Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 31, 2012

P. Marques
Contrail Systems
L. Fang
Cisco Systems
D. Winkworth
FIS
Y. Cai
Microsoft Corporation

May 2012

Edge multicast replication for BGP IP VPNs. draft-marques-l3vpn-mcast-edge-00

Abstract

In data-center networks it is common to use Clos network topologies [clos] in order to provide a non-blocking switched network. In these topologies it is often not desirable to provide native IP multicast service.

This document defines a multicast replication algorithm along with its control and data forwarding procedures that provides a multicast service for a BGP IP VPN network without assuming that the underlying infrastructure supports IP multicast.

Status of this Memo

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

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

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

This Internet-Draft will expire on October 31, 2012.

Table of Contents

1. Introduction

In Wide-Area Networks having native multicast service on hop-by-hop basis allows for more efficient use of scarse link bandwidth. In Clos network topologies [clos] the trade-offs are different.

A Clos network is often used to provide full cross-sectional bandwidth between all the ports on the network. When used in a switching infrastructure it achieves this goal by spreading flows across multiple equal cost paths.

For Clos topologies with multiple stages native multicast support within the switching infrastructure is both unnecessary and undesirable. By definition the Clos network has enough bandwidth to deliver a packet from any input port to any output port. Native multicast support would however make it such that the network would

no longer be non-blocking. Bringing with it the need to devise congestion management procedures.

In this type of environments it is desirable to provide multicast service as an edge functionality on top of a unicast clos fabric. Early versions of IP VPN multicast services have used ingress replication. The drawback with that approach is the load imposed on the ingress node which is specially relevant for situations in which the multicast group has a large number of receivers. This document

takes a different approach by leveraging the receivers in order to build an edge based replication tree on a per-flow basis.

Data-center networks often require network virtualization services such as the one described in [I-D.marques-l3vpn-end-system]. This document defines a set of procedures to be implemented in a VPN forwarder in order to provide multicast service for a BGP IP VPN.

It meets several important requirements:

- Support for both source-specific and shared multicast trees.
- Support for variable degrees of replication per tree node.
- Loop-free forwarding topology.

The solution itself does not assume a specific topology on the underlying infrastructure network. We simply assume that it is undesirable to use native multicast service. This can be a result of topology as per the CLOS example above or some other constraint that makes it undesirable to create multicast groups based on the overlay topology.

2. Overview

IP hosts use IGMP [RFC3376]/MLD [RFC3810] to request the delivery of multicast packets for a particular (*, g) or (s, g). Discovery applications where the intent is to allow applications to discover the group membership use (*, g) JOINs. Content delivery applications may use an (s, g) JOIN after initially performing discovery either via multicast or by other means.

In the context of end-system VPNs, the VPN Forwarder acts as an IGMP querier on the virtual interfaces and receives IGMP/MLD Membership Report packets. It uses this information to generate VPN-specific multicast membership information.

This information is communicated to the Signaling Gateway as a triple (vrf-id, s/*, g) via an XMPP publish request. This is similar to the process used to publish unicast IP addresses associated with virtual-interfaces.

This message also indicates the label range that can be used to assign 20-bit forwarding labels to this multicast traffic flow. The same label range can be shared between different multicast groups. It is the responsibility of the VPN Gateway to ensure that a given label is not used for multiple groups simultaneously.

VPN Forwarders can choose to advertise a single label range for all multicast groups or different label ranges for different sets of multicast groups. The set granularity can be as small as single multicast group.

The label range advertised by the VPN Forwarder should be larger than the expected number of active multicast groups within the set plus an additional constant that ensures that a label will not be reused within a time frame greater than the time it takes for topology updates to propagate.

Signaling Gateways construct multicast distribution trees such that each node in the tree is a VPN Forwarder and each node in the tree has no more than K-edges where K is defined by configuration. The parameter K may be different for different VPN gateways.

The multicast distribution tree is an acyclic graph. The Signaling Gateway assigns edges between nodes ensuring that all nodes are connected and there are no cycles. The resulting graph is a spanning tree.

The Signaling Gateway can use any algorithm to manage the graph. In practice, we expect that the Signaling Gateway would attempt to minimize the cost of the tree subject to the out-degree constraint (at most K edges) while also minimizing the disruption caused by each individual node JOIN or LEAVE.

The Signaling Gateway constructs an OLIST for each VPN Forwarder, where its OLIST is constituted by an incoming edge (for all nodes except for the root) plus up-to K outgoing edges.

Whenever the OLIST for a given node changes, the Signaling Gateway MUST allocate a different label that corresponds to that version of the OLIST. This is used to avoid forwarding loops. The assumption is that at each run of its tree management algorithm the Gateway is capable of building a acyclic graph. However signaling updates from the Gateway to the VPN Forwarders are not synchronous. Each modified OLIST will have a different label assigned, which means that in transient state traffic may be discarded if a VPN forwarder with information regarding an old edge send traffic to a VPN forwarder which has already received information of the new topology. However this eliminates the possibility of forwarding loops.

Traffic forwarding is done according to a bi-directional forwarding algorithm. Packets flowing from the root are distributed to all the outgoing edges. Traffic received from one of the leaves is sent to the root facing interface plus remaining descendants. This assumes that the VPN forwarder has the ability to determine the source of the traffic, by examining the outer IP header of the packet.

Signaling Gateways communicate multicast membership information to each other using BGP L3VPN C-Multicast routes [RFC6514]. Associated with each C-Multicast route, the Signaling Gateway also advertises up-to K edges that can be use to interconnect the multicast distribution tree that it manages with other trees managed by its

peers. The C-Multicast routes are known to all signaling gateways which have local membership in the corresponding VPNs.

A predefined hash function is used to determine a 32-bit value X associated with the specific multicast group. This value is used to elect the multicast tree manager for the specific group. The tree manager is the Signaling Gateway for which the value (RouterId - X) is lower using 32-bit unsigned math.

As previously described in the case of the Signaling Gateways managing distribution trees of VPN Forwarders, the tree manager is responsible to determine the edges between the several nodes in order

to build an acyclic graph. In this case the nodes are themselves replication trees.

The tree manager is responsible to assign the forwarding labels used by the particular graph edge. These labels are offered in the C-Multicast membership information as a list of available labels per edge.

3. VPN Forwarder behavior

VPN Forwarders act as IGMP/MLD queriers on the virtual interfaces that provide connectivity to end-systems. They receive IGMP/MLD Membership Report packets on these point-to-point interfaces which are then used to build the local per VRF membership information.

Each VRF may have a list of (s, g), (*, g) and (*, *) multicast routing entries associated with it. These are the result of IGMP/MLD Membership Reports or Queries. Routing entries can also be created as a result of detecting a local source on one of the virtual-interfaces associated with the VRF.

Multicast groups in the Source-Specific Multicast [RFC4607] address prefix use both (s, g) and (*, g) routing entries while the Any-Source Multicast (ASM) groups use (*, g) routing entries only.

The forwarding table on a VPN Forwarder contains (vrf, *, g) entries for ASM groups and (s, g) entries for SSM groups. Multicast packets that do not match an existing forwarding entry SHALL result in the creation of a local routing entry, when received from a virtual interface. The VPN Forwarder MAY decide to hold on the the first packet that triggers the creating of a routing entry.

Locally-know multicast routes, either the result of IGMP/MLD Membership Reports or locally sourced traffic are subject to expiration.

When a multicast route is created locally, the VPN Forwarder generates an XMPP subscription message to the corresponding vrf-name,

group and source. When the source is not specified a (*, g) is implied. When a multicast router is detected on the virtual-interface, via the receipt of IGMP/MLD Query messages the VPN forwarder subscribes to the group 0.0.0.0.

Group Join form VPN Forwarder to gateway:

```
<iq type='set'
    from='01020304abcd@domain.org'  <!-- VPN forwarder system-id -->
    to='network-control.domain.org'
    id='sub1'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <subscribe node='vpn-customer-name/224.1.1.1'/>
    <options>
      <x xmlns='jabber:x:data' type='submit'>
        <field var='label-range'><value>10000-20000</value></field>
      </x>
    </options>
  </pubsub>
</iq>
```

Signaling Gateways build the multicast distribution tree for a specific group. When the distribution tree is built, the signaling gateway will include as members all the (*, *) receivers of ASM groups and all (*, *) and (*, g) receivers of SSM groups.

Once the subscription is received, the gateway sends XMPP event notifications that contain forwarding information for the specific group. These messages contain an incoming label, assigned by the gateway, and a list of up-to K+1 next-hops, where each next-hop consists of an IP destination address and an outgoing label.

Multicast forwarding state update from gateway to VPN forwarder:

```
<message to='system-id@domain.org from='network-control.domain.org>
  <event xmlns='http://jabber.org/protocol/pubsub#event'>
    <items node='vpn-customer-name/224.1.1.1'>
      <item id='ae890ac52d0df67ed7cfdf51b644e901'>
        <entry xmlns='http://ietf.org/protocol/bgpvpn'>
      <label>10000</label>      <!-- incoming label number -->
      <olist>
        <next-hop address='10.1.1.1' label='10101'/>
        [...]
        <next-hop address='10.1.10.10' label='10222'/>
      </olist>
        </entry>
       </item>
      <item >
        ...
      </item>
    </items>
  </event>
</message>
```

The VPN forwarder updates its multicast forwarding table with the information received in this event notification. Any label that was previously assigned to the (vrf, *, g) or (vrf, s, g) forwarding entry is implicitly withdrawn.

Multicast packets are encapsulated in an IP tunnel that contains a 20-bit as well as the original multicast datagram. This 20-bit label uniquely identifies the multicast replication state as specified by the OLIST.

The VPN Forwarder MUST drop an incoming multicast packet unless it is either received from a local virtual interface or the source is present in the OLIST.

The VPN Forwarder MUST generate a copy of the incoming packet to all next-hops in the OLIST except the next-hop with the same IP address as the outer header source of the incoming packet.

Additionally, the VPN Forwarder MUST generate additional copies to the virtual interfaces associated with the VRF that have expressed interest in the specific multicast group.

4. Multicast tree management

The multicast forwarding tree associated with a specific multicast group is built hierarchically. At the lowest level, Signaling Gateways build a acyclic graph in which nodes are VPN Forwarders and where nodes have up-to (K+1) edges. Above this level, the graph nodes are multicast replication trees themselves.

At the lowest level, VPN Forwarders implicitly select the signaling gateway responsible to manage its tree by subscribing to a single gateway. At higher levels, the forwarding tree manager is elected by selecting the gateway with the smaller value of (RouterId - HashFunction(g)) in unsigned 32-bit arithmetic.

While the multicast tree management algorithm is a local matter to the gateway implementation, the algorithm used SHOULD minimize the height of the multicast replication tree and attempt to minimize the number of state changes to the tree. As an example Prim's algorithm [prim] can be used to generate a minimum spanning tree.

In this application all the nodes in the graph can have an edge to any other node as long as the total number of edges does not exceed K + 1. The implementation may choose to assign the same cost to all the edges or i may use external information to determine cost. For instance, an implementation may choose to assign lower cost to edges between nodes in the same server rack versus nodes in different racks.

Signaling gateways assign forwarding labels from an interval provided by the VPN Forwarder. Whenever the tree topology changes such that nodes in with different versions of the topology could create a forwarding loop the gateway MUST allocate a new label. When leaf nodes in the tree are added or deleted these changes can be performed without concern for the formation of transient loops. However, in the case of tree rotations to rebalance the tree, there is a clear potential for forwarding loops.

In generic terms, a transient forwarding loop can be formed if there exist multiple versions of the graph that are being executed by different nodes. As an example consider a graph with nodes (a, b, c)

that goes through the following topology versions:

Version	Edges
1	a-b, a-c
2	a-b, b-c
3	a-c, b-c

In a scenario where node 'a' is at version 1, node 'b' at version 2 and node 'c' at version 3 a transient loop will occur. In this example, a packet that is injected at node 'a' will propagate to both 'b' and 'c'. 'b' accepts the packet since the edge (a-b) is a valid edge and propagates the traffic to 'c'. 'c' accepts packet from both 'a' and 'b'. The packet 'c' receives from 'b' will be forwarded to 'a' which will accept it, creating a loop. Likewise the packet 'c' receives from 'a' will be forwarded to 'b', which will then forward to 'a'.

All of the topologies in the example above are loop-free. However the fact that routing information propagates is not synchronized allows for the formation of loops.

Given that the propagation of forwarding entries to VPN Forwarders is asynchronous and that it would be undesirable to attempt to synchronize the process, we use the incoming label to break the potential forwarding loop. For the loop to be broken it is necessary that the forwarding labels used in the edge (a-c) in the example above be different in configuration 1 versus configuration 3.

As an example, forwarding loop avoidance can be implemented by keeping a list of edges that have been previously present in a node and modifying the label every time the tree management algorithm adds an edge that had been removed from the node.

A Signaling Gateway that has received multicast routing information from locally connected VPN Forwarders shall advertise the corresponding multicast group as a C-Multicast route. These C-Multicast routes shall include an Edge Discovery attribute that describes up-to K + 1 multicast next-hops, each containing an IP address and a label range that can be used to assign forwarding

labels.

The tree manager election algorithm selects which of the signaling gateways is responsible to determine the topology of the multicast distribution tree. At this level in the hierarchy, the distribution tree consists of graph nodes that are themselves distribution trees. In the case where tree nodes were VPN Forwarders, the tree management algorithm can assign up-to (K + 1) edges to a node (where K can potentially be configurable per node). In the case where tree nodes are distribution trees, the tree management algorithm is limited to the number of edges received in the C-Multicast route.

The algorithm used to manage the lower and higher levels in the hierarchy can be the same.

When the tree manager modifies the tree topology it shall generate BGP routes that describe the current topology. These routes are encoded using the MCAST-VPN NLRI using the Multicast Tree Route Type defined bellow.

Multicast Tree routes contain an Edge Forwarding attribute that describes the active edges between different nodes.

Multicast Tree routes are interpreted only by the Signaling Gateway that is identified by the Router-Id contained in the NLRI prefix. On a receipt of such a route, the Signaling Gateway connects its own multicast distribution tree with the edges contained in the Edge Forwarding attribute.

In order to illustrate the operation of the hierarchical label management process, we present the following example.

Consider a scenario where the out-degree constant K is 4 and where 4 Signaling Gateways (A, B, C, D) are present. The Signaling Gateways are the multicast tree managers for a set of VPN forwarders. We use the notation a1 .. an for the VPN forwarders managed by node A.

Assume that Signaling Gateway A has built a multicast distribution tree such that node a3 is the root. This node has 2 descendants a1 and a2. Each of these have at most 3 locally connected edges. In this scenario the Signaling Gateway A has chosen to reserve edges at the top of its tree in order to connect to external trees.

Signaling Gateway A advertises its state to the BGP network by

generating a C-multicast route containing a Multicast Edge Discovery attribute with the next-hops (a1, a1, a2, a3). Although signaling gateway A may have many other VPN forwarders that are receivers of the specified group this information is not propagated through BGP.

Each of the next-hops in the list has an incoming label that is currently in use. The Edge Discovery attribute contains a interval of free (unused labels) that is valid for each of the next-hops.

In this example, we assume that the Signaling Gateway B was elected as the tree manager for the higher level tree. At this stage, the tree manager has the following state:

Router-Id	Edges
A	(a1, 0), (a1, 0), (a2, 0), (a3, 0)
B	(b1, 0)
C	(c1, 0), (c2, 0), (c3, 0)
D	(d1, 0), (d2, 0), (d3, 0)

In this example all the signaling gateways decided to advertise less than K+1 edges.

One possible assignment is to make the node A the root of the top-level distribution tree.This can be accomplished by creating the edges (a1, b1), (a1, c1), (a2, d1). The tree manager must allocate a label for each of the next-hops from their respective label space.

As a result of this tree assignment, the multicast tree manager (B) generates the following Multicast Tree Route Type updates:

Router-Id	Edges
A	(a1, b1, 10000, 20000), (a1, c1, 10000, 21000), (a2, d1, 11000, 22000)
C	(c1, a1, 21000, 10000)
D	(d1, a2, 22000, 11000)

When A, C and D receive their respective routing updates they will generate the corresponding XMPP event notification messages to the

affected VPN forwarders. In A's case this implies updating the state of a1 and a2. a1 forwarding table now has a new incoming label (10000) and the next-hops (b1, a2, a3, c1).

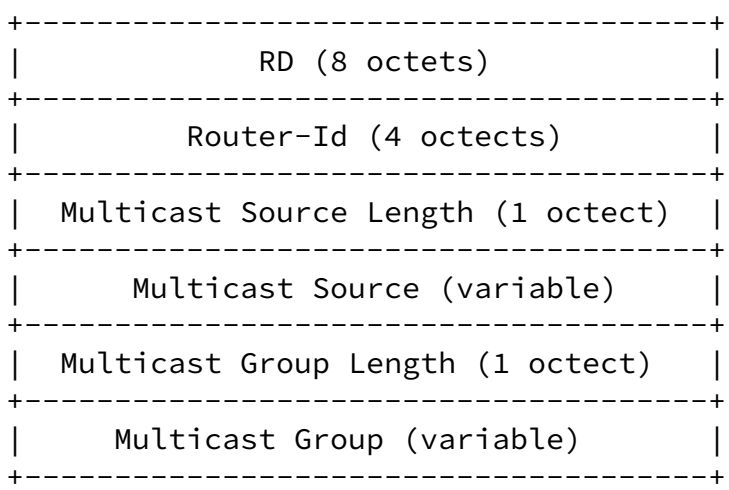
5. BGP Protocol Extensions

This document defines an additional Route Type for the MCAST-VPN NLRI [RFC6514], called Multicast Tree Route Type.

5.1. Multicast Tree Route Type

Multicast Tree Routes are used by multicast tree managers to advertise a acyclic graph topology of nodes which themselves may consist of multicast distribution trees. A BGP UPDATE containing a Multicast Tree Route as part of the MP_REACH Path Attribute MUST also contain the Multicast Edge Forwarding Attribute.

A Multicast Tree Route is encoded as a MCAST-VPN NLRI with Route Type 8 and consists of the following:



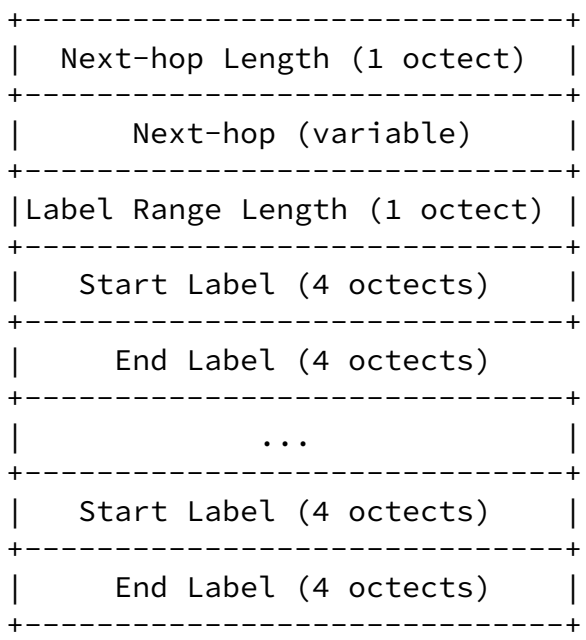
The Route Distinguisher (RD) is encoded as described in [RFC4364].

The Router-Id field identifies the multicast tree node for which the edges are being advertised.

The Multicast Source and Group fields specify the multicast group for which the multicast forwarding state is being advertised.

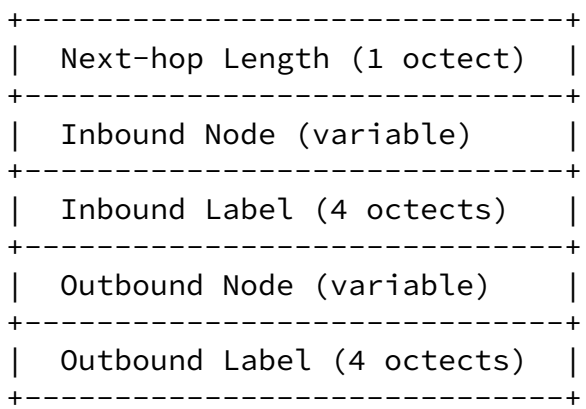
5.2. Multicast Edge Discovery Attribute

The Multicast Edge Discovery Path Attribute is associated with C-Multicast routes and contains one or more next-hop information elements where each information element follows the model described bellow:



5.3. Multicast Edge Forwarding Attribute

The Multicast Edge Forwarding Path Attribute is associated with Multicast Tree Route Type NLRI routes and contains one or more edge information elements where each information element follows the model described bellow:



Each edge element contained in this list contains the address on an inbound node that has been advertised via the Edge Discovery attribute as well as a label assigned from the respective interval. The outbound node address and label connect specify the destination node for this edge.

6. Security Considerations

It is helpful to differentiate between the control plane and data plane security aspects of the solution.

The control plane assumes that XMPP sessions between VPN forwarders and Signaling Gateway are authenticated such that the Signaling Gateway is able to verify the identity of the VPN Forwarder.

BGP sessions bewteen Signaling Gateways should also be subject to authentication.

At the data-plane, it is important to note that a comprimised VPN forwarder is able to modify message that traverse through it.

7. References

7.1. Normative References

[RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B. and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.

[RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.

[RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, August 2006.

[RFC6514] Aggarwal, R., Rosen, E., Morin, T. and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, February 2012.

7.2. Informational References

[I-D.marques-l3vpn-end-system]
Marques, P., Fang, L., Pan, P., Shukla, A., Napierala, M. and N. Bitar, "BGP-signaled end-system IP/VPNs.", Internet-Draft draft-marques-l3vpn-end-system-05, March

2012.

[clos] "A study of non-blocking switching networks", Bell System Technical Journal 32, March 1953.

[prim] Prim, R.C., "Shortest connection networks and some generalizations", Bell System Technical Journal 36, 1957.

Authors' Addresses

Pedro Marques
Contrail Systems
440 N. Wolfe Rd.
Sunnyvale, CA 94085

Email: roque@contrailsystems.com

Luyuan Fang
Cisco Systems
111 Wood Avenue South
Iselin, NJ 08830

Email: lufang@cisco.com

Derick Winkworth
FIS

Email: derick.winkworth@fisglobal.com

Yiqun Cai
Microsoft Corporation
1065 La Avenida
Mountain View, CA 94043

Email: yiqunc@microsoft.com