Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 18, 2017

S. Mackie
Juniper Networks
L. Fang
eBay
N. Sheth
Juniper Networks
M. Napierala
AT&T Labs
N. Bitar
Nokia
December 15, 2016

BGP-Signaled End-System IP/VPNs
draft-ietf-l3vpn-end-system-06

Abstract

This document describes a solution in which the control plane protocol specified in BGP/MPLS IP VPNs is used and extended via the XMPP protocol to provide a Virtual Network service to end-systems (hosts). These end-systems may be used to provide network services or may host end-user applications.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document describes the requirements for a network virtualization solution that provides an IP service to end-system virtual interfaces. It then discusses how the control plane for BGP IP VPNs [RFC4364] can be used and extended via the XMPP protocol to provide a solution that meets these requirements. Subsequent sections provide a detailed discussion of the control and forwarding plane components.

In BGP IP VPNs, Customer Edge (CE) interfaces connect to a Provider Edge (PE) device which provides both the control plane and VPN encapsulation functions required to implement a Virtual Network service. This document describes how the control plane and forwarding functionality of a PE device can be decoupled in order to enable the forwarding functionality to be implemented in multiple devices. For instance, the forwarding function can be implemented directly on the operating system of application servers or network appliances.

1.1. Terminology

This document makes use of the following terms:

End-System: A compute node whose primary function is to run applications. It is assumed that end-systems support multiple application instances (e.g., virtual machines), each with its independent network configuration.

End-System Route Server: A software application that implements the control plane functionality of a BGP IP VPN PE device and an 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. This is similar to the concept of Virtual Access Point (VAP) in RFC 7365 [RFC7365].

VPN Forwarder: The forwarding component of a BGP IP VPN PE device. This functionality may be co-located with the virtual interface or implemented by an external device.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Requirements

Network virtualization is used in both service provider as well as enterprise networks to support multi-tenancy and network-based access control. It may also be used to facilitate application instance mobility.

Multi-tenancy allows a physical network to provide services to multiple "customers" or "tenants", whether these are external entities in the case of a Service Provider providing managed VPN services, or internal departments of an enterprise sharing an IT facility. Multi-tenancy requires isolation of traffic and routing information between tenants.

Within a tenant, it is often required to create multiple distinct virtual networks, in order to be able to provide network-based access control. In this service model, each virtual network behaves as a "Closed User Group" (CUG) of virtual interfaces that are allowed to exchange traffic freely, while traffic between virtual networks is subject to access controls. This scenario can be found in enterprise campus networks, branch offices and data centers.

It is often the case when network access control is used, that the traffic patterns are such that there is significantly more traffic crossing a CUG boundary than staying within such boundary. As an example, in campus networks it is common to segregate users into CUGs based on some classification such as the user's department. Campus networks often see traffic patterns in which almost all the traffic flows northbound to the data center or internet boundaries. Similar traffic patterns can be found in multi-tier applications in IT data centers.

Virtual interfaces are often configured to expect the concept of IP subnet to match its closed user group. A network virtualization solution should be able to provide this concept of IP subnet regardless of whether the underlying implementation uses a multi-access network or not.

Virtual interfaces should be able to directly access multiple closed user groups without needing to traverse a gateway. Network access policy should allow this access whether the source and destination CUGs for a particular traffic flow belong to the same tenant or different tenants. It is often the case that infrastructure services are provided to multiple tenants. One such example is voice-over-IP gateway services for branch offices.

Independently, but often associated with the previous two functions, IP mobility is another network function that can be implemented using network virtualization. By abstracting the externally visible network address from the underlying infrastructure address, mobility can be implemented without having to rely upon home agents or large L2 broadcast domains.

IP Mobility requires the ability to "move" a virtual interface without disrupting its TCP (or UDP) transport sessions. This requires a mechanism that can efficiently communicate the mappings between logical and physical addressing.

IP Mobility can be a result of devices physically moving (e.g., a WiFi enabled laptop) or workload being diverted between physical systems such as network appliances or application servers.

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").

The BGP IP VPN control plane has been designed to be able to distribute the mapping between virtual address and location (next-hop) to the subset of network nodes for which this information is relevant, whenever that mapping changes. This provides an efficient mechanism to address IP mobility requirements as compared to methods that depend on a (cached) mapping request from the end-systems.

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 as 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 software appliances, the End-System Route Servers, implement the control plane functionality of a PE device and a VPN Forwarder implements the forwarding function usually found in a PE device "line-card". The VPN Forwarder functionality may be co-located with the end-system (e.g., implemented in the hypervisor switch or host OS network drivers) or it may be external. For instance, residing in a data center switch or specialized appliance.

Operationally, BGP IP VPN technology has several important characteristics:

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

o It defines VPNs as policies, allowing an interface to directly exchange traffic with multiple VPNs and allowing for the topology of the virtual network to be modified by modifying the policy configuration.

o 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 for events that originate in a specific location and which 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 End-System Route Servers.

As an example consider a topology in which 100 End-System Route Servers are deployed in a network each serving a subset of the VPN forwarding elements. The Route Servers 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 end-system to 10,000 clients randomly distributed across the network, each of the End-System Route Servers must generate 100 updates to its respective downstream clients.

By modifying this topology such that another 100 End-System Route Servers are added, each Route Server is now responsible for generating 50 client updates. This example illustrates the linear scaling properties of BGP: doubling the number of Route Servers (i.e., the processing capacity) reduces by half the number of updates generated by each one (i.e. the load at each processing node is halved).

The same horizontal scaling techniques can be applied to the Route Reflector layer in the example above by dividing 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 previous example we assumed a dense membership in which all Route Servers 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 XMPP stanza and message propagation based on VPN membership. It ensures that only the PEs 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 not primarily focused on route convergence times, but on the memory footprint of embedded devices. While memory footprint does not have a similar linear scaling behavior as load, memory technology available to software appliances is often at 10x the scale of what is commonly found in WAN environments, and so is not so much of a concern.

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].

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 label of sufficient length to enumerate all the VPNs in an administrative domain (e.g. an MPLS label, which has 20 bits).

Several network environments use a network 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 RFC 7510 [RFC7510] and VXLAN [RFC7348].

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 underlying network infrastructure. A virtual network interfaces can be associated with a specific application via a OS-dependent mechanisms like a Virtual Machine (VM), or they can be used to provide network connectivity to all user 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.

Each virtual network interface is assigned an IP addresses from a subnet associated with a "closed user group" or VPN, while the physical interface of the machine is addressed in the network infrastructure topology.

A virtual network interface is connected to a VPN Forwarder. This VPN Forwarder MAY be co-located in the end-system or external. In cases where the VPN Forwarder is external to the end-system, they can either be directly connected or interconnected with a dedicated 802.1Q VLAN on a per virtual interface basis.

Both static and dynamic IP address allocation can be supported. The latter assumes that the VPN Forwarder implements DHCP relay or DHCP proxy functionality.

Traffic that ingresses or egresses through a 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:

o Point-to-point interface model

o Multipoint interface model

In a point-to-point interface model, the VPN client routing table (e.g., on the guest OS) contains the following routing entries: 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 either an address from the tenant's IP address space or a link-local address. This address SHOULD be the same on all first-hop routers across a specific deployment so that it does not change when a virtual interface moves between end systems.

In a multi-point interface model, the VPN client routing table (e.g., on the guest OS) contains the following routing entries: a host route to the local IP address, a subnet route to the local interface and optionally a default route to a specific router address within that subnet. In this model, the VPN client IP stack 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 via Proxy ARP [RFC1027].The same technique is applicable when Neighbor Discovery is used to resolve IPv6 addresses. Address resolution request SHOULD be answered using 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 Virtual Router Identifier (VRID) 1.

When the virtual topology first-hop router resides on the same physical machine, the host OS is responsible for mapping 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 physical interface of the end-system and a 802.1Q VLAN tag. The first-hop switch should use a virtual router MAC address to answer any address resolution queries.

Whenever external VPN forwarding is used, and resiliency is desired, multiple external VPN Forwarder may be employed in a redundant configuration. 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, in order to support a client virtual interface moving between locations.

While the VRRP Virtual Router MAC will be used to answer 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 end system OS such that applications on one virtual interface cannot send traffic to networks they are not authorized to communicate with or using source addresses not assigned to the virtual interface.

When VPN forwarder functionality is implemented by the Host OS/ Hypervisor, intermediate systems in the network do not require any knowledge of the virtual network topology. This can simplify the design and operation of the physical network.

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.

Both models, co-located and external 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:

o 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 label allocated by the destination Forwarder. The VRF table lookup follows the standard IP lookup (best-match) algorithm.

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

   When the Forwarder is co-located with the end-system, this association is implemented by an internal mechanism. When the Forwarder is external the association is performed using the MAC address of the end-system and an IEEE 802.1Q tag that identifies the virtual interface within the end-system.

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

o Associate incoming packets (network to end-system) to a virtual interface for direct forwarding, or to a VRF for lookup, according to the label contained in the packet;

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. This is intended to ensure that the forwarding behavior is the same whether the VRF is shared or between multiple interfaces of the same virtual-network or not.

As an example, if a given VRF contains two virtual interfaces, "veth0" and "veth1", with the addresses 203.0.113.1/32 and 203.0.113.2/32 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 Route Server such that 203.0.113.2/32 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. This means that forwarding table entries apply to downstream traffic only, by default. This capability can be used to implement a hub-and-spoke topology, if required.

The 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 SHOULD 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

End-System Route Servers must be aware of VPN membership on each Forwarder as well as what IP addresses are currently associated with each virtual interface.

VPN Forwarders receive VPN route information from which to populate their forwarding tables. External VPN Forwarders also need to receive the virtual interface and IP address allocation events for the end-system for which they are VPN forwarders. In this case, the end-system assigns an 802.1Q VLAN tag to each virtual interface and communicates that information to the Forwarder directly, or via the Route Server.

In order to exchange this information this specification uses the XMPP [RFC6120] protocol along with the Publish-Subscribe [pubsub] extension.

VPN forwarders (both co-located and external) establish XMPP sessions with End-System Route Servers, acting as XMPP clients. When an external VPN Forwarder is used, end-systems MAY establish XMPP sessions with VPN Forwarders. In such cases, external VPN Forwarders act as XMPP servers for end-systems which are associated with them.

A VPN Forwarder MAY connect to multiple End-System Route Servers for reliability. In this case it SHOULD publish its information to each of the Route Servers. It MAY choose to subscribe to VPN routing information from only one of the available Route Servers. In this case, the Forwarder is responsible for switching subscriptions over to an alternate Route Server in the case of Route Server failure. Alternatively, it MAY choose to subscribe to VPN routing information from more than one End-System Route Server. In this case, the Forwarder is responsible for selecting which Route Server is

authoritative for each forwarding entry. The Route Servers SHOULD produce the same forwarding information for each destination. The VPN Forwarder is expected to select the entry that it deems as more recent for positive updates. It SHOULD NOT consider a forwarding entry to be withdrawn unless it is withdrawn by both Route Servers.

Each End-System Route Server MUST monitor the XMPP connection status of each VPN Forwarder that is connected to it. The information advertised by an XMPP client SHOULD be deleted after a configurable timeout, after XMPP session closes. This timeout SHOULD default to 60 seconds.

An End-System Route Server MAY monitor the status of each VPN Forwarder that is connected to it, using, for example, the BFD [RFC5880] protocol and to delete advertised information after a timeout when a failure is detected. The Route Server MAY choose to immediately reduce the preference of routing information received from an XMPP client for which a failure has been detected, either through an XMPP session close event, or a failure detection mechanism such as BFD.

```
                        +---------+        +--------+
                        |    RS   |--------|  BGP   |
                        +---------+        +--------+
                        /         \        /
                      XMPP         \      /
                      /             \    /
   +--------------------+            \  /
   |  End   |   VPN     |             \/
   | System | Forwarder |             /\
   +--------------------+            /  \
                      \             /    \
                      XMPP         /      \
                        \         /        \
                        +---------+        +--------+
                        |    RS   |--------|  BGP   |
                        +---------+        +--------+
```

VPN Forwarder Connected to Two Routing Systems

Figure 1

The figure above represents a typical configuration in which an end-system with a co-located VPN Forwarder is directly connected to two End-System Route Servers, which are in turn connected to multiple BGP speakers which may be other L3VPN PEs or BGP route reflectors.

In deployment, the number of End-System Route Servers used will depend on the desired Route Server to VPN Forwarder ratio which affects the convergence time of event propagation.

The XMPP JID used by the client SHALL be a RFC 7622 [RFC7622] compliant address that uniquely identifies it in its administrative domain. The VPN Forwarder SHOULD use its hostname as JID, when available, or a unique IP address within the infrastructure network using its string representation. The same naming convention SHOULD be used for an End System which has an XMPP session with an external VPN Forwarder.

The XMPP JID used by an End-System Route Server SHOULD be the constant string 'route-server@ietf.org'.

Each VPN shall be identified by an ASCII character string that SHOULD NOT exceed 128 octets and MUST be unique within each administrative domain. The VPN identifier is an attribute of each virtual interface. It is assumed that a configuration management system exists such that it provisions the Route Servers with VPN identifier values and the VPN Forwarders with the mapping of virtual interface to VPN identifier. Such a configuration management system is outside the scope of this document.

Each VPN identifier corresponds to a Pub-Sub node in the Route Server XMPP servers. This Pub-Sub nodes SHOULD be configured such that Pub-Sub items are persistent and that event notifications include the item payload. Implementations MAY choose to perform this operation explicitly or implicitly by mapping XMPP subscription requests to an event observer mechanism that tracks the VRF table corresponding to the VPN in question.

When an external Forwarder is used, its control software MAY operate as an XMPP server which processes requests from end-systems and SHALL operate as a client of one or more End-System Route Servers. The control software relays to the End-System Route Server(s) VPN membership stanzas it receives from the end-system. VPN routing information received from the Route Server(s) SHOULD NOT be propagated to the end-system unless it specifically requests such information. End systems MAY have sessions directly with the End-System Route Servers, and in this case no XMPP sessions are required with VPN Forwarders.

When a virtual interface is created on an end-system, the host End System XMPP client SHALL generate an XMPP Subscribe stanza to its server (a Route Server or the external VPN Forwarder).

Each Subscribe stanza SHALL be addressed to the JID of the Route Server (e.g. route-server@ietf.org), using the VPN Identifier as the NodeID.

If subsequent Virtual Interfaces are created with the same VPN Identifier, and the previous Pub-Sub subscription is still in effect, then additional XMPP Pub-Sub Subscribe stanzas SHOULD NOT be sent to the End-System Route Server.

Example subscription request from co-located VPN Forwarder to Route Server:

```
<iq type='set'
    from='forwarder@domain.org'
    to='route-server@ietf.org'
    id='sub1'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <subscribe node='vpn-customer-name' jid='fowarder@domain.org'/>
    <options>
      <instance-id>1</instance-id>
    </options>
  </pubsub>
</iq>
```

The above request instructs the End-System Route Server 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 implicitly created by the End-System Route Server. Creation of a virtual interface may precede any IP address becoming active on the interface, as is the case with VM instantiation.

The optional "instance-id" element allows the VPN Forwarder to specify a unique 16 bit index that can be used by the Route Server to automatically assign a Route Distinguisher (RD) to any route subsequently advertised by the VPN Forwarder. In a scenario where the VPN Forwarder is advertising reachability information to multiple Route Servers it is desirable for reachability information to have an RD composed of the VPN Forwarder identifier (e.g., IPv4 address) and the "instance-id".

Example subscription request from end-system to external VPN Forwarder:

```
<iq type='set'
    from='end-system@domain.org'
    to='forwarder@domain.org'
    id='sub1'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <subscribe node='vpn-customer-name' jid='forwarder@domain.org'/>
    <options>
      <x xmlns='jabber:x:data' type='submit'>
        <field var='vpn#vlan_id'><value>100</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. This option is represented in the XMPP Pub-Sub Subscribe stanza a data form [xep-0004] field with the name "vpn#vlan_id". The example above uses the 802.1Q tag value of 100.

When a Route Server receives a subscription request for a specific VPN identifier it SHALL treat this request as an implicit request for item retrieval for all items in the Pub-Sub node that corresponds to the VPN.

If at any point all Virtual Interfaces associated with a given VPN Identifier are removed or deactivated from the End-System, then the End System XMPP client SHOULD generate an XMPP Pub-Sub Unsubscribe stanza to its server for the Pub-Sub node associated with the VPN Identifier.

Example unsubscribe request from co-located VPN Forwarder to Route Server:

```
<iq type='set'
    from='forwarder@domain.org'
    to='route-server@ietf.org'
    id='unsub1'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <unsubscribe
      node='vpn-identifier'
      jid='forwarder@domain.org'/>
  </pubsub>
</iq>
```

For a collocated VPN forwarder, and for an external VPN forwarder when there is an XMPP session with the End System, when an IP address is added to a virtual interface and the interface is activated, the end-system SHALL generate an XMPP Pub-Sub Publish request. This request publishes an item containing a single entry element based on the XML Schema Definition in Section 11. The ItemID of this item MUST be generated by the VPN Forwarder such that the value is unique within a Pub-Sub node. The ItemID MAY be formed by combining the VPN Forwarder's IP address, the instance-id value, and the entry address element. This format corresponds to the string representation of a BGP L3VPN NLRI in which the Route Distinguisher is given by the VPN Forwarder IP address and instance-id, and is easily identifiable by network operators. However, the format and/or structure of the ItemID is not stricly defined in this document, so long as uniqueness is guaranteed.

Publish request from VPN Forwarder to End-System Route Server:

```
<iq type='set'
    from='forwarder@domain.org'
    to='route-server@ietf.org'
    id='request1'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='vpn-customer-name'>
      <item id='192.0.2.1:1:203.0.113.42/32'>
        <entry xmlns='urn:ietf:params:xml:ns:bgp:l3vpn:unicast'>
          <nlri>
            <af>1</af>
            <address>203.0.113.42</address>
          </nlri>
          <next-hops>
            <next-hop>
              <af>1</af>
              <address>192.0.2.1</address>
              <label>10000</label>
              <tunnel-encapsulation-list>
                <tunnel-encapsulation>gre</tunnel-encapsulation>
                <tunnel-encapsulation>udp</tunnel-encapsulation>
              </tunnel-encapsulation-list>
            </next-hop>
          </next-hops>
          <sequence-number>1</sequence-number>
        </entry>
      </item>
    </publish>
  </pubsub>
</iq>
```

In this example, the VPN Forwarder JID is "forwarder@domain.org". The VPN Identifier "vpn-identifier" is used as the value of the node attribute of the subscribe element. The IP address of the Virtual Interface is 203.0.113.42/32. The IP address of the VPN Forwarder is 192.0.2.1 and it supports receiving MPLS packets via both GRE and UDP tunneling. Label 10000 has been assigned to this particular Virtual Interface.

The End-System Route Server will convert the information received in a 'publish' request into the corresponding BGP route information such that:

o It associates the specific request with a local VRF which it resolves by using the Pub-Sub 'node' attribute.

- It creates a BGP VPN route with a 'Route Distinguisher' (RD) which contains a unique 32bit value per end-system plus a 16bit instance-id, the specified IP prefix and 'label' received from the VPN Forwarder as the Network Layer Reachability Information (NLRI). The instance-id is either the value specified by the XMPP client in the subscribe stanza for the specific pubsub node or a locally generated value when that parameter is omitted.

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

- A BGP Tunnel Encapsulation Attribute [RFC5512] is generated for each 'tunnel-encapsulation' element specified in the XMPP message.

- The route is optionally associated with a MAC Mobility extended community [RFC7432] containing a sequence number for the route advertisement.

Conversely, when an interface operational status is determined to be down or an IP address is unconfigured the VPN forwarder generates an XMPP retract message to withdraw the route advertisement.

Retract request from VPN Forwarder to End-System Route Server:

```
<iq type='set'
    from='forwarder@domain.org'
    to='route-server@ietf.org'
    id='retract1'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <retract node='vpn-customer-name'>
      <item id='192.0.2.1:1:203.0.113.42/32'/>
    </retract>
  </pubsub>
</iq>
```

The retract stanza uses the ItemId to identify the item being retracted. The example retract stanza above uses the L3VPN NLRI string representation ItemId format used in the publish example.

Consistent with XMPP Pub-Sub [pubsub], event notifications will be generated whenever a VPN route is added, modified or deleted. This is true for VPN routes learned via XMPP clients as well as routes learned via BGP. For VPN routes that are learned via BGP (rather than XMPP clients) the Route Server SHOULD create XMPP Pub-Sub Publish stanzas or otherwise take steps to publish a persistent item under the NodeID associated with the VPN Identifier of the appropriate VRF(s). Thus the Pub-Sub node will contain items for every route for the associated VPN. Upon successfully publishing a

Pub-Sub item the XMPP server SHALL generate event notification messages and send them to all VPN Forwarders that are actively subscribed to that node. These event notification messages SHOULD be sent as soon as possible (without delay) in order to facilitate convergence and consistent reachability.

Example update notification message from Route Server to VPN Forwarder:


```
<message to='forwarder@domain.org' from='route-server@ietf.org'>
  <event xmlns='http://jabber.org/protocol/pubsub#event'>
    <items node='vpn-customer-name'>
      <item id='192.0.2.1:1:203.0.113.42/32'>
        <entry xmlns='urn:ietf:params:xml:ns:bgp:l3vpn:unicast'>
          <nlri>
            <af>1</af>
            <address>203.0.113.42/32</address>
          </nlri>
          <next-hops>
            <next-hop>
              <af>1</af>
              <address>192.0.2.1</address>
              <label>10000</label>
              <tunnel-encapsulation-list>
                <tunnel-encapsulation>gre</tunnel-encapsulation>
                <tunnel-encapsulation>udp</tunnel-encapsulation>
              </tunnel-encapsulation-list>
            </next-hop>
          </next-hops>
          <sequence-number>1</sequence-number>
        </entry>
      </item>
      <item >
        ...
      </item>
    </items>
  </event>
</message>
```


Notification messages SHOULD be generated whenever a VPN route is added, modified or deleted. These notification messages SHOULD contain only items that have been added, modified or deleted since any previous information that was sent to the VPN Forwarder. Notification messages can be segmented at the convenience of the Route Server.

Note that the Update from the Route Server to the VPN Forwarder does not contain the JID of the destination end-system. The "from"

attribute in the 'message' element contains the Route Server JID. The XMPP messages are point-to-point in nature, between a Forwarder and Route Server, even in the case when one XMPP publish request from a Forwarder may cause the Route Server to generate one or more event notifications.

When multiple possible routes exist for a given VPN IP address within a VRF it is the responsibility of the Route Server to select the best path to advertise to the VPN Forwarders. The routing entries published by the Route Server to VPN Forwarders MAY include multiple next-hops for the same forwarding entry. While BGP L3VPN NLRI encodes a single next-hop, multiple NLRI with different RDs may result in a single forwarding entry in a VRF with multiple next-hops. This functionality is known as "vrf multipath" in standard BGP L3VPN implementations. This "vrf multipath" behavior can be applied to both BGP and XMPP learned routing information. The criteria used for multipath selection is outside the scope of this document but SHOULD be consistent between the Route Servers within an administrative domain.

A VPN Forwarder uses locally originated information to generate MPLS label forwarding state, and this used to forward downstream traffic (i.e., traffic received from the network). Upstream traffic (i.e., received from a virtual interface) is forwarded according to the routing information received from one or more Route Servers that the VPN forwarder has an XMPP session with. In the case where multiple Router Servers are providing routing information for a specific NLRI the VPN Forwarder SHOULD select the following algorithm:

- o Prefer the highest local-preference value

- o Prefer the highest sequence-number

- o Tie-break on the Route Server IP address

When routes are withdrawn, the End-System Route Server generates an item "retract" request.

Route advertisements can have an optional sequence-number which help the route server determine the most recent route advertisement. The sequence number is determined by a mechanism outside the scope of this document. One option is to use time synchronization between compute nodes in order to have a globally coordinated timestamp. This timestamp can be used to identify the time of interface creation on the compute node.

Routes can also be associated with a "local-preference" attribute. This attribute maps to the BGP attribute of the same name for the purposes of route selection.

7. End-System Route Server behavior

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

When an End-System Route Server 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 End-System Route Servers have information regarding the mapping between the tuple ('end-system', 'vpn-name') and the BGP Route Targets used to import and export information from associated VRFs. This mapping is known via an out-of-band mechanism not specified in this document.

Whenever the End-System Route Server receives an XMPP subscription request, it SHALL consult its RT-Constraint Routing Information Base (RIB). If the Route Server does not have a locally originated RT-Constraint route that corresponds to the vpn-name present 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.

End-System Route Servers 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 entries 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 H1 and host H2 both contain a virtual interface that is a member of the same VPN.

```
                                 .------.
   +----+       +-----+        /        \        +-----+       +----+
   | H1 | <===> | RS1 | <===> ( BGP mesh ) <===> | RS2 | <===> | H2 |
   +----+       +-----+        \        /        +-----+       +----+
                                 `------'
```

Example Network with Two Hosts and Two Route Servers

Figure 2

Each of these hosts has a collocated VPN forwarder that has an XMPP session with an End-System Route Server, RS1 and RS2 our example, and these Route Servers are part of the same BGP mesh.

When a virtual interface is created on host H1, the local XMPP client generates an XMPP subscription stanza to its respective Route Server. This stanza contains a VPN identifier that has been assigned by the provisioning system. The Route Server 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 interface is operational, host H1 will publish any IP addresses that are configured on the respective virtual interface. This will in turn cause the End-System Route Server to advertise these (directly or indirectly) to any other BGP speaker on the network which is connected to an attachment point of that VPN.

The following table represents the contents of the VRF routing table on RS1 after the IPv4 address 203.0.113.42 has been added to the virtual interface on H1.

VPN IP address	NEXT-HOP	label	Known via
203.0.113.42/32	192.0.2.1	16	XMPP
203.0.113.48/32	198.51.100.10	20	BGP

It assumes that there is an attachment point for this VPN with the IPv4 address of 203.0.113.48 which is advertising a route to the IP address of an application running on H2 (203.0.113.48/32). Host H1 has an infrastructure IP address of 192.0.2.1 configured on its physical interface while host H2 has IP address 198.51.100.10.

The contents of the VRF routing table in the End-System Route Servers are advertised via XMPP Update notifications sent to H1, and a route

update for the IP address of H1 will be sent into the BGP mesh on to Route Server RS2 and from there, via XMPP to H2.

This information is used by the host to populate the forwarding table associated with that VPN. The following shows the VRF table on host H1

VPN IP address	Host address	label
203.0.113.42/32	localhost	16
203.0.113.48/32	198.51.100.10	20

When an application that uses the virtual interface on host H1 generates packets with a destination IP address of 203.0.113.48 these are routed by the VPN Forwarder implemented in the Host OS. The packets are encapsulated with a header that contains a label assigned by host H2, as shown in the figure, below.

```
               +--------+                 +--------+
app -- veth0 --|   H1   |=== [network] ===|   H2   |-- veth0 -- app
               +--------+                 +--------+

IP pkt ===> encap (GRE + label) ===> [IP net] ===> decap ===> IP pkt
            [192.51.100.10, 20]                 map 20 to veth0
```

Packet Flow from Application in H1 to Application in H2

Figure 3

In the case that the virtual interface on the host is associated with a guest OS, this guest OS has had its address resolution queries answered with the Virtual Router MAC address, or the MAC address of the destination MAY be supplied if it is in the same IP subnet (broadcast domain). When the Virtual Router MAC address is supplied, this is the address the guest OS uses as the destination MAC address in packets it originates that are outside its IP subnet. The VPN forwarder will replace the its MAC address with the MAC address of the next hop in the tenant virtual network (another End System or default gateway, for instance) before encapsulating the packet. packet.

End-System Route Servers are software applications that implement both the BGP IP VPN PE control plane as well as XMPP server functionality. These applications are not in the forwarding plane and MAY not be co-located with a network device.

Network devices MAY have direct BGP sessions to the End-System Route Servers. 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 or other network appliances that support this specification.

End-System Route Servers 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 the VPN clients must be redirected though a middle-box for inspection. Assume that the virtual interfaces 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. The End-System Route Server configuration is changed 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 VPN clients IP addresses. The "hub" VPN imports the VPN 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 VPN clients). In such a scenario, a single physical interface can connect the middle-box to the clients 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 (End-System Route Servers) 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. IANA Considerations

IANA has allocated the value 13 corresponding to "MPLS in UDP Encapsulation" from the "BGP Tunnel Encapsulation Attribute Tunnel Types" registry, using this document as reference. We request that this allocation be made permanent.

This document defines a URN namespace used to encode L3VPN Unicast routing information compliant with the registration procedure define in [RFC3688].

URI: urn:ietf:params:xml:ns:bgp:l3vpn:unicast

Description: This is the XML namespace name for L3VPN Unicast routing information.

Registrant Contact: IETF BESS Working Group <bess@ietf.org>

10. Security Considerations

As with BGP/MPLS L3VPN, we assume that the tenant networks have no direct reachability to the infrastructure network. The threat models to consider are:

- The possibility that an attacker on a tenant network may inject traffic to a different network (for instance belonging to a different tenant).

- Denial of service attacks from within a tenant network.

- Attacks from a tenant network to the infrastructure via unauthorized or malicious control traffic.

- Attacks from within the infrastructure network.

Traffic in BGP/MPLS L3VPNs is forwarded based on the contents of VRF tables, calculated according to configured routing policy (route-target import/export policies). It is assumed that the configuration management system responsible for provisioning these policies only accepts requests that are correctly authenticated, and follow a pre-defined access policy. It is also assumed that an attacker doesn't have the ability to inject packets in the infrastructure that mimic the encapsulated used between PE devices. This specification recommends that operators ensure that MPLS over GRE and MPLS over UDP traffic is not allowed to enter the infrastructure network. VPN forwarders MAY also choose to perform a reverse path forwarding lookup (i.e., lookup the source IP address of the payload packet) and discard traffic that doesn't match the expected next-hop(s) for the reverse route.

As with BGP/MPLS L3VPN, an attacker on a tenant network may inject packets that consume a disproportional share of infrastructure resources, either in terms of bandwidth or CE packet forwarding capacity. VPN forwarders SHOULD provide the ability to rate limit traffic from a specific virtual interface. When the VPN forwarder uses other finite resources on a per traffic basis, such as internal tables used to cache the result access control validation, it SHOULD provide a mechanism to limit the usage of these resources on a per virtual interface basis.

The control protocol exchanges between application instances (e.g., the virtual machine) behind a virtual interface and the VPN forwarder are typically limited to ARP/ND exchanges and the proxying of services such as DHCP and DNS. The ARP/ND information received from the application instance SHOULD NOT be used to populate routing or forwarding tables directly. The control of what MACs and IP addresses are accepted by a virtual interface SHOULD reside in the configuration management system that creates said virtual interface.

The XMPP session between end-systems and the Route Servers SHOULD use TLS with mutual authentication. One possible strategy is to distribute pre-signed certificates to end-systems which are presented as proof of authorization to the Route Server. BGP sessions SHOULD be authenticated. 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.

11. XML schema

The following schema defines the XML elements that are used to communicate unicast reachability information between the Route Server and VPN Forwarder:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace=
        "urn:ietf:params:xml:ns:bgp:l3vpn:unicast">

<xsd:simpleType name="TunnelEncapsulationType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="gre"/>
            <!-- RFC 4023 -->
        <xsd:enumeration value="udp"/>
            <!-- RFC 7510 -->
        <xsd:enumeration value="vxlan"/>
            <!-- RFC 7348 -->
    </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="TunnelEncapsulationListType">
  <xsd:sequence>
    <xsd:element name="tunnel-encapsulation"
        type="TunnelEncapsulationType"
        maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="NextHopType">
  <xsd:sequence>
    <xsd:element name="af" type="xsd:integer"/>
    <xsd:element name="address" type="xsd:string"/>
    <xsd:element name="label" type="xsd:integer"/>
    <xsd:element name="tunnel-encapsulation-list"
        type="TunnelEncapsulationListType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="NextHopListType">
  <xsd:sequence>
    <xsd:element name="next-hop" type="NextHopType"
        maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="IPAddressType">
  <xsd:sequence>
    <xsd:element name="af" type="xsd:integer"/>
    <xsd:element name="safi" type="xsd:integer"/>
    <xsd:element name="address" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
```

```
<xsd:complexType name="EntryType">
    <xsd:all>
        <xsd:element name="nlri" type="IPAddressType"/>
        <xsd:element name="next-hops" type="NextHopListType"/>
        <xsd:element name="sequence-number" type="xsd:integer"/>
        <xsd:element name="local-preference" type="xsd:integer"/>
    </xsd:all>
</xsd:complexType>

<xsd:complexType name="ItemType">
  <xsd:sequence>
    <xsd:element name="entry" type="EntryType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ItemsType">
    <xsd:sequence>
        <xsd:element name="item" type="ItemType"
            maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:element name="items" type="ItemsType"/>

</xsd:schema>
```

12. Acknowledgements

Pedro Marques contributed much of the original content of this document.

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.

Amit Shukla and Ping Pan contributed to earlier versions of this document.

Benson Schliesser provided a detailed review of the document and helped clarify several sections.

13. References

13.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <http://www.rfc-editor.org/info/rfc3688>.

[RFC4023] Worster, T., Rekhter, Y., and E. Rosen, Ed., "Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)", RFC 4023, DOI 10.17487/RFC4023, March 2005, <http://www.rfc-editor.org/info/rfc4023>.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <http://www.rfc-editor.org/info/rfc4364>.

[RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, DOI 10.17487/RFC4456, April 2006, <http://www.rfc-editor.org/info/rfc4456>.

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

[RFC5512] Mohapatra, P. and E. Rosen, "The BGP Encapsulation Subsequent Address Family Identifier (SAFI) and the BGP Tunnel Encapsulation Attribute", RFC 5512, DOI 10.17487/RFC5512, April 2009, <http://www.rfc-editor.org/info/rfc5512>.

[RFC5798] Nadas, S., Ed., "Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6", RFC 5798, DOI 10.17487/RFC5798, March 2010, <http://www.rfc-editor.org/info/rfc5798>.

[RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, DOI 10.17487/RFC6120, March 2011, <http://www.rfc-editor.org/info/rfc6120>.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <http://www.rfc-editor.org/info/rfc7348>.

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

[RFC7510] Xu, X., Sheth, N., Yong, L., Callon, R., and D. Black, "Encapsulating MPLS in UDP", RFC 7510, DOI 10.17487/RFC7510, April 2015, <http://www.rfc-editor.org/info/rfc7510>.

[RFC7622] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Address Format", RFC 7622, DOI 10.17487/RFC7622, September 2015, <http://www.rfc-editor.org/info/rfc7622>.

[xep-0004] Eatmon, R., Hildebrand, J., Miller, J., Muldowney, T., and P. Saint-Andre, "Data Forms", XEP 0004, August 2007.

[pubsub] Millard, P., Saint-Andre, P., and R. Meijer, "Publish-Subscribe", XEP 0060, July 2010.

13.2. Informational References

[RFC1027] Carl-Mitchell, S. and J. Quarterman, "Using ARP to implement transparent subnet gateways", RFC 1027, DOI 10.17487/RFC1027, October 1987, <http://www.rfc-editor.org/info/rfc1027>.

[RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, DOI 10.17487/RFC5575, August 2009, <http://www.rfc-editor.org/info/rfc5575>.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <http://www.rfc-editor.org/info/rfc5880>.

[RFC7365] Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for Data Center (DC) Network Virtualization", RFC 7365, DOI 10.17487/RFC7365, October 2014, <http://www.rfc-editor.org/info/rfc7365>.

[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

Stuart Mackie
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089

Email: wsmackie@juniper.net

Luyuan Fang
eBay
2025 Hamilton Avenue
San Jose, CA 95125

Email: lufang@ebay.com

Nischal Sheth
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089

Email: nsheth@juniper.net

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

Email: mnapierala@att.com

Nabil Bitar
Nokia

Email: nabil.bitar@nokia.com