Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 12, 2012

P. Marques
L. Fang
Cisco Systems
P. Pan
Infinera Corp
A. Shukla
Juniper Networks
M. Napierala
AT&T Labs
N. Bitar
Verizon

March 2012

BGP-signaled end-system IP/VPNs.
draft-marques-l3vpn-end-system-05

Abstract

This document describes how the control plane specified by BGP/MPLS IP VPNs [RFC4364] can be used to provide a network virtualization solution for end-systems that meets the requirements of large scale data-centers.

It specifies how the control and forwarding functions of a Provide Edge (PE) device described in [RFC4364] can be separated such that the forwarding function can be implemented in end-systems themselves.

The solution is applicable to any encapsulation that can deliver packets across an IP network as a tunneled IP datagram plus a 20-bit label.

Status of this Memo

This Internet-Draft will expire on September 12, 2012.

Copyright Notice

Table of Contents

1. Introduction

This document describes the requirements for a network virtualization solution that satisfies the needs of large scale data-centers. It then discusses how the BGP IP VPNs [RFC4364] control plane can be used to provide a solution that meets these requirements. Subsequent sections provide a detailed discussion of the control and forwarding plane components.

1.1. Terminoloy

This document makes use of the following terms:

Data-center A logical set of compute, storage and network resources that may spread over multiple physical facilities. Its geographic scope is limited by a communication latency requirement. Typically, it represents an availability zone for applications. It may be the case that the same physical facilities support multiple logical data-centers.

Signaling Gateway A software application that implements the control plane functionality of a BGP IP VPN PE device and a XMPP server that interacts with VPN forwarders.

Virtual Interface An interface in an end-system that is used by a virtual machine or by applications. It performs the role of a CE interface in a BGP IP VPN network.

VPN Forwarder The forwarding component of a BGP IP VPN PE device.

2. Requirements

The main drivers for network virtualization in data-center solutions are network based access-control, multi-tenancy support and VM mobility. Data-centers often have the need to support one or more of these network functions which have often in the past been implemented using VLAN [IEEE.802-1Q] technologies.

The prevalence of L2 based implementations in small scale deployments has created the perception that the network functionality itself requires L2 transparency and is best served by technologies that attempt to drive up the scope and scale of L2 broadcast domains.

This document takes the view that the desired functionality requires a virtualization technology that provides an IP service, with layer 2 topology being irrelevant as long as the service goals are met.

IP-based access control was the first of these functions to be commonly deployed in data-centers. Its goal is to provide a "closed user-group" among a set of end-points, often compute resources dedicated to one application, such that communication within that group occurs unfettered. This was accomplished by placing all these end-points on the same IP subnet. At the same time IP-based access control allows the use of traffic filtering polices to control/ restrict communication between the members of the group and other groups (inter-subnet communication). These "closed user-groups" are often used by IT organizations to both segregate applications as well a separate production, quality-assurance and development environments which often must be contained in different communication domains.

The term "closed user-group" does not imply that there is no communication with external groups. Only that the membership in the group is administratively defined. Each "closed user-group" is a VPN in the terminology used by BGP IP VPNs. Traditionally this functionality has been implemented in data-center designs by using a VLAN between access and aggregation switches and controlling the VLAN-to-VLAN access control policies at the aggregation switches.

This solution works well in an environment where the I/O bandwidth between compute resources is considered to be the resource to optimize for. In this scenario resources associated to a given application are dedicated to it and kept in physical proximity.

Solutions that optimize for the maximum utilization of fungible compute and storage resources need a different approach. They require that the compute resources associated with an application (and thus a "closed user-group"/VPN) may be located anywhere in the data-center. And given that this physical spreading of resources already implies a very significant increase in data-center core bandwidth requirements, they must also make sure that packets only traverse the switching infrastructure once.

If members of a "closed-user group" may be be present anywhere in the data-center, that implies a different operational environment for a VLAN based implementation. Either it would have to operate under the assumption that all switches belong to all VLANs or to be able to dynamically adjust the topology of each VLAN. The later implies that the dimension and scope of each broadcast domain is not know a-priori.

In order to minimize core bandwidth and traffic latency the inter-subnet traffic exchange policies should be pushed as far as possible to the edge. The end-systems that source and receive the traffic are able to implement the virtualization and policy enforcement functionality while using the optimal path for traffic.

With VLAN [IEEE.802-1Q] an ethernet end-point can only be a member of a single "closed user group". Ideally a single application end-point should be able to be a member of multiple "closed user groups". That is possible with L3 based technologies such as BGP IP VPNs.

The second function, multi-tenant support is often combined with network based access control. Each tenant should be able to define multiple "closed user-groups", for instance on a per-application basis, and while "closed user-groups" from multiple tenants are often not allowed to interact directly, tenants are typically allowed to use common infrastructure services (e.g. storage, database services, application-services, etc).

In both of these scenarios, the requirement is to control the IP traffic crossing multiple subnets (where an IP subnet is a "closed user group") such that it conforms to the defined traffic policies.

The third function to be considered is the need to support "VM life-migration". This functionality requires a virtual IP topology by itself even in data-centers where network based access control or multi-tenancy are not a concern. Life-migration requires virtual machines to be moved from a physical server A to a physical server B such that the total migration time does not disrupt its communication sessions. This implies that transport sessions must answer keepalives before the user-specified timeout (often a few seconds). In some cases, the operation may also be constrained by the application latency requirements which are typically in the sub-second range. While the main bottleneck in this process is the process of saving and restoring the VM's modified memory pages, the connectivity restoration time is critical.

Support for minimizing connectivity interruption and minimum latency impact would benefit from the ability for the same IP end-point to be received at the new physical location of the VM (B) as well as the ability to tunnel traffic from the old location (A) during the convergence period. It also requires a control plane that can minimize convergence time and that can decouple the instantiation of the end-point (i.e. the VM IP address becoming active) from its advertisement to the network as the preferred route to that IP address.

It is important to note that, as with the previous network functions, we need to consider IP transport sessions with system both in the same "closed user group" (or subnet) as the VM as well as in different "closed user groups" (i.e. subnets). Any solution to this problem must be able to minimize connectivity restoration time across different IP subnets.

3. Applicability of BGP IP VPNs

BGP IP VPNs [RFC4364] is the industry de-facto standard for providing "closed user group" functionality in WAN environments. It is used by service providers in environments where several millions of routes are present. It supports both isolated VPNs as well as overlapping VPNs (often referred to as "extranets").

In its traditional usage in Service Provider networks, BGP IP VPN functionality is implemented in a Provider Edge (PE) device that combines both BGP signaling as well a VRF-based forwarding functions. In practice, most PE devices in current use are multi-component systems with the signaling and forwarding functionality actually implemented in different processors attached to an internal network.

This document assumes a similar separation of functionality in which signaling devices implement the control plane functionality of a PE device and a VPN forwarder (in the hypervisor/host OS or first-hop switch) implements the forwarding function usually found in a PE device "line-card".

Operationally, BGP IP VPN technology has several important characteristics:

It has a high-level of aggregation between customer interfaces and managed entities (Provider Edge devices).

It defines VPNs as policies, allowing an interface to be a member of multiple VPNs and allowing for the topology of the virtual network to be modified by modifying the policy configuration.

It scales horizontally in terms of event propagation. By increasing the number of signaling devices, implementing the PE control plane, it is possible to decrease the load on each signaling device when it comes to propagating events that originate in a specific location and must be propagated across the network.

The last point is particularly relevant to the convergence characteristics required for large scale deployments. BGP's hierarchical route distribution capabilities allow a deployment to divide the workload by increasing the number of BGP signaling gateways.

As an example consider a topology in which 100 BGP signaling gateways are deployed in a data-center each serving a subset of the VPN forwarding elements. The signaling gateways inter-connect to two top-level BGP Route Reflectors [RFC4456].

If an event (i.e. a VPN route change) needs to be propagated from a specific machine that activates a VM to 10.000 clients randomly distributed across a data-center, each of the BGP signaling gateways must generate 100 updates to its respective downstream clients.

By modifying this topology such that another 100 signaling gateways are added, then each signaling gateway is now responsible to generate 50 client updates. This example illustrates the linear scaling properties of BGP: doubling the number of signaling gateways (i.e. the processing capacity) reduces in half the number of updates generated by each (i.e. load at each processing node).

The same horizontal scaling techniques can be applied to the Route Reflector layer in the example above by subsetting the VPN Route space according to some pre-defined criteria (for instance VPN route target) and using a pair of Route Reflectors per subset.

In the example above we assumed a dense membership in which all signaling gateways have local clients that are interested in a particular event. BGP also optimizes the route distribution for sparse events. The Route Target Constraint [RFC4684] extension, builds an optimal distribution tree for message propagation based on VPN membership. It ensures that only the Signaling Gateways with local receivers for a particular event do receive it also decreasing the total load on the upstream BGP speaker.

In the WAN environment, BGP IP VPN control plane scaling is focused not primarily on route convergence times but on memory footprint of embedded devices. While memory footprint does not have a similar linear scaling behavior, memory technology in the data-center is often at 10x the scale of what is commonly found in WAN environments.

The functionality present in the BGP IP VPN control plane addresses the requirements specified in the previous section. Specifically, it supports multiple potentially overlapping "groups", regular or "hub and spoke" topologies and the scaling characteristics necessary.

The BGP IP VPN control plane supports not only the definition of "closed user-groups" (VPNs in its terminology) but also the propagation of inter-VPN traffic policies [RFC5575]. An application of that mechanism to "end-system" VPN is presented in [I-D.marques-sdnp-flow-spec].

Note that the signaling protocol itself is rather agnostic of the encapsulation used on the wire as long as this encapsulation has the ability to carry a 20 bit label.

Several data-center deployments use a switching infrastructure that is only capable of providing an IP unicast service. In order to support them, implementations of this document should support the MPLS in GRE [RFC4023] encapsulation. Other encapsulations are possible, including UDP based encapsulations.

4. Virtual network end-points

This document assumes that end-systems support one or more virtual network interfaces in addition to a physical interface that is associated with the data-center switching infrastructure. Virtual network interfaces can be associated with a VM or they can be used to provide network connectivity directly to applications in the same way that a "VPN tunnel" interface is used to provide access between an end-system (e.g. a laptop) and a remote corporate network.

From an IP address assignment point of view, a virtual network interface is addressed out of the virtual IP topology and associated with a "closed user group" or VPN, while the physical interface of the machine is addressed in the network infrastructure topology. As a security measure, it is recommended that virtual and infrastructure topologies never be allowed to exchange traffic directly.

In data-center environments, static IP address allocation is common since it is desirable to associate a permanent IP address to a VM and have that IP address remain constant even as the VM migrates. However dynamic address assignment through DHCP is also possible assuming that the VPN forwarder implements DHCP relay functionality.

A virtual network interface is connected to a VPN forwarder. This VPN forwarder MAY be located on the hypervisor or host OS that co-resides on the same physical machine or it could be located in an external system, such as the first-hop switch. We refer to this second system as the external VPN forwarder.

All traffic that ingresses or egresses through this virtual network interface is routed at the VPN forwarder which acts as the first-hop router (in the virtual topology). The IP configuration on the client side of this virtual network interface (e.g. in the guest OS) can follow one of two models:

point-to-point interface model.

multipoint interface model.

In a point-to-point interface model, the system routing table (e.g. the guest OS) contains the following routing entires: a host route to the local IP address, a host route to the first-hop router via the virtual interface and a default route to the first-hop router. This is the model typically used in "VPN tunnel" configurations or other access technologies such as cable deployments or DSL. When this model is used, the first-hop router IP address is a link-local address that is the same on all first-hop routers across a specific deployment. This first-hop IP address should not change when a VM migrates between different machines.

In a multi-point interface model, the system routing table (on the guest OS) contains the following routing entires: a host route to the local IP address, a subnet route to the local interface and optionally a default route to an specific router address within that subnet. In this model, the end-system will issue address resolution requests for any IP addresses it considers to be directly attached to the subnet. The VPN forwarder shall answer all address resolution requests with a virtual MAC address which SHOULD be the same across all VPN forwarders in a specific deployment. This virtual MAC address SHALL default to the VRRP [RFC5798] virtual router MAC address for VRID 1..

When the virtual topology first-hop router resides on the same physical machine, the host OS is responsible to map the virtual interface with a VPN specific routing table (without taking L2 addresses into consideration). In this case the mac-addresses known to the guest OS are not used on the wire.

When the virtual topology first-hop router resides in an external system (e.g. the first hop-switch) the virtual interface shall be identified by the combination of the physical interface mac-address and a 802.1Q VLAN tag. The first-hop switch should use a virtual router MAC address to answer any address resolution queries.

Whenever an external VPN forwarder is used and resiliency is desired, the external VPN forwarder should be redundant. It is desirable to use VRRP as a mechanism to control the flow of traffic between the end-system and the external VPN forwarder. VRRP already defines the necessary procedures to elect a single forwarder for a LAN.

This specification uses the VRRP virtual router MAC address as the default L2 address for the VPN forwarder as a VM may move between locations where redundancy is and is not present.

While the VRRP Virtual Router MAC will be used resolve any address resolution request made by the virtual interface client (e.g. the guest VM) this does not imply that a single default router is elected per virtual IP subnet. The ingress VPN forwarder will perform an IP forwarding decision based on the destination IP address of the (payload) traffic.

VRRP router election is only relevant in selecting the VPN forwarder associated with a specific machine, when external forwarders are in use.

5. VPN forwarder

In this solution, the Host OS/Hypervisor in the end-system must participate in the virtual network service. Given an end-system with multiple virtual interfaces, these virtual interfaces must be mapped onto the network by the guest OS such that applications on one virtual interface are not allowed to impersonate another virtual interface.

When VPN forwarder functionality is implemented by the Host OS/ Hypervisor, intermediate systems in the network, be they access, aggregation or core switches, do not require any knowledge of the virtual network topology. This can simplify the design an operation of the data-center switching infrastructure.

When it is not possible or desirable to add the VPN forwarding functionality to the end-system, it may be implemented by an external system, typically located as close as possible to the end-system itself. This could be a top-of-rack or aggregation switch.

Both models, end-system and extern VPN forwarder can co-exist in a deployment.

In order to implement the BGP IP VPN forwarder functionality a device MUST implement the following functionality:

Support for multiple "Virtual Routing and Forwarding" (VRF) tables;

VRF route entries map prefixes in the virtual network topology to a next-hop containing a infrastructure IP address and a 20-bit label allocated by the destination forwarder. The VRF table lookup follows the standard IP lookup (best-match) algorithm.

Associate a end-system virtual interface with a specific VRF table;

When the first-hop router is the Host OS/hypervisor this association is performed by an internal mechanism. When the first-hop router is external this association is performed using the mac-address of the end-system and a IEEE 802.1Q tag that identifies the virtual interface within the end-system.

Encapsulate outgoing traffic (end-system to network) according to the result of the VRF lookup;

Associate incoming packets (network to end-system) to a VRF according to the 20-bit label contained immediately after the GRE header;

The VPN forwarder MAY support the ability to associate multiple virtual interfaces with the same VRF. When that is the case, locally originated 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 other VPN forwarders, 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 VPN forwarder.

When an external VPN forwarder is used the end-system MUST associate each virtual interface with a VLAN [IEEE.802-1Q] that is unique on the end-system. The switching infrastructure MUST be configured such that multi-destination frames sourced from an end-system are only delivered to VPN forwarders used by this end-system and not to other end-systems.

6. XMPP signaling protocol

Signaling Gateways must be made aware of virtual interface creation and deletion and of when IP addresses are added or removed from these virtual interfaces.

VPN forwarders must receive VPN route information from which to populate their forwarding tables. When the VPN forwarder is a different system than the end-system where the virtual interface resides, it also needs to receive the interface and IP address events from the end-system. In this case it is the VPN forwarder that propagates these to the Signaling Gateway.

When an external VPN forwarder is used, the end-system assigns the VLAN identifier used for each virtual interface. This information is conveyed by an optional parameter in the VPN subscription request. It is not propagated by the VPN forwarder.

In order to exchange this information this specification uses the XMPP [RFC6120] protocol along with the PubSub Collection Nodes [pubsub] extension.

Clients (end-systems and external VPN forwarders) establish persistent XMPP sessions. These sessions MUST use the XMPP Ping [xmpp-ping] extension in order to detect end-system failures.

A client MAY connect to multiple servers (e.g. 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 client SHOULD be deleted after a configurable timeout, when the session closes. This timeout should default to 60 seconds.

```
                   +---------+             +--------+
                   | gateway | ----------- |  BGP   |
                   +---------+             +--------+
                   //          \          /
                 XMPP           \        /
                 //              \      /
   +------------+                 \    /
   | end-system |                  \  /
   +------------+                   \/
                 \\                 /\
                 XMPP              /  \
                   \\             /    \
                   +---------+   /      \  +--------+
                   | gateway | ----------- |  BGP   |
                   +---------+             +--------+
```

The figure above represents a typical configuration in which an end-system (implementing the VPN forwarder functionality) is directly connected to two gateways, which are in turn connected to multiple BGP spakers which may be other BGP signaling gateways or BGP route reflectors.

In deployment the number of gateways used will depend on the desired gateway to VPN forwarder ratio which affects the convergence time of event propagation.

The XMPP "jid" used by the client shall be a 6-byte value that uniquely identifies it in the domain. This specification recommends the use of the MAC address of one of the physical ethernet interfaces.

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

When external forwarders are used, its control software operates as a XMPP server processing requests from end-systems and as a client of one or more Signaling Gateways. The control software relays to the

Signaling Gateways(s) messages it receives from the end-system. VPN routing information received from the Signaling Gateways(s) SHOULD NOT be propagated to the end-system.

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 Subscribe message to its server (the VPN-signaling gateway or external VPN forwarder).

Subscription request from end-system to gateway (local VPN forwarder):

```
<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>
```

The request above, instructs the signaling gateway to start populating the client's VRF table with any routing information that is available for this VPN. The XMPP node 'vpn-customer-name' is assumed to be a collection which is implicitly created by the VPN-signaling gateway. Creation of a virtual interface may precede any IP address becoming active on the interface, as it is the case with VM life migration.

Subscription request from end-system to external VPN forwarder:

```
<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'/>
    <options>
      <x xmlns='jabber:x:data' type='submit'>
        <field var='vpn#vlan_id'><value>vlan-id</value></field>
      </x>
    </options>
  </pubsub>
</iq>
```

When an external VPN forwarder is used the end-system should include the VLAN identifier it assigned to the virtual interface as a subscription option.

When a IP address is added to a virtual interface, the end-system will generate an XMPP Publish request.

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>
         <next-hop af='1'>'infrastructure-ip-address'</next-hop>
             <version id='1'>      <!-- non-decreasing VM version # -->
         <label>10000</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>
```

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 which it resolves by using a combination of the originator system-id and 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) .

The BGP next-hop address is set to the address of egress VPN forwarder.

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

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>
      <next-hop af='1'>'infrastructure-ip-address'</next-hop>
          <version id='1'>      <!-- non-decreasing VM version # -->
      <label>10000</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.

7. Signaling gateway behavior

BGP Signaling gateways SHALL support the address families: VPN-IPv4 (1, 128), VPN-IPv6 (2, 128) and RT-Constraint (1, 132) [RFC4684].

When a VPN-signaling gateway receives a request to create or modify a VPN route it SHALL generate a BGP VPN route advertisement with the corresponding information.

It is assumed that the VPN-signaling gateways contain information regarding the mapping between end-system the tuple ('system-id', '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 the Signaling Gateway receives an XMPP subscription request, it SHALL consult its RT-Constraint Routing Information Base (RIB). If the Signaling Gateway does not already have locally originated route that corresponds to the route target being carried in the request, it SHALL create one and generate the corresponding BGP route advertisement. This route advertisement should only be withdrawn when there are no more downstream XMPP clients subscribed to the VPN.

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

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

8. Operational Model

In the simplest case, a VPN is a collection of systems that are allowed to exchange traffic with each other and only with each other. Since all the forwarding tables in this VPN have the same routing entires they are often referred to as symmetrical VPNs.

In order to better illustrate the operation of the protocol we consider a simple example in which "host 1" and "host 2" both contain a VM that is a member of the same VPN.

Each of these hosts has an XMPP session with a Signaling Gateway, SG1 and SG2 our example, and these Signaling Gateways are part of the same BGP mesh.

When a virtual interface is created on "host 1", the local XMPP client generates a XMPP subscription message to its respective Signaling Gateway. This message contains a VPN identifier that has been assigned by the VM provisioning system. The Signaling Gateway maps that identifier to a BGP IP VPN configuration which contains the list of import and export route targets to be used for that particular VRF.

Once the VM is operational, "host 1" will publish any IP addresses that are configured on the respective virtual interface. This will in turn cause the Signaling Gateway to advertise these to any other BGP speaker on the network which is connected to an attachement point of that VPN.

```
+---------+        +------------+        +-----------+
| host 1 | <===> | signaling | <===> | BGP mesh |
+---------+        | gateway   |        +-----------+
                   +------------+

+-----------------+-------------+--------+-----------+
```

VPN IP address	NEXT-HOP	label	Known via
10.1.1.1/32	192.168.1.1	10000	XMPP
10.1.1.2/32	192.168.2.1	20000	BGP

VPN Routing table on signaling gateway

The figure above represents the contents of the VRF routing table on Signaling Gateway 1 after the IPv4 address 10.1.1.1 has been added to the virtual interface on host 1. It assumes that there is another attachement point for this VPN with the IPv4 address of 10.1.1.2. Host 1 has an infrastructure IP address of 192.168.1.1 configured on its physical interface while host 2 has IP address 192.168.2.1.

The contents of the VRF routing table in the Signaling Gateway are advertised via XMPP Update notifications sent to host 1. This information is the used by the host to populate the forwarding table associated with that VPN.

```
               +--------+                 +--------+
VM1 -- veth0 --| host 1 |=== [network] ===| host 2 |-- veth0 -- VM2
               +--------+                 +--------+

 IP pkt  ===> GRE encap  ===> [IP net] ===> GRE decap ===> IP pkt
           [192.168.2.1, 20]               map 20 to veth0
```

VPN IP address	Host address	label
10.1.1.1/32	localhost	10000
10.1.1.2/32	192.168.2.1	20000

VRF table on host1

When the VM on host 1 generates packets with a destination IP address of 10.1.1.2 these are routed by the VPN forwarder implemented in the Host OS. The packets are encapsulated with a GRE header that contains a 20-bit label assigned by host 2.

When the VM on host 1 sends packets on the virtual interface it is using the Virtual Router MAC address as the destination MAC. When host 2 delivers packets to the remote VM it sets the Virtual Router MAC address as the source MAC address. This MAC address is not present on the GRE encapsulated packet.

BGP/XMPP Signaling Gateways are software applications the implement both the BGP IP VPN PE control plane as well as XMPP server functionality. These application are not in the forwarding plane and do not need to be co-located with a network device.

Network devices MAY have direct BGP sessions to the Signaling Gateways. 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.

Signaling Gateways implement the VRF import policy and export policy functionality that is associated with PE routers in standard BGP IP/VPN deployments. VPN forwarders receive forwarding information after policy and route selection is applied. These are unqualified routes in a specific VRF rather than VPN routing information qualified by a Route Distinguisher and with a set of Route Targets.

A symmetrical VPN uses a vrf import and vrf export polices that contain a single route target, where the route target used for both import and export is the same.

Different VPN topologies can be created by manipulating the vrf import and export configuration including "hub-and-spoke" topologies or overlapping VPNs.

An example of a hub-and-spoke 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 Signaling 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.

The functionality defined in this document in which the BGP IP VPN PE functionality is split into its control (Signaling Gateway) and forwarding (VPN forwarder) components is fully interoperable with existing BGP IP VPN PEs.

This makes it possible to reuse existing systems. For example, at the edge of a data-center facility it may be desirable to use an existing router or appliance that aggregates IP VPN routing information and/or provides IP based services such as stateful packet inspection.

Such a system can be configured, based on existing functionality, to suppress more specific routes than a specified aggregate while advertising the aggregate with a BGP NEXT_HOP containing the PE's IP address and a locally assigned label corresponding to a VRF where the more specific routes are present.

9. 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.

10. Acknowledgements

Yakov Rekhter has contributed to this document by providing detailed feedback and suggestions. The authors would also like to thank Thomas Morin for his comments.

11. References

11.1. Normative References

[RFC4023] Worster, T., Rekhter, Y. and E. Rosen, "Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)", RFC 4023, March 2005.

[RFC4271] Rekhter, Y., Li, T. and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

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

[RFC4456] Bates, T., Chen, E. and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, April 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.

[RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J. and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, August 2009.

[RFC5798] Nadas, S., "Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6", RFC 5798, March 2010.

[RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, March 2011.

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

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

11.2. Informational References

[I-D.marques-sdnp-flow-spec]
Marques, P, Fang, L, Pan, P and A Shukla, "Traffic classification, filtering and redirection for end-system IP VPNs.", Internet-Draft draft-marques-sdnp-flow-spec-00, October 2011.

[IEEE.802-1Q]
Institute of Electrical and Electronics Engineers, "Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks", IEEE Std 802.1Q-2005, May 2006.

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

Maria Napierala
AT&T Labs
200 Laurel Avenue
Middletown, NJ 07748

Email: mnapierala@att.com

Nabil Bitar
Verizon
40 Sylvan Rd.
Waltham, MA 02145

Email: nabil.bitar@verizon.com