Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 13, 2012

P. Marques
L. Fang
Cisco Systems
P. Pan
Infinera Corp
A. Shukla
Juniper Networks

December 2011

End-system support for BGP-signaled IP/VPNs. draft-marques-l3vpn-end-system-03

Abstract

Network Service Providers often use BGP/MPLS IP VPNs [RFC4364] as the control plane for overlay networks. That solution has proven to scale to large number of VPNs and attachment points and is one familiar to network equipment software.

There is a significant interest in the industry in building overlay networks in which end-systems are themselves the direct participant, along with network equipment such as service appliances.

This document proposes an extension of the BGP IP VPN model to serve as the signaling protocol for host-based overlay networks along with an XMPP interface that provides a bridge between the software concepts familiar to end-points and those familiar to network equipment.

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 June 13, 2012.

Table of Contents

1. Introduction 2
2. End-system functionality 3
3. Virtual Machine Networking 4
4. Operational Model 6
5. XMPP client interface 9
6. VPN NLRI 11
7. Security Considerations 12
8. Acknowledgements 12
9. References 12
Authors' Addresses 13

1. Introduction

Data center applications require private networks connecting multiple "Virtual Machines" belonging to the same administrative "user" and between them and network elements and appliances.

In this context, it is a common goal, for the data-center forwarding infrastructure to be isolated from the knowledge of the private network. The set of routers and switches that interconnects physical machines in the data-center is assumed to provide an IP service (with or without the use of IEEE 802.1 technologies).

The Virtual Private Networks (VPNs) associated with each individual administrative domain can be built without the knowledge of the data-center connectivity layer as an overlay network. This proposal leverages the technology used in the Service Provide managed VPN space and extends it to address the problem of interconnecting virtual interfaces on end-systems. In both applications there is the need to be able to manage at scale a very large number of VPNs and attachement points. And in both cases there is the need to support the interchange of traffic between different VPNs.

This document defines how BGP-signaled IP/VPNs can be used to interconnect end-systems and network elements. It assumes that the forwarding layer uses IP over GRE as defined by [RFC4023]. Other transport layers such as native MPLS or 802.1ah can also be used with

the same signaling approach.

When this document uses the term 'Infrastructure IP' addresses, it refers to the addresses used in the outer header of GRE packets. In the case of a transport other than IP over GRE, this would be the

Subnetwork Point of Attachement (SNPA) address corresponding to the multi-access network providing connectivity to the end-systems.

BGP is not an interface that application software is familiar with. In order to bridge the gap between concepts familiar to network devices and those familiar to end-system developers, this document defines an XMPP client interface to be used by end-systems. It defines the procedures to interchange data between XMPP and BGP IP VPN sessions along with the corresponding data schemas. Networking devices may opt to receive the signaling information directly via BGP.

2. End-system functionality

For the purposes of this document we assume that each end-system executes an 'Host Operating System' with the ability to:

- Create virtual interfaces (typically ethernet interfaces).
- Associate a given virtual interface with a specific "Virtual and Routing Forwarding (VRF)" table.
- Store entries in the VRF table that map an VRF-specific IP prefix into a next-hop which contains a destination IP address and a 20-bit label.
- Encapsulate outgoing packets according [RFC4023] using the result of the VRF lookup.
- Associate incoming packets with a VRF according to the 20-bit label contained immediately after the GRE header.
- Expose a programmatic interface to create, update and delete VRF table entries.

The 'Host Operating System' may choose to associate the virtual interfaces with specific 'Virtual Machines' or use other policies to manage the application access to these interfaces.

The description above assumes that each virtual interface is associated with its own VRF table as the contents of the VRF table are not known in advance. If two virtual interfaces belonging to the same host are allowed to exchange traffic the default behavior of an IP implementation would result in the traffic being encapsulated into GRE and delivered by the IP stack to the host itself via a loopback mechanism.

As an optimization, hosts MAY support the ability to associate multiple virtual interfaces with the same VRF, for interfaces associated with the same VPN. When that is the case, locally known routes, that is IP routes to the local virtual interfaces SHALL NOT be used to forward outbound traffic (from the virtual interfaces to the outside) unless a route advertisement has been received that matches that specific IP prefix and next-hop information. As an example, if a given VRF contains two virtual interfaces, "veth0" and "veth1", with the addresses 10.0.1.1/32 and 10.0.1.2/32 respectively, the initial forwarding state must be initialized such that traffic from either of these interfaces does not match the other's routing table entry. It may for instance match a default route advertised by a remote system. Traffic received from the tunnel interface however must be delivered to the correct local interface. If at a subsequent stage a route is received from the signaling gateway such that 10.0.1.2/32 has a next-hop with the IP address of the local host and the correct label, the system may subsequently install a local routing table entry that delivers traffic directly to the "veth1" interface.

The 20-bit label which is associated with a virtual-interface is of local significance only and SHOULD be allocated by the end-system.

The procedure that determines that a VRF should be configured on a particular end-system as well as which IP addresses to be associated with each interface are outside the scope of this document. We assume that statically assigned IP addresses are used.

The VRFs support IP unicast traffic only. Multicast support is subject for further study and will be detailed in a separate document. Both IPv4 and IPv6 are supported and the term 'IP' can refer to either version of the Internet Protocol.

The VRF table is populated by the signaling mechanisms described bellow and may contain both host length (i.e. /32 and /128 for IPv4 and v6 respectively) or subnet prefixes. As an example a VPN with access to the external networks would probably contain a default route plus a set of host length entries for all the Virtual Machines (VMs) in the same VPN.

In the terminology used in the BGP-signaled IP/VPN standard [RFC4364], a end-system acts as a 'Provider Edge' (PE) device in terms of its forwarding capabilities, with the virtual interfaces that it exposes (for instance to virtual machines) as the 'Customer Edge' (CE) interfaces.

3. Virtual Machine Networking

When virtual machines are associated with a virtual interface on the end-system, this document assumes that there is a single route entry to a default route on the Virtual Machine (VM). Packets are then

routed by the Host OS, which imposes the VPN encapsulation header. Link-local addresses on the virtual ethernet interface that connects the virtual machine are not globally significant.

When discussing VM connectivity, it is frequent to encounter the assumption that the VM routing table contains a subnet route entry with reachability to all other VMs in the VPN.

VM route table using LAN adjacency:

IP prefix	Next-hop	Interface
10.1.1.1/32	local	veth0
10.1/16	direct	veth0

In the scenario above, the VM assumes a direct LAN adjacency with its peers (e.g. 10.1.1.2). It uses ARP to build an L2 adjacency to its communication peers. Scaling ARP broadcasts and updating ARP entries across the data-center becomes an important problem. Often the conclusion reached is that the VM mac-addresses must be global and constant for a given VM. That can be a problematic requirement when interconnecting data-centers administered by different orchestration systems.

This document proposes a VM routing table configuration where there are no ARP adjacencies between different VMs.

VM route table using default gateway:

IP prefix	Next-hop	Interface
10.1.1.1/32	local	veth0
169.254.254.254/32	direct	veth0
0/0	169.254.254.254	veth0

The configuration above eliminates the need for L2 adjacencies between VMs. The VM contains a single ARP entry to its default gateway, which is the Host OS. The Host OS performs a route lookup in the corresponding VRF in order to route the packet. In this approach, the Host OS VRF is the point of control that determines the destination end-system associated with the VPN destination IP address.

One of the advantages of this model is that it eliminates the need to support broadcast across a VPN.

The guest OS should be configured with a default gateway address in the IP link-local address space. This address should be constant across all hosts that the VM can be instantiated on. The example above uses the highest numbered address in the IPv4 link-local range and assumes that the Host OS has been configured to recognize that address as local and answer local ARP requests on the virtual interface.

4. Operational Model

In the simplest case, a VPN is a collection of systems that are allowed to exchange traffic with each other and where all the VRFs in the VPN contain all the routing entries for the VPN. Only members of the VPN are allowed to exchange traffic with each other. We can refer to these as symmetrical VPNs since all VRFs contain the same routing information.

When end-systems join a given VPN they advertise their membership by advertising the VPN-specific IP address associated with a particular virtual interface as well as its binding to the infrastructure IP address associated with the host.

Infrastructure addresses are routable in the underlying transport network (e.g. the data-center network). While VPN addresses are routable on the VPN network only.

End-systems subscribe to the contents of the VPN routing tables for which they have members associated with. This information is then used to populate the host's routing tables. It may contain both host routes (i.e. IPv4 32-bit prefixes or IPv6 128-bit prefixes) or routes to gateways that interconnect other networks.

The signaling network delivers the membership advertisements generated by the end-systems to other members of the same VPN, subjet to policy controls.

When a particular VM "moves" from one physical end-system to another, its respective VPN address will be advertised by the new system and that notification propagated to all attachment points of that VPN.

This document assumes two types of applications that perform network signaling functions: BGP Route Reflectors (RRs) and BGP/XMPP signaling gateways. Both functions may be collocated in the same physical device.

The BGP Route Reflectors accept connection from gateways or native BGP devices. These BGP peering sessions SHALL be support the address families: VPN-IPv4 (1, 128), VPN-IPv6 (2, 128) and RT-Constraint (1, 132) [RFC4684].

The XMPP signaling gateways maintain persistent connection to a subset of the end-systems of the domain and provide a 'pubsub' API to the contents of each specific VPN routing table. These systems are

not in the forwarding plane and do not need to be collocated with a network device.

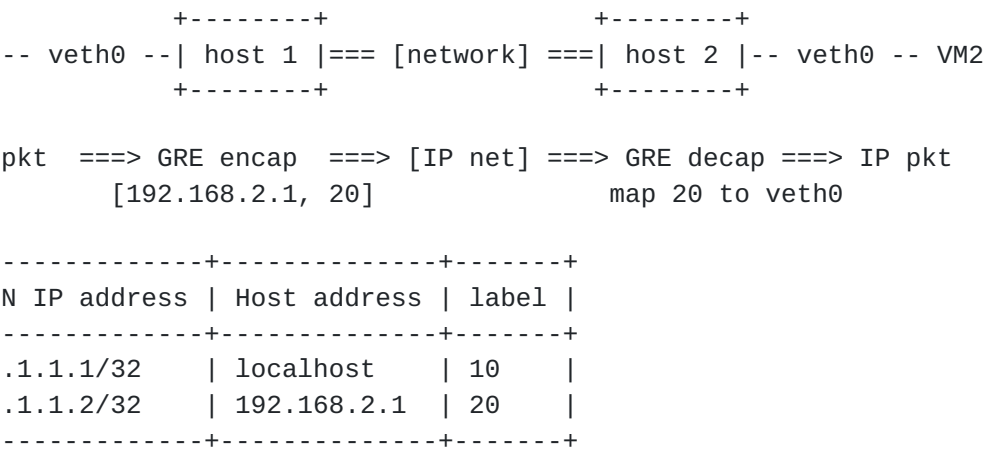
Network devices MAY have direct BGP sessions to the BGP Route Reflectors. For instance, a router or security appliance that supports BGP/MPLS IP VPNs over GRE may use its existing functionality to inter-operate directly with a collection of Virtual Machines.

The BGP/XMPP gateways implement the VRF policy functionality that is associated with PE routers in the pure BGP IP/VPN case. In these signaling gateways, the 'publish-subscribe' messages from the end-systems are associated with a VRF-specific signaling table. Each of these routing tables contains import and export policies which provide fine grain control over the table contents.

An export policy associates VPN routing information with one or more 6 byte values known as 'Route Targets'. These 'Route Targets' are associated with the routes as they are advertised out to other BGP speakers.

Import policies, on the other hand, select via 'Route Targets', from all the available routing information which routes should be imported into a VPN-specific routing table.

A symmetrical VPN uses the same configuration for both import and export. By controlling these policies it is possible to selectively allow direct traffic exchanges between members of different VPNs, assuming their respective IP addresses are non-overlapping.

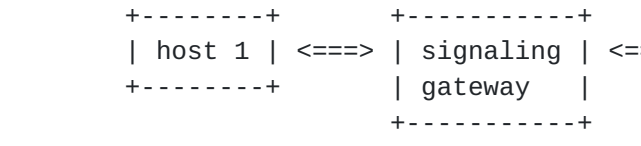


VPN IP address	Host address	label
10.1.1.1/32	localhost	10
10.1.1.2/32	192.168.2.1	20

VRF table on host1

The figure and table above contain an example in which IP packets are transmitted from one VPN interface (with address 10.1.1.1) to another VPN interface (with address 10.1.1.2). As previously mentioned, the virtual ethernet interfaces function as a CE interace in a traditional BGP-signaled IP VPN. While the end-system provide the

forwarding functionality equivalent to a PE device.



VPN IP address	SNPA	label	Known via
10.1.1.1/32	192.168.1.1	10	XMPP
10.1.1.2/32	192.168.2.1	20	BGP

VPN Routing table on signaling gateway

The signaling network corresponding to the same example is depicted above. The signaling gateway is an out-of-band system which speaks both XMPP to the host as well as BGP to the BGP RRs. The table above represents the routing table on the gateway that corresponds to the VPN of the example. Host 2 would be connected to another signaling gateway which would be in turn connected to the BGP RR mesh.

The gateway is configured via an external mechanism with the parameters that correspond to the VPNs in use by its clients along with its respective vrf import and export policies.

XMPP publish request are translated into routing entries on this table, which are then advertised via BGP, using standard BGP-signaled IP VPN mechanism.BGP learned routes are also imported into this routing table. Any changes to its content are advertised to local XMPP clients.

In comparison with traditional IP VPNs, the signaling gateway is performing the PE functionality, with XMPP used as a PE-CE routing protocol.

An example of an asymmetrical VPN configuration is one where all the traffic from VMs must be redirected though a middle-box (on a VM) for inspection. Assuming that the VMs of a particular user are configured to be in the VPN "tenant1" at an initial stage. This "tenant1" VPN is symmetrical and uses a single Route Target in both its import and export policies. The middle-box functionality can be incrementally deployed by defining a new VPN, "tenant1-hub", and an associated Route Target. Accompanied with a change in the gateway configuration such that VPN "tenant1" only imports routes with the Route Target associated with the hub. The "hub" VPN is assumed to advertise a prefix that covers all the VMs IP addresses. The "hub" VPN imports the VMs routes in order for it to be able to generate the

XMPP updates to the "hub" end-system. This information is required for the return traffic from the hub to the spokes (the standard VMs). In such a scenario a single interface can connect the middle-box to the VMs in a given VPN which appear logically as downstream from it. Such a middle-box would often require connectivity to multiple VPNs, such as for instance an "outside" VPN which provides external

connectivity to one or more "inside" VPNs.

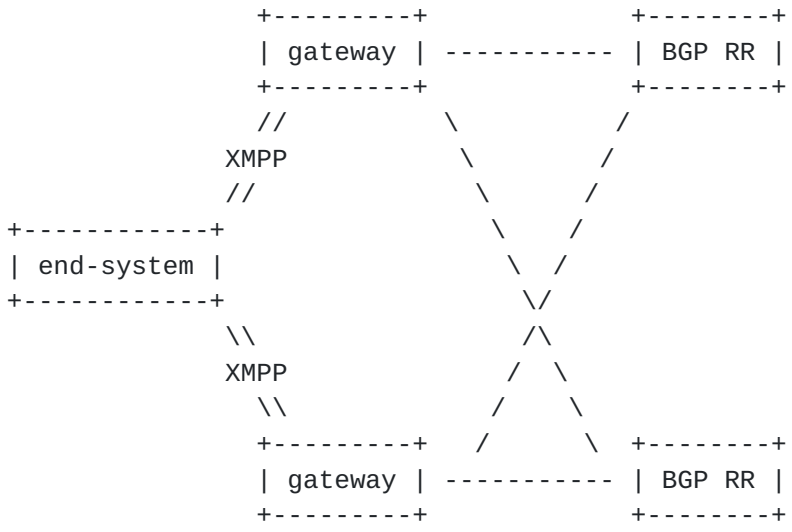
5. XMPP client interface

The communication between end-systems and the signaling gateway uses the XMPP protocol with the PubSub Collection Nodes [pubsub] extension in order to exchange VPN route information.

End-systems establish persistent XMPP sessions. These sessions MUST use the XMPP Ping [xmpp-ping] extension in order to detect end-system failures.

An End-system MAY connect to multiple VPN-signaling gateways for reliability. In this case it SHOULD publish its information to each of the gateways. It MAY choose to subscribe to VPN routing information once only from one of the available gateways.

The information advertised by a end-system SHOULD be deleted after a configurable timeout, when the session closes. This timeout should default to 60 seconds.



The figure above represents a typical configuration in which a end-system is homed to multiple gateways, which are in turn connected to multiple BGP route reflectors. In a deployment there would be a number of gateways corresponding to the number of end-systems divided by the gateway capacity in terms of number of XMPP sessions. While the BGP RR scale in terms of the number of gateways attached to it.

The XMPP "jid" used by the end-system shall be a 6-byte value that uniquely identifies the host in the domain. This specification recommends the use of the 802 MAC address of one of the physical ethernet interfaces of the end-system, when present.

Each VPN shall be identified by a 64 ASCII character string.

The host system software on an end-system SHALL establish an XMPP session with its configured signaling gateways before creating virtual interfaces.

When a virtual interface is created, for instance as result of a Virtual Machine being instantiated on a end-system, the host operating-system software shall generate an XMPP Publish message to the VPN-signaling gateway.

Publish request from end-system to gateway:

```
<iq type='set'
    from='01020304abcd@domain.org'  <!-- system-id@domain.org -->
    to='network-control.domain.org'
    id='request1'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='01020304abcd:vpn-ip-address/32'>
      <item>
        <entry xmlns='http://ietf.org/protocol/bgpvpn'>
          <nlri af='1'>'vpn-ip-address>/32'</nlri>
      <snpa af='1'>'infrastructure-ip-address'</snpa>
          <version id='1'>      <!-- non-decreasing VM version # -->
      <label>1</label>      <!-- 24 bit number -->
        </entry>
       </item>
    </publish>
  </pubsub>
</iq>

<iq type='set'
    from='01020304abcd@domain.org'
    to='network-control.domain.org'
    id='request2'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <collection node='vpn-customer-name'>
      <associate node='01020304abcd:vpn-ip-address/32'/>
    </collection>
  </pubsub>
</iq>
```

In the request above the node 'vpn-customer-name' is assumed to be a collection which is implicitly created by the VPN-signaling gateway.

The VPN-signaling gateway will convert the information received in a the 'publish' request into the corresponding BGP route information such that:.

- It associates the specific request with a local VRF with the name specified in the collection 'node' attribute.

- It creates a BGP VPN route with a 'Route Distinguisher' (RD) which contains the the end-system's 'system-id' value and the specified

IP prefix and 'label' as the Network Layer Reachability Information (NLRI) .

It associates this route with the specified SNPA address.

It associates the route with an extended community TDB containing the version number.

Subscription request from end-system to gateway:

```
<iq type='set'
    from='01020304abcd@domain.org'
    to='network-control.domain.org'
    id='sub1'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <subscribe node='vpn-customer-name'/>
  </pubsub>
</iq>
```

Update notification from gateway to end-system:

```
<message to='system-id@domain.org from='network-control.domain.org>
  <event xmlns='http://jabber.org/protocol/pubsub#event'>
    <items node='vpn-customer-name'>
      <item id='ae890ac52d0df67ed7cfdf51b644e901'>
        <entry xmlns='http://ietf.org/protocol/bgpvpn'>
          <nlri af='1'>'vpn-ip-address>/32'</nlri>
      <snpa af='1'>'infrastructure-ip-address'</snpa>
          <version id='1'>      <!-- non-decreasing VM version # -->
      <label>1</label>      <!-- 24 bit number -->
        </entry>
       </item>
      <item >
        ...
      </item>
    </items>
  </event>
</message>
```

Notifications should be generated whenever a VPN route is added, modified or deleted.

Note that the Update from the signaling gateway to the end-point does not contain the system-id of the destination end-point. When multiple possible routes exist for a given VPN IP address, for instance because the VM may be in the process of moving location, it is the responsibility of the gateway to select the best path to advertise to the end-system.

When routes are withdrawn, the signaling gateway generates both a "collection disassociate" request as well as a node "delete" request.

In situations where an automated system is controlling the instantiation of VMs it may be possible to have that system assign a non-decreasing version number for each instantiation of that particular VM. In that case, this number, carried in the 'version'

field may be used to help gateways select the most recent instantiation of a VM during the interval of time where multiple routes are present.

6. VPN NLRI

When a VPN-signaling gateway receives a request to create or modify a VPN route is SHALL generate a BGP VPN route advertisement with the corresponding information using the BGP address family corresponding to the address family specified by the end-system.

It is assumed that the VPN-signaling gateways contain information regarding the mapping between 'vpn-customer-names' and BGP Route Targets used to import and export information from the associated VRFs. This mapping is known via an out-of-band mechanism not specified in this document.

Whenever a VRF in the gateway contains local routing information, the gateway shall advertise the corresponding RT-Constraint route target routes in BGP, which perform a parallel function to the subscription requests in XMPP.

The 32bit route version number defined in the XML schema is advertised into BGP as a Extended community with type TBD.

Signaling gateways SHOULD use automatically assign a BGP route distinguisher per VPN routing table.

7. Security Considerations

The signaling protocol defines the access control policies for each virtual interface and any VM associated with it. It is important to secure the end-system access to signaling gateways and the BGP infrastructure itself.

The XMPP session between end-systems and the XMPP gateways MUST use mutual authentication. One possible strategy is to distribute pre-signed certificates to end-systems which are presented as proof of authorization to the signaling gateway.

BGP sessions MUST be authenticated using a shared secret. This document recommends that BGP speaking systems filter traffic on port 179 such that only IP addresses which are known to participate in the BGP signaling protocol are allowed.

8. Acknowledgements

Yakov Rekhter provided valuable input as well as helped correct several technical inaccuracies in this document. The authors would also like to thank Thomas Morin for his comments.

9. References

[RFC4023] Worster, T., Rekhter, Y. and E. Rosen, "Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)", RFC 4023,

March 2005.

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

[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, November 2006.

[xmpp-ping] "XMPP Ping", XEP 0199, June 2009.

[pubsub] "PubSub Collection Nodes", XEP 0248, September 2010.

Authors' Addresses

Pedro Marques

Email: pedro.r.marques@gmail.com

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

Email: lufang@cisco.com

Ping Pan
Infinera Corp
140 Caspian Ct.
Sunnyvale, CA 94089

Email: ppan@infinera.com

Amit Shukla
Juniper Networks
1194 N. Mathilda Av.
Sunnyvale, CA 94089

Email: amit@juniper.net