the workflow of some example scenarios for illustrative purposes.

A.1. Scenario 1: unique RD (per VPN, per PE)

In this scenario, PE1-PE4 and RR are iBGP peers. RD is allocated per VPN per PE. The overload VPN routes only carry one RT. the network topology is shown in Figure 6.

```
 +-----------------------------------------------------------------+
 |    +-------+                                     +-------+      |
 |    |  PE1  +-----------------+    +-----------------+  PE4  |      |
 |    +-------+                 |    |              +-------+      |
 | VPN1(RD11,RT1)               |    |           VPN2(RD12,RT2) |
 | VPN2(RD12,RT2)               |    |                            |
 |                            +-+----+-+                          |
 |                            |   RR   |                          |
 |                            +-+----+-+                          |
 |                              |    |                            |
 |                              |    |                            |
 |    +-------+                 |    |              +-------+      |
 |    |  PE2  +-----------------+    +-----------------+  PE3  |      |
 |    +-------+                                     +-------+      |
 | VPN1(RD21,RT1)                            VPN1(RD31,RT1) |
 | VPN2(RD22,RT2,RT1)                        VPN2(RD32,RT2) |
 |                                                                 |
 |                             AS 100                              |
 +-----------------------------------------------------------------+
```

Figure 6 Network Topology of Scenario 1

When PE3 sends an excessive number of VPN routes with RT1, and both PE1 and PE2 import VPN routes with RT1, the process of overload VPN routes will influence performance of VRFs on PEs. PEs and RR need to have appropriate mechanisms to identify and control the advertising of overload VPN routes.

a) PE1

If no quota value is set on PE1 and each VRF on PE1 has a prefix limit, when PE1 receives VPN routes from its BGP peer, it performs the following actions:

```
S01. If (the prefix limit for VPN1 VRF is exceeded){
S02.     PE1 sends a VPN Prefix ORF message to the
         RR and a warning message to the operator.
         The VPN Prefix ORF message carries the
         following parameters: RD set to RD31, RT
         value set to RT1, and source PE set to PE3.
         The RR processes the overload VPN routes
         and controls the number of VPN routes based
         on the value of the "Overload VPN routes
         process method" field.
S03. } else {
S04.     PE1 does not trigger the VPN Prefix ORF
         mechanism and only performs VPN route
         filtering for the target VRF.
S05. }
```

NOTE: When the prefix limit for the VPN1 VRF is exceeded, no other VRFs on PE1 import VPN routes with RT1. PE1 sends a VPN Prefix ORF message to the RR and a warning message to the operator.

If a quota is configured for each <RD31, source PE3> tuple imported into a VRF and each VRF has a prefix limit, when PE1 receives VPN routes from its BGP peer, it performs the following actions:

```
S01. If (VPN routes associated with <RD31, PE3>
     tuple exceed the quota) {
S02.     If (the prefix limit of the VPN1 VRF
         is not exceeded) {
S03.         PE1 sends a warning message to the
             operator, and the VPN Prefix ORF
             mechanism is not triggered.
S04.     } else {
S05.         PE1 generates a BGP ROUTE-REFRESH
             message containing a VPN Prefix ORF
             entry with the parameters (RD = RD31,
             source PE = PE3, RT = RT1), and sends
             this entry to the RR.
             The RR processes the overload VPN
             routes based on the value of the
             "Overload VPN routes process method".
S06.     }
S07. }
```

b) PE2

If no quota value is set on PE2 and each VRF on PE2 has a prefix limit, when PE2 receives VPN routes from its BGP peer, it performs the following actions:

```
S01. If (the prefix limit for the VPN1 VRF is exceeded) {
S02.     If (the prefix limit for the VPN2 VRF is exceeded) {
S03.         PE2 sends a VPN Prefix ORF message to the RR and a
             warning message to the operator. The VPN Prefix ORF
             message specifies the RD set to RD31 and the RT
             value set to RT1. The RR processes the overload VPN
             routes and controls the number of VPN routes based
             on the value of the "Overload VPN routes process
             method" field.
S04.     } else {
S04. } else {
S05.     PE2 does not trigger the VPN Prefix ORF mechanism and
         only performs VPN route filtering for the target VRF.
S06. }
```

NOTE: PE2 does not directly trigger the VPN Prefix ORF mechanism when the prefix limit of the VPN1 VRF is exceeded, because the VPN2 VRF imports VPN routes with RT1. PE2 triggers the mechanism only when the prefix limits for both the VPN1 and VPN2 VRFs are exceeded.

If a quota is configured for each <RD31, source PE3> tuple imported into a VRF and each VRF has a prefix limit, when PE2 receives VPN routes from its BGP peer, it performs the following actions:

```
S01. If (the VPN routes associated with the <RD31, PE3> tuple
     exceed the quota) {
S02.     If (the prefix limit of the VPN1 VRF is not exceeded) {
S03.         PE2 sends a warning message to the operator, and the
             VPN Prefix ORF mechanism is not triggered.
S04.     } else {
S05.         If (the prefix limit of the VPN2 VRF is not exceeded)
             {
S06.             PE2 does not trigger the VPN Prefix ORF mechanism
                 and only performs VPN route filtering for the
                 target VPN1 VRF, stopping the import of VPN routes
                 associated with <RD31, PE3>.
S07.         } else {
S08.             PE2 generates a BGP ROUTE-REFRESH message
                 containing a VPN Prefix ORF entry with the
                 parameters (RD31, source PE = PE3, RTs = RT1 and
                 RT2), and sends this entry to the RR. The RR
                 processes the overload VPN routes based on the
                 value of the "Overload VPN routes process method"
                 field.
S09.         }
S10.     }
S11. }
```

A.2. Scenario 2: the same RD (per VPN, same on all PEs)

In this scenario, PE1-PE4 and RR are iBGP peers. RD is allocated per VPN. One/Multiple RTs are associated with the overload VPN routes and are imported into different VRFs on other PEs. The network topology is shown in Figure 7.

```
    +----------------------------------------------------------------+
    |                                                                |
    |                                                                |
    |    +-------+                                       +-------+   |
    |    |  PE1  +----------------+    +-----------------+  PE4  |   |
    |    +-------+                |    |                 +-------+   |
    | VPN1(RD1,RT1)               |    |              VPN2(RD12,RT2) |
    | VPN2(RD12,RT2)              |    |                             |
    |                           +-+----+-+                           |
    |                           |   RR   |                           |
    |                           +-+----+-+                           |
    |                             |    |                             |
    |                             |    |                             |
    |    +-------+                |    |                 +-------+   |
    |    |  PE2  +----------------+    +-----------------+  PE3  |   |
    |    +-------+                                       +-------+   |
    | VPN1(RD1,RT1)                                VPN1(RD1,RT1,RT2) |
    |                                              VPN2(RD32,RT2)    |
    |                                                                |
    |                             AS 100                             |
    |                                                                |
    +----------------------------------------------------------------+
```

Figure 7 Network Topology of Scenario 2

When PE3 sends an excessive number of VPN routes associated with RD1, RT1 and RT2, and both PE1 and PE2 import VPN routes with RT1, the process of overload VPN routes can affect the performance of the VRFs on PEs.

a) PE1

If no quota value is set on PE1 and each VRF on PE1 has a prefix limit, PE1 does not directly trigger the VPN Prefix ORF mechanism when the prefix limit of the VPN1 VRF is exceeded, because the VPN2 VRF imports VPN routes with RT2. This case is similar to that of PE2 without a quota in Scenario 1, with modifications as follows:

S03.   PE1 sends a VPN Prefix ORF message to the RR and a warning message to the operator. The VPN Prefix ORF message specifies the RD set to RD1, the RT values set to RT1 and RT2, and the source PE identified as PE3. The RR processes the overload VPN routes and controls the number of VPN routes based on the value of the "Overload VPN routes process method" field.

If a quota is configured for each <RD1, source PE3> tuple imported into a VRF and each VRF has a prefix limit, this case is similar to that of PE2 with a quota in Scenario 1, with modifications as follows:

S08. PE1 generates a BGP ROUTE-REFRESH message containing a VPN Prefix ORF entry with the parameters (RD1, source PE = PE3, RTs = RT1 and RT2), and sends this entry to the RR. The RR processes the overload VPN routes based on the value of the "Overload VPN routes process method" field.

b) PE2

If no quota value is set on PE2 and each VRF on PE2 has a prefix limit, since only the VPN1 VRF needs to import VPN routes with RT1, this case is similar to that of PE1 without a quota in Scenario 1, with modifications as follows:

S02. PE2 sends a VPN Prefix ORF message to the RR and a warning message to the operator. The VPN Prefix ORF message specifies the RD set to RD1, the RT values set to RT1 and RT2, and the source PE identified as PE3. The RR processes the overload VPN routes and controls the number of VPN routes based on the value of the "Overload VPN routes process method" field.

If a quota is configured for each <RD31, source PE3> tuple imported into a VRF and each VRF has a prefix limit, this case is similar to that of PE1 with a quota in Scenario 1, with modifications as follows:

S05. PE2 generates a BGP ROUTE-REFRESH message containing a VPN Prefix ORF entry with the parameters (RD1, source PE = PE3, RTs = RT1 and RT2), and sends this entry to the RR. The RR processes the overload VPN routes based on the value of the "Overload VPN routes process method" field.

Appendix B. Applicability

Using scenario 1 in Appendix B, this section demonstrates how to determine each field when the sender generates a VPN Prefix ORF entry. Assuming an IPv4 network. When the VPN Prefix ORF mechanism is triggered on PE1, PE1 generates a VPN Prefix ORF entry that contains the following information:

* AFI is equal to IPv4

* SAFI is equal to MPLS-labeled VPN address

* When-to-refresh is equal to IMMEDIATE

* ORF Type is equal to VPN Prefix ORF

* Length of ORF entries is equal to 41

* Action is equal to ADD

* Match is equal to DENY

* Overload VPN routes process method is equal to 0

* Sequence is equal to 1

* Length is equal to 27

* Route Distinguisher is equal to RD31

* Optional TLV:

  - Type is equal to 1 (Source PE TLV)

  - Length is equal to 4

  - value is equal to PE3’s IPv4 address

  - Type is equal to 5 (Route Target TLV)

  - Length is equal to 8

  - value is equal to RT1

Authors’ Addresses

Wei Wang
China Telecom
Beiqijia Town, Changping District
Beijing
Beijing, 102209
China
Email: weiwang94@foxmail.com

Aijun Wang
China Telecom
Beiqijia Town, Changping District
Beijing
Beijing, 102209
China
Email: wangaj3@chinatelecom.cn

Haibo Wang
Huawei Technologies
Huawei Building, No.156 Beiqing Rd.
Beijing
Beijing, 100095
China
Email: rainsword.wang@huawei.com

Gyan S. Mishra
Verizon Inc.
13101 Columbia Pike
Silver Spring, MD 20904
United States of America
Email: hayabusagsm@gmail.com

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